Package ‘DImodels’

October 1, 2020

Type Package

Title Diversity-Interactions (DI) Models

Version 1.0

Date 2020-09-28

Author Rafael de Andrade Moral [aut, cre],
John Connolly [aut],
Caroline Brophy [aut]

Maintainer Rafael de Andrade Moral <rafael.deandrademoral@mu.ie>

Imports methods, graphics, stats, utils, hnp, rootSolve

Suggests ggplot2, knitr, rmarkdown

VignetteBuilder knitr

Depends R (>= 3.6.0)

Description The 'DImodels' package is suitable for analysing data from biodiversity and ecosystem function studies using the Diversity-Interactions (DI) modelling approach introduced by Kirwan et al. (2009) <doi:10.1890/08-1684.1>. Suitable data will contain proportions for each species and a community-level response variable, and may also include additional factors, such as blocks or treatments. The package can perform data manipulation tasks, such as computing pairwise interactions (the DI_data_prepare() function), can perform an automated model selection process (the autoDI() function) and has the flexibility to fit a wide range of user-defined DI models (the DI() function).

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2020-10-01 09:00:05 UTC

R topics documented:

`DImodels-package` *Diversity-Interactions (DI) Models*

Description

The 'DImodels' package is suitable for analysing data from biodiversity and ecosystem function studies using the Diversity-Interactions (DI) modelling approach introduced by Kirwan et al. (2009) <doi:10.1890/08-1684.1>. Suitable data will contain proportions for each species and a community-level response variable, and may also include additional factors, such as blocks or treatments. The package can perform data manipulation tasks, such as computing pairwise interactions (the DI_data_prepare() function), can perform an automated model selection process (the autoDI() function) and has the flexibility to fit a wide range of user-defined DI models (the DI() function).

Details

Introduction to Diversity-Interactions models

Diversity-Interactions (DI) models (Kirwan et al 2009) are a set of tools for analysing and interpreting data from experiments that explore the effects of species diversity on community-level responses; for example, the effect of increasing community diversity on biomass production in a grassland ecosystem. Most analyses of diversity experiments quantify community diversity in terms of species richness, the number of species present. The DI method modifies this presence/absence approach in mixed communities by taking species relative abundance in the community into account. So, instead of ignoring differences in community responses across communities with the same species but with different species relative abundances, the DI approach aims to understand and explain these differences.

What variables will a suitable dataset contain?

The DI approach assesses the effect of species diversity on a community response over a period of time. For data from a biodiversity study with S species to be suited to the DI approach, the variables that are required on each experimental unit are:

1. A set of proportions $p1, p2, ..., pS$ that characterise the proportions of each species at a defined starting point in time. These proportions (or relative abundances) of species in the community (pi for the ith species) range between 0 (absence) and 1 (monoculture - the only species present) and the sum of all the pi values for a community is always 1.
2. A community-level response variable, recorded a period of time after the initial species proportions were recorded.

The dataset may also contain other variables such as a block, density or treatments.

What are Diversity-Interactions models?

A DI model typically has three components and takes the form:

$$y = Identities + Interactions + Structures + \epsilon$$

where *y* is a community-level response, the *Identities* are the effects of species identities and enter the model as individual species proportions at the beginning of the time period, the *Interactions* are the interactions among the species proportions, while *Structures* include other experimental structures such as blocks, treatments or density. In a three species system, with an experimental blocking structure, a possible DI model is:

$$y = \beta_1 p1 + \beta_2 p2 + \beta_3 p3 + \delta_{12} p1p2 + \delta_{13} p1p3 + \delta_{23} p2p3 + \alpha_k + \epsilon$$

Where *p1*, *p2* and *p3* are the species proportions of species 1, 2 and 3 respectively, and α_k is the effect of block *k*. The error term ϵ is usually assumed to be normally distributed with mean 0 and constant variance (and assumed independent and identically distributed).

In a monoculture of the ith species, the expected performance of *y* in block *k* is β_i. In mixture, the *Identities* component provides the expected performance as a weighted average of monoculture responses, and the *Interactions* component is added to it to give the overall expected performance of the mixed community. The *Interactions* component addresses the question: do mixed communities perform differently from what might be expected from the weighted averaging of monoculture performances? Note that in Kirwan et al 2009 the *Interactions* component is referred to as the diversity effect, however, here we use the more general term "*Interactions*", since the interpretation of how species diversity affects the response is a combination of both the *Identities* and *Interactions* components. The community with the best overall performance may depend on both the *Identities* and *Interactions* and will rarely be the community with the largest net interactions effect. The equation above provides an example of a DI model where there is a unique interaction term for each pair of species. It is possible to test various constraints among interactions, some of which may be motivated by the context of the data (Kirwan et al 2009).

The *Interactions* component may also include an non-linear exponent parameter θ on each $pipj$ term, where a value different to one allows the importance and impact of interaction terms to be altered (Connolly et al 2013).

What can the `DImodels` package do?

The DI approach is a full regression method where the response of the community is characterised by the effects of diversity variables such as *Identities* and *Interactions* components, and by experimental structural variables such as blocks, density and treatments. All of these may be important determinants of response and so should be included in the analysis of community responses. The `DImodels` package aims to make it easier to analyse data using DI models.

Currently, the DImodels package contains three main functions: autoDI, DI and DI_data_prepare. Here we give a brief overview of each, and link to the respective help files for further information.

auto_DI: This function gives a simple overview of the successive contribution of the *Structures*, *Identities* and *Interactions* components to the model via an automated model selection process. It will identify the 'best' model from a (limited) subset of all possible DI models. However, autoDI may need to be supplemented by more refined analysis, for example, auto_DI does not test for interactions between the terms in the *Structures* and *Identities* components. It can also only facilitate one block, one density and one treatment variable. However, it is a very useful starting point for DI model exploration. Further information at: autoDI.

DI: This function can fit a wide range of DI models and includes the flexibility to test for multiple treatments or additional interactions, for example, between terms in the *Identities* and *Structures* components. The DI function fits one user-defined DI model at a time, and it allows for flexibility in the form of the model through a combination of in-built argument options and additional user-defined options. Further information at: DI.

DI_data_prepare: This function can compute various forms of interactions among the pi variables. This function is not required when using autoDI, or for the in-built argument options in DI, but may be needed when specifying additional user-defined options in DI. Further information at: DI_data_prepare and for examples of when it is required see DI.

There are additional data manipulation functions: DI_data_E_AV, DI_data_FG, DI_data_ADD, that will individually prepare the various forms of interactions that DI_data_prepare does in one function. The function DI_data_fullpairwise will compute all individual pairwise interactions. Further information on all data manipulation functions at: DI_data_manipulation.

There are two other functions: 1) theta_CI can fit a confidence interval to the parameter theta, when it has been estimated using either autoDI or DI, and 2) DI_compare that can compare a fitted DI model to the 'reference' model (see autoDI) for details about the reference model).

Challenges with fitting and interpreting Diversity-Interactions models

Analysing data using DI methods can be tricky for people familiar with ordinary regression models and the DImodels package aims to make analysis easier. The difficulties lie in:

1. The lack of familiarity in dealing with the *Identities* component variables pi, which must sum to 1. This constraint can lead to interpretative issues, and estimation problems with some widely used R software.
2. The novelty of specifying the *Interactions* components in terms of many pairwise interaction terms whose numbers may greatly increase with *S*. The number of pairwise terms can often be reduced by identifying biologically meaningful patterns among them, for example, through functional grouping of species. This may greatly reduce the number of coefficients to be estimated and interpreted in the model (Kirwan et al, 2009).
3. The introduction of a power coefficient theta (θ) for all pairwise interaction terms (Connolly et al, 2013). This parameter can be very useful in describing the effect of community evenness on community response, i.e., whether response changes rapidly or slowly across communities where the relative abundance of species changes from being equal for all species to dominance by one or more species.

Limitations of the DImodels package

Currently, the DImodels package does not support multivariate responses or repeated measurements on the same experimental unit. It also does not currently support the random effects approach to modelling pairwise interactions that was developed by Brophy et al 2017.

Author(s)

Rafael de Andrade Moral [aut, cre], John Connolly [aut], Caroline Brophy [aut]

Maintainer: Rafael de Andrade Moral <rafael.deandrademoral@mu.ie>

References

Brophy C, A Dooley, L Kirwan, JA Finn, J McDonnell, T Bell, MW Cadotte and J Connolly. (2017) Biodiversity and ecosystem function: Making sense of numerous species interactions in multi-species communities. Ecology 98, 1771-1778.

Connolly J, T Bell, T Bolger, C Brophy, T Carnus, JA Finn, L Kirwan, F Isbell, J Levine, A Lüscher, V Picasso, C Roscher, MT Sebastia, M Suter and A Weigelt (2013) An improved model to predict the effects of changing biodiversity levels on ecosystem function. Journal of Ecology, 101, 344-355.

Kirwan L, J Connolly, JA Finn, C Brophy, A Lüscher, D Nyfeler and MT Sebastia (2009) Diversity-interaction modelling - estimating contributions of species identities and interactions to ecosystem function. Ecology, 90, 2032-2038.

See Also

autoDI DI DI_data_prepare DI_data_manipulation theta_CI

Example datasets: The Bell dataset. The sim1 dataset. The sim2 dataset. The sim3 dataset. The sim4 dataset. The sim5 dataset. The Switzerland dataset.

Examples

```
## Load the Switzerland data
  data(Switzerland)

## Check that the proportions sum to 1 (required for DI models)
## p1 to p4 are in the 4th to 7th columns in Switzerland
  Switzerlandsums <- rowSums(Switzerland[4:7])
  summary(Switzerlandsums)

## Example of autoDI
  auto1 <- autoDI(y = "yield", density = "density", prop = c("p1","p2","p3","p4"),
                  treat = "nitrogen", FG = c("G","G","L","L"), data = Switzerland,
                  selection = "Ftest")
  summary(auto1)

## Example of DI
  m1 <- DI(y = "yield", density = "density", prop = 4:7, treat = "nitrogen",
           FG = c("G","G","L","L"), DImodel = "FG", data = Switzerland)
  summary(m1)

## Example of DI_data_prepare.
## Create interaction variables and incorporate them into a new data frame Switzerland2.
## Switzerland2 will contain the new variables:  AV, E, p1_add, p2_add, p3_add, p4_add,
##  bfg_G_L, wfg_G, wfg_L.
 newlist <- DI_data_prepare(y = "yield", prop = c("p1","p2","p3","p4"), FG = c("G","G","L","L"),
                            data = Switzerland)
  Switzerland2 <- data.frame(newlist$newdata, newlist$FG)
```

autoDI *Automated Diversity-Interactions Model Fitting*

Description

This function provides an automated way to fit a (limited) range of Diversity-Interactions (DI) models. Using one of several selection criteria, `autoDI` will identify the best DI model from the range fitted via a three-step selection process (see Details for more information). While `autoDI` can be a useful starting point for fitting DI models, its range of models is not exhaustive and additional model fitting or testing via `DI` is likely to be required. For instance, `autoDI` does not test for interactions of a treatment with other variables in the model.

Usage

```
autoDI(y, block, density, prop, treat, FG = NULL, data,
       selection = c("Ftest","AIC","AICc","BIC","BICc"),
       step0 = FALSE, step4 = TRUE)
```

Arguments

`y`	The column name of the response vector, which must be in quotes, for example, `y = "yield"`.
`block`	The name of the block variable (if present), which must be in quotes, for example, `block = "block"`. If no blocking variable, omit this argument.
`density`	The name of the density variable (if present), which must be in quotes, for example, `density = "density"`. If no density variable, omit this argument.
`prop`	A vector of *s* column names identifying the species proportions in each row in the dataset. For example, if the species proportions columns are labelled p1 to p4, then `prop = c("p1","p2","p3","p4")`. Alternatively, the column numbers can be specified, for example, `prop = 4:7`, where the species proportions are in the 4th to 7th columns.
`treat`	The name of a column in the dataset containing the value of a treatment factor or covariate. The treatment name must be included in quotes, for example, `treat = "nitrogen"`. (Only one treatment or covariate is permitted in `autoDI`, but see `DI` for options involving more than one treatment.) If the treatment is a factor, the variable must already be specified as a factor prior to using `autoDI`.
`FG`	If species are classified by *g* functional groups, this parameter gives a text list (of length *s*) of the functional group to which each species belongs. For example, for four grassland species with two grasses and two legumes: FG could be `FG = c("G","G","L","L")`, where G stands for grass and L stands for legume.
`data`	Specify the dataset, for example, `data = Switzerland`. The dataset name should not appear in quotes.
`selection`	Selection method to be used in the automated model selection process. Options are `"Ftest"`, `"AIC"`, `"AICc"`, `"BIC"` and `"BICc"`. The default is `selection = "Ftest"`.

`step0` By default, `autoDI` outputs steps 1 - 4, however, an initial step 0 can be included by specifying `step0 = TRUE`. This will fit a model with an intercept only, and will sequentially add in and test the inclusion of `block`, `density` and `treat`, if they are specified in the `autoDI` arguments.

`step4` Step 4 performs a lack of fit test for the final model selected by steps 1 - 3. By default, `step4 = TRUE`, but it can be omitted by specifying `step4 = FALSE`.

Details

What are Diversity-Interactions models?

Diversity-Interactions (DI) models (Kirwan et al 2009) are a set of tools for analysing and interpreting data from experiments that explore the effects of species diversity on community-level responses. We recommend that users of the `DImodels` package read the short introduction to DI models (available at: `DImodels`). Further information on DI models is available in Kirwan et al 2009 and Connolly et al 2013.

Checks on data prior to using `autoDI`.

Before using `autoDI`, check that the species proportions for each row in your dataset sum to one. See the 'Examples' section for code to do this. An error message will be generated if the proportions don't sum to one.

How does the `autoDI` function work?

The `autoDI` function identifies the 'best' Diversity-Interactions (DI) model from a specific range of proposed models using a three-step process of selection (Steps 1 to 3) and performs a lack of fit test on the selected model (Step 4). Only a limited subset of all possible models are examined under autoDI, for example, interactions involving experiment structural terms (`block`, `density`, `treat`) are not explored. The `autoDI` function can provide an excellent initial analysis, but often additional modelling and exploration will be required using the `DI` function.

Steps 1-3 outlined below provide details on the automated model selection process followed by `autoDI`. Step 4 is a lack of fit test for the selected model. Step 0 may also be included (`step0 = TRUE`) as an initial step to sequentially test the inclusion of experiment structural variables (`block`, `density`, `treat`), prior to fitting the DI models in Step 1.

All models in `autoDI` Steps 1 and 2 are estimated using iteratively reweighted least squares, via the glm package. Profile likelihood estimation is used in Step 3 to estimate the parameter θ.

The default selection method used is `selection = "Ftest"`, which will return the appropriate F test statistic value(s) and p-value(s) in each step. When any of the information theoretic approaches (`"AIC"`, `"AICc"`, `"BIC"` or `"BICc"`) are specified, the model with the lowest value is selected in each step, even if it is only the lowest by a tiny margin; therefore, it is recommended to examine the information theoretic values across all models.

Step 1

Five models are fitted, sequentially, each building on the previous model, and compared. If the experiment structural variables `treat`, `block` or `density` are specified, they will be included as an additive factor or covariate in **each** of the five models, but interactions between them and the DI model terms will not be included or tested.

Assume that `FG` (functional groups) have been specified. The five DI models are:

- `STR`: This model contains an intercept, and block, density and treatment if present; STR stand for 'structural' and represents experiment structural variables.

- ID: The species proportions are added to the STR model. The proportions sum to 1, and are included in the model as 0 + p1 + ... + ps, where *s* is the number of species in the pool, as specified in the prop option.
- AV: The terms in the ID model, plus a single 'average' pairwise interaction term. For the Switzerland dataset, the single variable that is added to the model is computed as: AV = p1*p2 + p1*p3 + p1*p4 + p2*p3 + p2*p4 + p3*p4.
- FG: The terms in the ID model, plus interaction terms related to functional groups. These terms describe the average effects of interaction between pairs of species within each functional group, and between pairs of species from different functional groups. For example, in the Switzerland dataset there are four species with FG = c("Grass","Grass","Legume","Legume"), and there are six pairwise interactions, one between the two grasses, one between the two legumes, and four between grass and legume species. Grouping these interactions gives three terms, one for interactions between grasses, one for interactions between legumes and one for between a grass and a legume species. The model assumes that any grass will interaction with any legume in the same way and the 'between functional group grass-legume' variable is computed as: bfg_G_L = p1*p3 + p1*p4 + p2*p3 + p2*p4. If there were more than two grasses in the dataset, this model would assume that any pair of grasses interact in the same way, similarly for legumes. If the FG argument is not specified, this model is omitted from Step 1.
- FULL: The terms in the ID model, plus an interaction term for each pair of species. When there are *s* species, there are *s*(s-1)/2* pairwise interaction terms, i.e., for the Switzerland dataset with four species, there are six interactions that are each added to the model (in R formula syntax: p1:p2, p1:p3, p1:p4, p2:p3, p2:p4 and p3:p4).

If the FG argument is omitted, the FG model will be replaced by:

- ADD: The terms in the ID model, plus a species specific 'additive' interaction term for each species. These terms measure the interactive contribution of each species with any other species and are denoted λ_i for the *i*th species. The interaction between any pair of species *i* and *j* is computed as $\lambda_i + \lambda_j$.

Step 2

If treat is specified, the selected model in Step 1 includes the treat variable (since all models in Step 1 include treat if present). Here, the selected model from Step 1 is re-fitted without treat and the models with and without the treatment are compared using the method specified by the selection argument.

If treat was not specified, this step is redundant.

Step 3

For the selected model in Step 2 (or Step 1 if no treat was specified), a non-linear parameter theta is included as a power on all $pi * pj$ components of each interaction variable in the model, and the models with and without theta are compared using the method specified by the selection argument. For example, for the Switzerland data, with four species, for the AV model, the parameter theta enters the model as:

AV_theta = (p1*p2)^theta + (p1*p3)^theta + (p1*p4)^theta + (p2*p3)^theta + (p2*p4)^theta + (p3*p4)^theta

In Steps 1 and 2, theta is implicitly assumed to equal 1 (since, for example, (p1*p2)^1 = p1*p2), while in Step 3, theta is estimated using profile likelihood and then assessed for a difference from

1. In the profile likelihood estimation, `theta` is tested across the range 0.01 to 1.5. Then, the two models (with `theta` estimated and with `theta` set to 1) are compared using the method specified by the `selection` argument.

If there are no species interaction terms included in the best model selected in Step 1 or Step 2, this step is redundant. I.e., if there is no evidence of species interactions, then there is no need to test for `theta`.

Details on DI models that include `theta` are described in Connolly et al 2013.

Step 4

Step 4 provides a lack of fit test for the model selected by Steps 1 to 3. A factor called 'community' is created that has a level for each unique setting of the species proportions (as specified in the `prop` argument). The 'reference' model includes all terms in the model that was selected by Steps 1 to 3, plus the factor community. The reference model is compared to the model selected by Steps 1 to 3 via an F-test (regardless of the `selection` argument value), thus providing a lack of fit test.

Note, that the reference model is never intended to be a candidate model, it is only fitted for the purpose of testing lack of fit. If the test result is significant, it indicates that there is a lack of fit in the Diversity-Interactions model selected by `autoDI`.

Note also, that the test will not work if all combinations of the species proportions are unique. In this instance, the option `Step4 = FALSE` should be selected.

Value

The function returns a list including

`model_list` a list with all fitted models without any treatment effects, with `estimate_theta = FALSE`

`model_list_theta` a list with all fitted models without any treatment effects, with `estimate_theta = TRUE`

`model_list_treat` a list with all fitted models including treatment effects, with `estimate_theta = FALSE`

`model_list_theta_treat` a list with all fitted models including treatment effects, with `estimate_theta = TRUE`

`logLik` a vector with the log-likelihood values for the fitted models

`AIC` a vector with the AIC values for the fitted models

`AICc` a vector with the AICc values for the fitted models

`BIC` a vector with the BIC values for the fitted models

`BICc` a vector with the BICc values for the fitted models

`df` a vector with the residual d.f. values for the fitted models

`selected_model_code` the model tag for the selected model

`selected_model` the description of the selected model

`selected_model_obj` the selected model object

`data` the dataset including all manipulated variables

`family` the `family` passed to `glm`

Author(s)

Rafael A. Moral, John Connolly and Caroline Brophy

References

Connolly J, T Bell, T Bolger, C Brophy, T Carnus, JA Finn, L Kirwan, F Isbell, J Levine, A Lüscher, V Picasso, C Roscher, MT Sebastia, M Suter and A Weigelt (2013) An improved model to predict the effects of changing biodiversity levels on ecosystem function. Journal of Ecology, 101, 344-355.

Kirwan L, J Connolly, JA Finn, C Brophy, A Lüscher, D Nyfeler and MT Sebastia (2009) Diversity-interaction modelling - estimating contributions of species identities and interactions to ecosystem function. Ecology, 90, 2032-2038.

See Also

DI

Other examples using autoDI: The sim1 dataset examples. The sim2 dataset examples. The sim3 dataset examples. The sim4 dataset examples. The sim5 dataset examples. The Switzerland dataset examples.

Examples

```
## Load the Switzerland data
  data(Switzerland)
## Summarise the Switzerland data
  summary(Switzerland)

## Check that the proportions sum to 1 (required for DI models)
## p1 to p4 are in the 4th to 7th columns in Switzerland
  Switzerlandsums <- rowSums(Switzerland[4:7])
  summary(Switzerlandsums)

## Perform automated model fitting on the Switzerland dataset

## Model selection by F-test
  auto1 <- autoDI(y = "yield", density = "density", prop = c("p1","p2","p3","p4"),
                  treat = "nitrogen", FG = c("G", "G", "L", "L"), data = Switzerland,
                  selection = "Ftest")
  summary(auto1)


## Using column numbers to indicate which columns contain the proportions and including Step 0
  auto2 <- autoDI(y = "yield", density = "density", prop = 4:7, treat = "nitrogen",
             FG = c("G", "G", "L", "L"), data = Switzerland, selection = "Ftest", step0 = TRUE)
  summary(auto2)

## Exclude the FG (functional group) argument to include the additive species "ADD" model in Step 1
  auto3 <- autoDI(y = "yield", density = "density", prop = 4:7, treat = "nitrogen",
                  data = Switzerland, selection = "Ftest")
  summary(auto3)
```

autoDI_methods *Methods for* `autoDI` *Objects*

Description

Different methods that can be used with objects of class `autoDI`.

Usage

```
## S3 method for class 'autoDI'
AIC(object, ...)
## S3 method for class 'autoDI'
AICc(obj)
## S3 method for class 'autoDI'
BIC(object, ...)
## S3 method for class 'autoDI'
AICc(obj)
## S3 method for class 'autoDI'
anova(object, ...)
## S3 method for class 'autoDI'
coef(object, ...)
## S3 method for class 'autoDI'
print(x, ...)
## S3 method for class 'autoDI'
summary(object, ...)
## S3 method for class 'autoDI'
formula(x, ...)
## S3 method for class 'autoDI'
hnp(object, ...)
## S3 method for class 'autoDI'
logLik(object, ...)
## S3 method for class 'autoDI'
model.matrix(object, ...)
## S3 method for class 'autoDI'
plot(x, ...)
```

Arguments

`object`	an `autoDI` model object
`obj`	an `autoDI` model object
`x`	an `autoDI` model object
`...`	further arguments passed to `anova`, `coef`, `print`, `summary`, `formula`, `hnp`, `model.matrix`, or `plot`

Author(s)

Rafael A. Moral, John Connolly and Caroline Brophy

See Also

autoDI

`Bell` *The "Bell" Dataset*

Description

This dataset comes from a bacterial biodiversity experiment (Bell et al 2005). The bacterial ecosystems used were from semi-permanent rainpools that form in bark-lined depressions near the base of large European beech trees (*Fagus sylvatica*). Microcosms consisting of sterile beech leaf disks and 10 ml of liquid (phosphate buffer) were inoculated with random combinations of 72 bacterial species isolated from these ecosystems. A total of 1,374 microcosms were constructed at richness levels of 1, 2, 3, 4, 6, 8, 9, 12, 18, 24, 36 and 72 species. The daily respiration rate of the bacterial community in each microcosm was measured over three time intervals (days 0-7, 7-14 and 14-28) and the average over the three time intervals was recorded.

Usage

```
data("Bell")
```

Format

A data frame with 1374 observations on the following 76 variables:

`id` A numeric vector uniquely identifying each row of the dataset.

`community` A numeric vector identifying each unique community, i.e., two rows with the same community value also share the same set of p1 to p72 values.

`richness` The number of species included in the initial composition, i.e., the number of proportions from p1 to p72 that are >0.

`p1` A numeric vector indicating the initial proportion of species 1 in the community.

`p2` A numeric vector indicating the initial proportion of species 2 in the community.

`p3` A numeric vector indicating the initial proportion of species 3 in the community.

`p4` A numeric vector indicating the initial proportion of species 4 in the community.

`p5` A numeric vector indicating the initial proportion of species 5 in the community.

`p6` A numeric vector indicating the initial proportion of species 6 in the community.

`p7` A numeric vector indicating the initial proportion of species 7 in the community.

`p8` A numeric vector indicating the initial proportion of species 8 in the community.

`p9` A numeric vector indicating the initial proportion of species 9 in the community.

`p10` A numeric vector indicating the initial proportion of species 10 in the community.

`p11` A numeric vector indicating the initial proportion of species 11 in the community.
`p12` A numeric vector indicating the initial proportion of species 12 in the community.
`p13` A numeric vector indicating the initial proportion of species 13 in the community.
`p14` A numeric vector indicating the initial proportion of species 14 in the community.
`p15` A numeric vector indicating the initial proportion of species 15 in the community.
`p16` A numeric vector indicating the initial proportion of species 16 in the community.
`p17` A numeric vector indicating the initial proportion of species 17 in the community.
`p18` A numeric vector indicating the initial proportion of species 18 in the community.
`p19` A numeric vector indicating the initial proportion of species 19 in the community.
`p20` A numeric vector indicating the initial proportion of species 20 in the community.
`p21` A numeric vector indicating the initial proportion of species 21 in the community.
`p22` A numeric vector indicating the initial proportion of species 22 in the community.
`p23` A numeric vector indicating the initial proportion of species 23 in the community.
`p24` A numeric vector indicating the initial proportion of species 24 in the community.
`p25` A numeric vector indicating the initial proportion of species 25 in the community.
`p26` A numeric vector indicating the initial proportion of species 26 in the community.
`p27` A numeric vector indicating the initial proportion of species 27 in the community.
`p28` A numeric vector indicating the initial proportion of species 28 in the community.
`p29` A numeric vector indicating the initial proportion of species 29 in the community.
`p30` A numeric vector indicating the initial proportion of species 30 in the community.
`p31` A numeric vector indicating the initial proportion of species 31 in the community.
`p32` A numeric vector indicating the initial proportion of species 32 in the community.
`p33` A numeric vector indicating the initial proportion of species 33 in the community.
`p34` A numeric vector indicating the initial proportion of species 34 in the community.
`p35` A numeric vector indicating the initial proportion of species 35 in the community.
`p36` A numeric vector indicating the initial proportion of species 36 in the community.
`p37` A numeric vector indicating the initial proportion of species 37 in the community.
`p38` A numeric vector indicating the initial proportion of species 38 in the community.
`p39` A numeric vector indicating the initial proportion of species 39 in the community.
`p40` A numeric vector indicating the initial proportion of species 40 in the community.
`p41` A numeric vector indicating the initial proportion of species 41 in the community.
`p42` A numeric vector indicating the initial proportion of species 42 in the community.
`p43` A numeric vector indicating the initial proportion of species 43 in the community.
`p44` A numeric vector indicating the initial proportion of species 44 in the community.
`p45` A numeric vector indicating the initial proportion of species 45 in the community.
`p46` A numeric vector indicating the initial proportion of species 46 in the community.
`p47` A numeric vector indicating the initial proportion of species 47 in the community.

p48 A numeric vector indicating the initial proportion of species 48 in the community.

p49 A numeric vector indicating the initial proportion of species 49 in the community.

p50 A numeric vector indicating the initial proportion of species 50 in the community.

p51 A numeric vector indicating the initial proportion of species 51 in the community.

p52 A numeric vector indicating the initial proportion of species 52 in the community.

p53 A numeric vector indicating the initial proportion of species 53 in the community.

p54 A numeric vector indicating the initial proportion of species 54 in the community.

p55 A numeric vector indicating the initial proportion of species 55 in the community.

p56 A numeric vector indicating the initial proportion of species 56 in the community.

p57 A numeric vector indicating the initial proportion of species 57 in the community.

p58 A numeric vector indicating the initial proportion of species 58 in the community.

p59 A numeric vector indicating the initial proportion of species 59 in the community.

p60 A numeric vector indicating the initial proportion of species 60 in the community.

p61 A numeric vector indicating the initial proportion of species 61 in the community.

p62 A numeric vector indicating the initial proportion of species 62 in the community.

p63 A numeric vector indicating the initial proportion of species 63 in the community.

p64 A numeric vector indicating the initial proportion of species 64 in the community.

p65 A numeric vector indicating the initial proportion of species 65 in the community.

p66 A numeric vector indicating the initial proportion of species 66 in the community.

p67 A numeric vector indicating the initial proportion of species 67 in the community.

p68 A numeric vector indicating the initial proportion of species 68 in the community.

p69 A numeric vector indicating the initial proportion of species 69 in the community.

p70 A numeric vector indicating the initial proportion of species 70 in the community.

p71 A numeric vector indicating the initial proportion of species 71 in the community.

p72 A numeric vector indicating the initial proportion of species 72 in the community.

response A numeric vector giving the average daily respiration rate of the bacterial community.

Details

What are Diversity-Interactions (DI) models?

Diversity-Interactions (DI) models (Kirwan et al 2009) are a set of tools for analysing and interpreting data from experiments that explore the effects of species diversity on community-level responses. We strongly recommend that users read the short introduction to Diversity-Interactions models (available at: DImodels). Further information on Diversity-Interactions models is also available in Kirwan et al 2009 and Connolly et al 2013.

The Bell dataset is analysed using Diversity-Interactions models in both Brophy et al 2017 and Connolly et al 2013.

Source

Bell T, JA Newman, BW Silverman, SL Turner and AK Lilley (2005) The contribution of species richness and composition to bacterial services. Nature, 436, 1157-1160.

References

Brophy C, A Dooley, L Kirwan, JA Finn, J McDonnell, T Bell, MW Cadotte and J Connolly (2017) Biodiversity and ecosystem function: Making sense of numerous species interactions in multi-species communities. Ecology, 98, 1771-1778.

Connolly J, T Bell, T Bolger, C Brophy, T Carnus, JA Finn, L Kirwan, F Isbell, J Levine, A Lüscher, V Picasso, C Roscher, MT Sebastia, M Suter and A Weigelt (2013) An improved model to predict the effects of changing biodiversity levels on ecosystem function. Journal of Ecology, 101, 344-355.

Kirwan L, J Connolly, JA Finn, C Brophy, A Lüscher, D Nyfeler and MT Sebastia (2009) Diversity-interaction modelling - estimating contributions of species identities and interactions to ecosystem function. Ecology, 90, 2032-2038.

Examples

```
## Load the Bell data
  data(Bell)
## View the first five entries
  head(Bell)
## Explore the variabes in sim1
  str(Bell)

## Check that the proportions sum to 1 (required for DI models)
## p1 to p72 are in the 4th to 75th columns in Bell
  Bellsums <- rowSums(Bell[4:75])
  summary(Bellsums)

## Check characteristics of Bell
  hist(Bell$response)
  summary(Bell$response)
  plot(Bell$richness, Bell$response)

## Fit the average pairwise model using DI and the AV tag, with theta = 1
 m1 <- DI(y = "response", prop = 4:75, DImodel = "AV", estimate_theta = FALSE, data = Bell)
  summary(m1)


## This code takes around 11 seconds to run

## Fit the average pairwise model using DI and the AV tag, with theta estimated
 m2 <- DI(y = "response", prop = 4:75, DImodel = "AV", estimate_theta = TRUE, data = Bell)
  summary(m2)
  CI_95 <- theta_CI(m2, conf = .95)
  CI_95
  plot(m1)
  library(hnp)
  hnp(m1)
## Graph the profile likelihood
  library(ggplot2)
  ggplot(m2$profile_loglik, aes(x = grid, y = prof)) +
    theme_bw() +
    geom_line() +
```

```
    xlim(0,1.5) +
    xlab(expression(theta)) +
    ylab("Log-likelihood") +
    geom_vline(xintercept = CI_95, lty = 3) +
    labs(title = "   Log-likelihood versus theta",
        caption = "dotted vertical lines are upper and lower bounds of 95% CI for theta")
```

---

design_a *The "design_a" Dataset*

Description

This dataset contains a set of proportions p1 to p9 where each row sums to 1. It is used as the design matrix for simulating other datasets in the DImodels package.

Usage

```
data("design_a")
```

Format

A data frame with 206 observations on the following 11 variables:

- community A numeric vector identifying each unique community, i.e., two rows with the same community value also share the same set of p1 to p9 values.
- richness A numeric vector indicating the number of species in the initial composition, i.e., the number of proportions from p1 to p9 that are >0.
- p1 A numeric vector indicating a proportion (of species 1).
- p2 A numeric vector indicating a proportion (of species 2).
- p3 A numeric vector indicating a proportion (of species 3).
- p4 A numeric vector indicating a proportion (of species 4).
- p5 A numeric vector indicating a proportion (of species 5).
- p6 A numeric vector indicating a proportion (of species 6).
- p7 A numeric vector indicating a proportion (of species 7).
- p8 A numeric vector indicating a proportion (of species 8).
- p9 A numeric vector indicating a proportion (of species 9).

Details

The columns p1 to p9 form a simplex space.

Examples

```
## Load the design_a data
  data(design_a)
## View the first five entries
  head(design_a)
## Explore the variables in design_a
  str(design_a)

## Check that the proportions sum to 1 (required for DI models)
## p1 to p9 are in the 3rd to 11th columns in design_a
  design_a_sums <- rowSums(design_a[3:11])
  summary(design_a_sums)
```

---

design_b *The "design_b" Dataset*

Description

This dataset contains a set of proportions `p1` to `p6` where each row sums to 1. It is used as the design matrix for simulating other datasets in the `DImodels` package.

Usage

```
data("design_b")
```

Format

A data frame with 47 observations on the following seven variables:

`richness` A numeric vector indicating the number of species in the initial composition, i.e., the number of proportions from p1 to p6 that are >0.

`p1` A numeric vector indicating a proportion (of species 1).

`p2` A numeric vector indicating a proportion (of species 2).

`p3` A numeric vector indicating a proportion (of species 3).

`p4` A numeric vector indicating a proportion (of species 4).

`p5` A numeric vector indicating a proportion (of species 5).

`p6` A numeric vector indicating a proportion (of species 6).

Details

The columns p1 to p6 form a simplex space.

Examples

```
## Load the design_b data
  data(design_b)
## View the first five entries
  head(design_b)
## Explore the variables in design_b
  str(design_b)

## Check that the proportions sum to 1 (required for DI models)
## p1 to p6 are in the 2nd to 7th columns in design_b
  design_b_sums <- rowSums(design_b[2:7])
  summary(design_b_sums)
```

`DI` *Diversity-Interactions Model Fitting*

Description

This function will fit a wide range of Diversity-Interactions (DI) models, one at a time. It provides some assisted automated ways to fit DI models, and includes the flexibility to extend DI models in several directions.

Usage

```
DI(y, block, density, prop, treat, FG, DImodel, extra_formula, custom_formula,
  data, estimate_theta = FALSE)
```

Arguments

The minimum required arguments to use DI are either:

Argument `DImodel` with `data`, `y` and `prop`, or

- Argument `custom_formula` with `data`.

The `DImodel` argument allows fitting of DI models via a range of 'tag' options that determine the form of the species interactions terms (the tags, described below, are `STR`, `ID`, `AV`, `FG`, `ADD` and `FULL`) and extra terms can be added to the model using the `extra_formula` argument. Using the argument `custom_formula` requires full specification of the model to be fitted using standard `lm` or `glm` syntax.

The column name of the response vector, which must be in quotes, for example, `y = "yield"`.

`y` `block` The name of the block variable (if present), which must be in quotes, for example, `block = "block"`. If no blocking variable, omit this argument.

`density` The name of the density variable (if present), which must be in quotes, for example, `density = "density"`. If no density variable, omit this argument.

`prop` A vector of *s* column names identifying the species proportions in each community in the dataset. For example, if the species proportions columns are labelled p1 to p4, then `prop = c("p1","p2","p3","p4")`. Alternatively, the column numbers can be specified, for example, `prop = 4:7`, where species proportions are in the 4th to 7th columns.

`treat` The name of a column in the dataset containing the value of a treatment factor or covariate. The treatment name must be included in quotes, for example, `treat = "nitrogen"`. If the treatment is a factor, the variable must already be specified as a factor prior to using `DI`.

- When used in conjunction with `DImodel`, the treatment will be included in the model as an additive factor or covariate, for example, specifying `treat = nitrogen,DImodel = ID` will fit the model `p1 + p2 + ... + ps + nitrogen`. Additional treatments, or interactions between the treatment and other model terms can be included via the `extra_formula` argument.
- The `treat` argument is defunct when using the `custom_formula` argument, and any treatment must be included directly in the `custom_formula` argument.

`FG` If species are classified by *g* functional groups, this argument takes a text list (of length *s*) of the functional group to which each species belongs. For example, for four grassland species with two grasses and two legumes: FG could be `FG = c("G","G","L","L")`, where G stands for grass and L stands for legume.

- The `FG` argument is required if `DImodel = "FG"` is specified.
- The `FG` argument is defunct when using the `custom_formula` argument, since species interactions must be included directly in the `custom_formula` argument.

`DImodel` This argument is chosen (over `custom_formula`) to fit an automated version of a DI model. The chosen tag determines the form of the species interactions to be included in the model. The tags (or options) are:

- `STR` (no identity or interaction effects, only an intercept is fitted, plus the experiment structural variables `block`, `density` and `treat`, if specified).

Each of the following includes the species proportions as specified in `prop`, the interaction variables according to the tag, plus `block`, `density` and `treat` if specified.

- `ID` (no interaction terms),
- `AV` (a single average species pairwise interaction variable),
- `FG` (functional group interaction variables, the `FG` argument must be specified to use this option),
- `ADD` (the additive species interaction variables),
- `FULL` (all pairwise interaction variables).

The DImodel tag should appear in quotes, for example, `DImodel = "STR"`.

`extra_formula` In conjunction with `DImodel`, additional terms can be added using `extra_formula`. A ~ must be included before specifying the terms. For example, if `DImodel = "AV"` has been specified, adding `extra_formula = ~ I(AV**2)` will add a quadratic average pairwise interaction variable or `extra_formula = ~ treatment:AV` will add an interaction between the average pairwise species interaction variable and the treatment. Any variable included directly in `extra_formula` must already be contained in the dataset (interaction variables can be created using the functions in `DI_data_manipulation`, if required).

`custom_formula` To specify your own DI model, write your own model formula using the `custom_formula` argument. The standard notation from `lm` and `glm` is used here, for example, `custom_formula = yield ~ 0 + p1:treatment + p2:treatment + p3:treatment + p4:treatment` will fit the DI model with the identity effects each crossed with treatment, where treatment is a variable in the dataset. The `custom_formula` argument is recommended when the `DImodel` and `extra_formula` arguments are not sufficient. Any variable included directly in `custom_formula` must be already contained in the dataset (interaction variables can be created using the functions in `DI_data_manipulation`, if required).

`data` Specify the dataset, for example, `data = Switzerland`. The dataset name should not appear in quotes.

`estimate_theta` By default, theta (the power parameter on all $pi * pj$ components of each interaction variable in the model) is set equal to one. Specify `estimate_theta = TRUE` to include the estimation of θ in the specified model. The `estimate_theta` argument can only be used in conjunction with the `DImodel` argument; if the `custom_formula` is used, then theta estimation is not available and the default `estimate_theta = FALSE` must be used.

Details

What are Diversity-Interactions models?

Diversity-Interactions (DI) models (Kirwan et al 2009) are a set of tools for analysing and interpreting data from experiments that explore the effects of species diversity on community-level responses. We recommend that users of the `DImodels` package read the short introduction to DI models (available at: `DImodels`). Further information on DI models is available in Kirwan et al 2009 and Connolly et al 2013.

Checks on data prior to using `DI`.

Before using `DI`, check that the species proportions for each row in your dataset sum to one. See the 'Examples' section for code to do this. An error message will be generated if the proportions don't sum to one.

How does the `DI` function work?

The `DI` function provides wide flexibility in the types of Diversity-Interactions (DI) models that can be fitted. There are two ways to fit models in `DI`: 1) using `DImodel`, possibly augmented by `extra_formula`, or 2) using `custom_formula`. Models are estimated using iteratively reweighted least squares, via the `glm` package, when the option `estimate_theta = FALSE`.

Consider the following DI model, for example (in R formula syntax): `y ~ p1 + p2 + p3 + treatment + p1:p2 + p1:p3 + p2:p3 + p1:p2:treatment + p1:p3:treatment + p2:p3:treatment`

This model can be fitted using `DImodel` and `extra_formula`: `DI(y = "y",prop = c("p1","p2","p3"),treat = "nitrogen",DImodel = "FULL",extra_formula = ~ p1:p2:treatment + p1:p3:treatment + p2:p3:treatment,data = datasetname)` or, by specifying all of the terms in the model using `custom_formula`: `DI(custom_formula = y ~ p1 + p2 + p3 + treatment + p1:p2 + p1:p3 + p2:p3 + p1:p2:treatment + p1:p3:treatment + p2:p3:treatment,data = datasetname`

We recommend to use `DImodel` where possible, to augment with `extra_formula` where required, and to only use `custom_formula` when `DImodel` plus `extra_formula` is insufficient.

Including theta in DI models

A non-linear parameter θ can be included in DI models as a power on all $pi * pj$ components of each pairwise interaction variable. For example (in R formula syntax): `y ~ p1 + p2 + p3 + (p1:p2)^theta + (p1:p3)^theta + (p2:p3)^theta` for the full pairwise interaction model. Including θ alters the contribution of the interaction term to the response (Connolly et al 2013).

By default, the value of θ is 1. By specifying `estimate_theta = TRUE` within `DI`, a value of θ will be estimated using profile likelihood over the space $\theta = 0.01$ to 1.5. The option `estimate_theta = TRUE` can only be used with `DImodel`, it is not available when using `custom_formula`.

As a general guideline to testing if θ is required, we recommend:

1) finding the best form of the species interaction terms assuming theta = 1, and then,

2) testing if theta differs from 1.

If no species interaction terms are needed, then there is no need to do any testing for theta.

Value

A model object of class `"glm"` including the components detailed in `glm`, plus the following:

`DIcall`	the call of the `DI` function
`DIinternalcall`	an internal call made within the DI model fitting process

Author(s)

Rafael A. Moral, John Connolly and Caroline Brophy

References

Connolly J, T Bell, T Bolger, C Brophy, T Carnus, JA Finn, L Kirwan, F Isbell, J Levine, A Lüscher, V Picasso, C Roscher, MT Sebastia, M Suter and A Weigelt (2013) An improved model to predict the effects of changing biodiversity levels on ecosystem function. Journal of Ecology, 101, 344-355.

Kirwan L, J Connolly, JA Finn, C Brophy, A Lüscher, D Nyfeler and MT Sebastia (2009) Diversity-interaction modelling - estimating contributions of species identities and interactions to ecosystem function. Ecology, 90, 2032-2038.

See Also

`autoDI` `theta_CI`

Other examples using `DI`: The `Bell` dataset examples. The `sim1` dataset examples. The `sim2` dataset examples. The `sim3` dataset examples. The `sim4` dataset examples. The `sim5` dataset examples. The `Switzerland` dataset examples.

Examples

```
## Load the Switzerland data
  data(Switzerland)

## Check that the proportions sum to 1 (required for DI models)
## p1 to p4 are in the 4th to 7th columns in Switzerland
  Switzerlandsums <- rowSums(Switzerland[4:7])
  summary(Switzerlandsums)

## Fit the FG DImodel, with factors density and treatment, and with theta = 1
  m1 <- DI(y = "yield", density = "density", prop = 4:7, treat = "nitrogen",
           FG = c("G","G","L","L"), DImodel = "FG", data = Switzerland)
  summary(m1)

## Fit the FG DImodel, with factors density and treatment, and theta estimated
  m2 <- DI(y = "yield", density = "density", prop = 4:7, treat = "nitrogen",
        FG = c("G","G","L","L"), DImodel = "FG", data = Switzerland, estimate_theta = TRUE)
  summary(m2)

## Test if the identity effects interact with nitrogen (and main effect of nitrogen excluded)
 m3 <- DI(y = "yield", density = "density", prop = 4:7, FG = c("G", "G", "L", "L"), DImodel = "FG",
           extra_formula = ~ (p1 + p2 + p3 + p4):nitrogen, data = Switzerland)
  summary(m3)

## Fit the average pairwise model and check for a quadratic term using extra_formula.
## Need to create AV variable to be included in extra_formula and put in new dataset Switzerland2.
  newlist <- DI_data_E_AV(prop = c("p1","p2","p3","p4"), data = Switzerland)
  Switzerland2 <- data.frame(Switzerland, "AV" = newlist$AV)
  m4 <- DI(y = "yield", density = "density", prop = 4:7, DImodel = "AV",
           extra_formula = ~ I(AV**2), data = Switzerland2)
  summary(m4)

## Using the custom_formula option to fit some, but not all, of the FG interactions.
## Fit the FG DImodel using custom_formula, with factors density and treatment, and theta = 1.
## Need to create functional group interaction variables for inclusion in custom_formulaand put
##  in new dataset Switzerland3.
  newlist <- DI_data_FG(prop = 4:7, FG = c("G","G","L","L"), data = Switzerland)
  Switzerland3 <- data.frame(Switzerland, newlist$FG)
  m5 <- DI(y = "yield", prop = c("p1","p2","p3","p4"),
           custom_formula = yield ~ 0 + p1 + p2 + p3 + p4 + bfg_G_L + wfg_G
                           + density + nitrogen,
           data = Switzerland3)
  summary(m5)
```

DImodels-internal *Internal Functions*

Description

Internal functions used within the DI and autoDI functions.

Author(s)

Rafael A. Moral, John Connolly and Caroline Brophy

DI_compare *Compare a Fitted Diversity-Interactions Model to the Reference Model*

Description

This function fits the reference model internally and compares a DI model fit to it using anova.

Usage

```
DI_compare(model, ...)
```

Arguments

model	A DI model object.
...	Further arguments passed to anova.

Details

This function takes a DI model object as input, fits the reference model internally and compares the two models using anova. The reference model includes an additive effect of community (each unique combination of species proportions) in the linear predictor. For more details on the reference model, see Connolly et al. (2013).

Value

The function returns the reference model, a glm model object.

Author(s)

Rafael A. Moral, John Connolly and Caroline Brophy

References

Connolly J, T Bell, T Bolger, C Brophy, T Carnus, JA Finn, L Kirwan, F Isbell, J Levine, A Lüscher, V Picasso, C Roscher, MT Sebastia, M Suter and A Weigelt (2013) An improved model to predict the effects of changing biodiversity levels on ecosystem function. Journal of Ecology, 101, 344-355.

See Also

DI autoDI

Examples

```
## Load the sim1 data
  data(sim1)

## Fit the FULL model
  m1 <- DI(y = "response", block = "block", prop = 3:6,
           DImodel = "FULL", data = sim1)

## Compare with the reference model
  DI_compare(m1, test = "F")
```

DI_data_manipulation *Data manipulation functions*

Description

The following five functions compute additional variables for the various types of interactions among pairs of species proportions. These variables are defined following Kirwan et al 2007 and 2009, and Connolly et al 2013.

DI_data_E_AV: creates the average pairwise interaction variable (AV) and a scaled version of it (E). The variable AV is the sum of products of the proportions of each pair of species in the mixture. The variable E is a scaled version of AV that ranges between 0 (for a monoculture community) to 1 for the equi-proportional mixture of all species in the pool.

DI_data_FG: creates the functional group interaction variables. There is a variable for (within) each functional group and one for (between) each pair of functional groups, i.e., if there are two functional groups, there will be three functional group interaction variables, while if there are three functional groups, there will be six functional group interaction variables.

DI_data_ADD: creates the additive species interaction variables, one for each species.

DI_data_prepare: computes all types of interactions in the preceding three functions. Use this function to implement all three previous functions in one step (avoiding the need to use the individual ones).

DI_data_fullpairwise: computes all individual pairwise interactions. There will be $s*(s-1)/2$ new variables created, where s is the number of species in the pool.

By default, the interaction variables described above are created with theta = 1, but a different value of theta can also be specified (Connolly et al 2013).

Usage

```
DI_data_E_AV(prop, data, theta = 1)
DI_data_FG(prop, FG, data, theta = 1)
DI_data_ADD(prop, data, theta = 1)
DI_data_prepare(y, block, density, prop, treat, FG = NULL, data, theta = 1)
DI_data_fullpairwise(prop, data, theta = 1)
```

Arguments

prop
: A vector of column names identifying the species proportions in the dataset. For example, if the species proportions columns are labelled p1 to p4, then `prop = c("p1","p2","p3","p4")`. The column numbers in which the proportions are stored can also be referred to, for example, `prop = 4:7` for the `Switzerland` data.

FG
: If species are classified by g functional groups, this argument takes a text list (of length s) of the functional group to which each species belongs. For example, for four grassland species with two grasses and two legumes, it could be `FG = c("G","G","L","L")`, where G stands for grass and L stands for legume. This argument is required in `DI_data_FG`. This argument is not required in `DI_data_prepare`, but if omitted, the functional group interactions will not be computed by the function.

data
: Specify the dataset, for example, `data = Switzerland`. The dataset name should not appear in quotes.

theta
: Interaction variables will be computed with the theta power, equal to the value specified, on all $pi * pj$ components of each interaction variable, with default value one. For example, with three species `AV = p1*p2 + p1*p3 + p2*p3` and if computed with `theta` = 0.5, this becomes `(p1*p2)^0.5 + (p1*p3)^0.5 + (p2*p3)^0.5`.

y
: The column name of the response vector, for example, `y = "yield"`. The name of the response variable must be contained in quotes. This argument is not required (but is used internally in the `DI` and `autoDI` functions). If this argument is omitted, there are no implications for the calculation of interaction variables here.

block
: The column name of the block variable. This argument is not required (but is used internally in the `DI` and `autoDI` functions). If there is no blocking variable, omit this argument; in this case, a new variable `block_zero`, which is a column of zeros, will be computed. This column of zeros is used internally in the `DI` and `autoDI` functions, but has no implications the calculation of interaction variables here.

density
: The column name of the density variable. This argument is not required (but is used internally in the `DI` and `autoDI` functions). If there is no density variable, omit this argument; in this case, a new variable `density_zero`, which is a column of zeros, will be computed. This column of zeros is used internally in the `DI` and `autoDI` functions, but has no implications for the calculation of interaction variables here.

treat
: The column name of the treatment variable. This argument is not required (but is used internally in the `DI` and `autoDI` functions). If there is no treatment variable, omit this argument; in this case, a new variable `treat_zero`, which is a column of zeros, will be computed. This column of zeros is used internally in the `DI` and `autoDI` functions, but has no implications for the calculation of interaction variables here.

Details

What are Diversity-Interactions models?

Diversity-Interactions (DI) models (Kirwan et al 2009) are a set of tools for analysing and interpreting data from experiments that explore the effects of species diversity on community-level responses. We recommend that users of the `DImodels` package read the short introduction to DI models (available at: `DImodels`). Further information on DI models is available in Kirwan et al 2009 and Connolly et al 2013.

Checks on data prior to using the data manipulation functions.

Before applying the data manipulation functions to your dataset, check that the species proportions in each row sum to one. See the 'Examples' section for code to do this. An error message will be generated if the proportions don't sum to one.

When are the data manipulation functions needed?

It is not required to use the data manipulation functions if using the `autoDI` function, or the `DImodel` option in the `DI` function, as they will automatically create the species interaction variables needed. If using species interaction variables in the `extra_formula` or `custom_formula` options in `DI`, then it is required to have the variables already in the dataset and these functions can do that.

Short worked example to illustrate how the data manipulation functions work

The code to implement this example is provided in the 'Examples' section.

Assume four species with initial proportions in two communities: (0.1, 0.2, 0.3, 0.4) and (0.25, 0.25, 0.25, 0.25), with `FG = c("G","G","L","L")`.

For community 1: (0.1,0.2,0.3,0.4), assuming theta = 1, the data preparation functions will compute the following additional variables (details in Kirwan et al 2007 and 2009):

AV = 0.1*0.2 + 0.1*0.3 + 0.1*0.4 + 0.2*0.3 + 0.2*0.4 + 0.3*0.4 = 0.35

E = *(2s/(s-1))**AV = 0.9333

p1_add = 0.1 * (1 - 0.1) = 0.09

p2_add = 0.2 * (1 - 0.2) = 0.16

p3_add = 0.3 * (1 - 0.3) = 0.21

p4_add = 0.4 * (1 - 0.4) = 0.24

bfg_G_L = 0.1*0.3 + 0.1*0.4 + 0.2*0.3 + 0.2*0.4 = 0.21

wfg_G = 0.1*0.2 = 0.02

wfg_L = 0.3*0.4 = 0.12

For community 1: (0.1,0.2,0.3,0.4), assuming theta = 0.5, the data preparation functions will compute the follow additional variables (details in Connolly et al 2013):

AV = (0.1*0.2)^0.5 + (0.1*0.3)^0.5 + (0.1*0.4)^0.5 + (0.2*0.3)^0.5 + (0.2*0.4)^0.5 + (0.3*0.4)^0.5 = 1.3888

E =*(2s/(s-1))**AV = 3.7035

p1_add = 0.1^0.5 * (0.2^0.5 + 0.3^0.5 + 0.4^0.5) = 0.5146

p2_add = 0.2^0.5 * (0.1^0.5 + 0.3^0.5 + 0.4^0.5) = 0.6692

p3_add = 0.3^0.5 * (0.1^0.5 + 0.2^0.5 + 0.4^0.5) = 0.7646

p4_add = 0.4^0.5 * (0.1^0.5 + 0.2^0.5 + 0.3^0.5) = 0.8293

bfg_G_L = (0.1*0.3)^0.5 + (0.1*0.4)^0.5 + (0.2*0.3)^0.5 + (0.2*0.4)^0.5 = 0.9010

wfg_G = (0.1*0.2)^0.5 = 0.1414

wfg_L = (0.3*0.4)^0.5 = 0.3464

When using the data manipulation functions to create interactions for theta values for a value different from 1, it is recommended to rename the new interaction variables to include `_theta`.

The data manipulation values for community 2 can be seen when the 'Examples' section code is run.

Value

The `DI_data_prepare` function returns a named list with the following components:

`newdata`	a `data.frame` containing all manipulated variables
`y`	the response variable name
`block`	the block variable name
`density`	the density variable name
`prop`	the species proportions variable names
`treat`	the treatment variable name
`FG`	the variables used in the FG model
`P_int_flag`	a logical value used internally
`even_flag`	a logical value used internally
`nSpecies`	the number of species in the design

The `DI_data_E_AV`, `DI_data_FG` and `DI_data_ADD` functions return a named list including one or more of the components below (depending on the data manipulation function):

`AV`	the AV variable
`E`	the E variable
`ADD_vars`	the variables used in the ADD model
`ADD_theta`	the variables used in the ADD model when theta is estimated
`FG`	the variables used in the FG model
`even_flag`	a logical value used internally
`P_int_flag`	a logical value used internally

The `DI_data_fullpairwise` function returns a matrix with the pairwise interactions between species.

Author(s)

Rafael A. Moral, John Connolly and Caroline Brophy

References

Connolly J, T Bell, T Bolger, C Brophy, T Carnus, JA Finn, L Kirwan, F Isbell, J Levine, A Lüscher, V Picasso, C Roscher, MT Sebastia, M Suter and A Weigelt (2013) An improved model to predict the effects of changing biodiversity levels on ecosystem function. Journal of Ecology, 101, 344-355.

Kirwan L, A Lüscher, MT Sebastia, JA Finn, RP Collins, C Porqueddu, A Helgadottir, OH Baadshaug, C Brophy, C Coran, S Dalmannsdottir, I Delgado, A Elgersma, M Fothergill, BE Frankow-Lindberg, P Golinski, P Grieu, AM Gustavsson, M Höglind, O Huguenin-Elie, C Iliadis, M Jørgensen, Z Kadziuliene, T Karyotis, T Lunnan, M Malengier, S Maltoni, V Meyer, D Nyfeler, P Nykanen-Kurki, J Parente, HJ Smit, U Thumm, & J Connolly (2007) Evenness drives consistent diversity effects in intensive grassland systems across 28 European sites. Journal of Ecology, 95, 530-539.

Kirwan L, J Connolly, JA Finn, C Brophy, A Lüscher, D Nyfeler and MT Sebastia (2009) Diversity-interaction modelling - estimating contributions of species identities and interactions to ecosystem function. Ecology, 90, 2032-2038.

See Also

DI autoDI

Other examples using the data manipulation functions: The Bell dataset examples. The sim2 dataset examples. The sim3 dataset examples. The sim4 dataset examples. The sim5 dataset examples. The Switzerland dataset examples.

Examples

```
#################################

#### Data manipulation for the Switzerland dataset

## Load the Switzerland data
  data(Switzerland)

## Check that the proportions sum to 1 (required for DI models)
## p1 to p4 are in the 4th to 7th columns in Switzerland
  Switzerlandsums <- rowSums(Switzerland[4:7])
  summary(Switzerlandsums)


## Create new interaction variables and incorporate them into a new data frame Switzerland2.
## Switzerland2 will contain the new variables:  AV, E, p1_add, p2_add, p3_add, p4_add,
##  bfg_G_L, wfg_G and wfg_L.
  newlist <- DI_data_prepare(prop = c("p1","p2","p3","p4"), FG = c("G","G","L","L"),
                             data = Switzerland)
  Switzerland2 <- data.frame(newlist$newdata, newlist$FG)


## Create new interaction variables and incorporate them into a new data frame Switzerland3.
## Use theta = 0.5.
  newlist <- DI_data_prepare(prop = c("p1","p2","p3","p4"), FG = c("G","G","L","L"),
                             data = Switzerland, theta = 0.5)
  Switzerland3 <- data.frame(newlist$newdata, newlist$FG)
## Add "_theta" to the new interaction variables to differentiate from when theta = 1
  names(Switzerland3)[12:20] <- paste0(names(Switzerland3)[12:20], "_theta")


#### The various interactions can also be added to a new dataset individually:
```

```
## Create the average pairwise interaction and evenness variables
##  and store them in a new data frame called Switzerland4.
## Switzerland4 will contain the new variables: AV, E
  newlist <- DI_data_E_AV(prop = c("p1","p2","p3","p4"), data = Switzerland)
  Switzerland4 <- data.frame(Switzerland, "AV" = newlist$AV, "E" = newlist$E)

## Create the functional group variables and add them to Switzerland4.
## In the FG names vector: G stands for grass, L stands for legume.
## Switzerland4 will contain: bfg_G_L, wfg_G and wfg_L
  newlist <- DI_data_FG(prop = 4:7, FG = c("G","G","L","L"), data = Switzerland)
  Switzerland4 <- data.frame(Switzerland4, newlist$FG)

## Create the additive species variables and add them to Switzerland4.
## Switzerland4 will contain the new variables: p1_add, p2_add, p3_add and p4_add.
  newlist <- DI_data_ADD(prop = c("p1","p2","p3","p4"), data = Switzerland)
  Switzerland4 <- data.frame(Switzerland4, newlist$ADD)

## Create all pairwise interaction variables and add them to Switzerland4.
## Switzerland5 will contain the new variables: p1.p2, p1.p3, p1.p4, p2.p3, p2.p4, p3.p4.
  newlist <- DI_data_fullpairwise(prop = c("p1","p2","p3","p4"), data = Switzerland)
  Switzerland4 <- data.frame(Switzerland4, newlist)

################################


################################

#### Short worked example (as illustrated the Details section)

## Create a dataframe
  p1 <- c(0.1, 0.25)
  p2 <- c(0.2, 0.25)
  p3 <- c(0.3, 0.25)
  p4 <- c(0.4, 0.25)
  minidataset1 <- data.frame(p1,p2,p3,p4)

## Check the rows sum to 1
  rowSums(minidataset1[1:4])

## Create the interaction variables, assume two functional groups and theta = 1
  newlist <- DI_data_prepare(prop = c("p1","p2","p3","p4"), FG = c("G","G","L","L"),
                             data = minidataset1)
  minidataset2 <- data.frame(newlist$newdata, newlist$FG)

## Create the interaction variables, assume two functional groups and theta = 0.5
  newlist <- DI_data_prepare(prop = c("p1","p2","p3","p4"), FG = c("G","G","L","L"),
                             y = "response", data = minidataset1, theta = 0.5)
  minidataset3 <- data.frame(newlist$newdata, newlist$FG, check.names = FALSE)
## Add "_theta" to the new interaction variables to differentiate from when theta = 1
  names(minidataset3)[8:16] <- paste0(names(minidataset3)[8:16], "_theta")

################################
```

sim1 *The Simulated "sim1" Dataset*

Description

The `sim1` dataset was simulated. There are four blocks and four species that vary in proportions (`p1-p4`). There are 15 unique sets of proportions identified by the variable `community`. Each unique community appears once in each block. The response was simulated assuming that there were species identity effects and block effects, but no diversity effects.

Usage

```
data(sim1)
```

Format

A data frame with 60 observations on the following seven variables:

- `community` A numeric vector identifying each unique community, i.e., two rows with the same community value also share the same set of p1 to p4 values.
- `block` A factor taking values 1 to 4 indicating block membership.
- `p1` A numeric vector indicating the initial proportion of species 1.
- `p2` A numeric vector indicating the initial proportion of species 2.
- `p3` A numeric vector indicating the initial proportion of species 3.
- `p4` A numeric vector indicating the initial proportion of species 4.
- `response` A numeric vector giving the simulated response variable.

Details

What are Diversity-Interactions (DI) models?

Diversity-Interactions (DI) models (Kirwan et al 2009) are a set of tools for analysing and interpreting data from experiments that explore the effects of species diversity on community-level responses. We strongly recommend that users read the short introduction to Diversity-Interactions models (available at: `DImodels`). Further information on Diversity-Interactions models is also available in Kirwan et al 2009 and Connolly et al 2013.

Parameter values for the simulation

DI models take the general form of:

$$y = Identities + Interactions + Structures + \epsilon$$

where *y* is a community-level response, the *Identities* are the effects of species identities and enter the model as individual species proportions at the beginning of the time period, the *Interactions* are the interactions among the species proportions, while *Structures* include other experimental structures such as blocks, treatments or density.

The dataset `sim1` was simulated with:

- identity effects for the four species with values = 10, 9, 8, 7
- block effects for the four blocks with values = 1, 1.5, 2, 0
- no interaction effects
- ϵ assumed normally distributed with mean 0 and standard deviation 1.1.

References

Connolly J, T Bell, T Bolger, C Brophy, T Carnus, JA Finn, L Kirwan, F Isbell, J Levine, A Lüscher, V Picasso, C Roscher, MT Sebastia, M Suter and A Weigelt (2013) An improved model to predict the effects of changing biodiversity levels on ecosystem function. Journal of Ecology, 101, 344-355.

Kirwan L, J Connolly, JA Finn, C Brophy, A Lüscher, D Nyfeler and MT Sebastia (2009) Diversity-interaction modelling - estimating contributions of species identities and interactions to ecosystem function. Ecology, 90, 2032-2038.

Examples

```
####################################
## Code to simulate the sim1 dataset



## Simulate dataset sim1 with species identity effects and block effects, but no interaction effect

## Use the proportions from the first fifteen plots in Switzerland
  data(Switzerland)

## Repeat the 15 plots over four blocks.
## Give each community type a unique (community) number.
  sim1 <- data.frame(community = rep(1:15, each = 4),
                    block = factor(rep(1:4, times = 15)),
                    p1 = rep(Switzerland$p1[1:15], each = 4),
                    p2 = rep(Switzerland$p2[1:15], each = 4),
                    p3 = rep(Switzerland$p3[1:15], each = 4),
                    p4 = rep(Switzerland$p4[1:15], each = 4))

## To simulate the response, first create a matrix of predictors that includes
##  p1-p4 and the four block dummy variables.
  X <- model.matrix(~ p1 + p2 + p3 + p4 + block -1, data = sim1)

## Create a vector of 'known' parameter values for simulating the response.
## The first four are the p1-p4 parameters, the second four are the block effects.
  sim1_coeff <- c(10,9,8,7,   1,1.5,2,0)

## Create response and add normally distributed error
  sim1$response <- as.numeric(X %*% sim1_coeff)
  set.seed(2020)
  r <- rnorm(n = 60, mean = 0, sd = 1.1)
  sim1$response <- round(sim1$response + r, digits = 3)
```

```
###########################
## Analyse the sim1 dataset

## Load the sim1 data
  data(sim1)
## View the first five entries
  head(sim1)
## Explore the variables in sim1
  str(sim1)

## Check that the proportions sum to 1 (required for DI models)
## p1 to p4 are in the 3rd to 6th columns in sim1
  sim1sums <- rowSums(sim1[3:6])
  summary(sim1sums)

## Check characteristics of sim1
  hist(sim1$response)
  summary(sim1$response)
  plot(sim1$p1, sim1$response)
  plot(sim1$p2, sim1$response)
  plot(sim1$p3, sim1$response)
  plot(sim1$p4, sim1$response)

## Find the best DI model using autoDI and F-test selection
  auto1 <- autoDI(y = "response", prop = c("p1","p2","p3","p4"), block = "block", data = sim1,
                  selection = "Ftest")
  summary(auto1)

## Fit the identity model using DI and the ID tag
  m1 <- DI(y = "response", prop = c("p1","p2","p3","p4"), block = "block", DImodel = "ID",
           data = sim1)
  summary(m1)
  plot(m1)


## Check goodness-of-fit using a half-normal plot with a simulated envelope
  library(hnp)
  hnp(m1)


## Fit the identity model using DI and custom_formula
  m2 <- DI(y = "response", custom_formula = response ~ 0 + p1 + p2 + p3 + p4 + block, data = sim1)
  summary(m2)
```

---

sim2 *The Simulated "sim2" Dataset*

Description

The `sim2` dataset was simulated. There are four blocks and four species that vary in proportions (`p1` `-p4`). There are 15 unique sets of proportions identified by the variable `community`. Each unique community appears once in each block. The response was simulated assuming that there were species identity effects, block effects, an average pairwise interaction effect and a theta value of 0.5.

Usage

```
data(sim2)
```

Format

A data frame with 60 observations on the following seven variables:

`community` A numeric vector identifying each unique community, i.e., two rows with the same community value also share the same set of p1 to p4 values.

`block` A factor taking values 1 to 4 indicating block membership.

`p1` A numeric vector indicating the initial proportion of species 1.

`p2` A numeric vector indicating the initial proportion of species 2.

`p3` A numeric vector indicating the initial proportion of species 3.

`p4` A numeric vector indicating the initial proportion of species 4.

`response` A numeric vector giving the simulated response variable.

Details

What are Diversity-Interactions (DI) models?

Diversity-Interactions (DI) models (Kirwan et al 2009) are a set of tools for analysing and interpreting data from experiments that explore the effects of species diversity on community-level responses. We strongly recommend that users read the short introduction to Diversity-Interactions models (available at: `DImodels`). Further information on Diversity-Interactions models is also available in Kirwan et al 2009 and Connolly et al 2013.

Parameter values for the simulation

DI models take the general form of:

$$y = Identities + Interactions + Structures + \epsilon$$

where *y* is a community-level response, the *Identities* are the effects of species identities and enter the model as individual species proportions at the beginning of the time period, the *Interactions* are the interactions among the species proportions, while *Structures* include other experimental structures such as blocks, treatments or density.

The dataset `sim2` was simulated with:

- identity effects for the four species with values = 10, 9, 8, 7
- block effects for the four blocks with values = 1, 1.5, 2, 0
- an average pairwise interaction effect = 8

- theta = 0.5 (where θ is a non-linear parameter included as a power on each $pipj$ product within interaction variables, see Connolly et al 2013 for details)
- ϵ assumed normally distributed with mean 0 and standard deviation 1.1.

References

Connolly J, T Bell, T Bolger, C Brophy, T Carnus, JA Finn, L Kirwan, F Isbell, J Levine, A Lüscher, V Picasso, C Roscher, MT Sebastia, M Suter and A Weigelt (2013) An improved model to predict the effects of changing biodiversity levels on ecosystem function. Journal of Ecology, 101, 344-355.

Kirwan L, J Connolly, JA Finn, C Brophy, A Lüscher, D Nyfeler and MT Sebastia (2009) Diversity-interaction modelling - estimating contributions of species identities and interactions to ecosystem function. Ecology, 90, 2032-2038.

Examples

```
####################################
## Code to simulate the sim2 dataset



## Simulate dataset sim2 with species identity effects, block effects and
##  an average pairwise interaction effect with theta=0.5.

## Use the proportions from the first fifteen plots in Switzerland
  data(Switzerland)

## Repeat the 15 plots over four blocks.
## Give each community type a unique (community) number.
  sim2 <- data.frame(community = rep(1:15, each = 4),
                   block = factor(rep(1:4, times = 15)),
                   p1 = rep(Switzerland$p1[1:15], each = 4),
                   p2 = rep(Switzerland$p2[1:15], each = 4),
                   p3 = rep(Switzerland$p3[1:15], each = 4),
                   p4 = rep(Switzerland$p4[1:15], each = 4))

## Create the average pairwise interaction variable, with theta = 0.5
  newlist <- DI_data_E_AV(prop=c("p1","p2","p3","p4"), data = sim2, theta = 0.5)
  sim2 <- data.frame(sim2, "AV_theta" = newlist$AV)

## To simulate the response, first create a matrix of predictors that includes p1-p4 and
##  the four block variables and the average pairwise interaction variable with theta=0.5.
  X <- model.matrix(~ p1 + p2 + p3 + p4 + block + AV_theta -1, data = sim2)

## Create a vector of 'known' parameter values for simulating the response.
## The first four are the p1-p4 parameters, the second four are the block effects and
##  the last one is the interaction parameter.
  sim2_coeff <- c(10,9,8,7,   1,1.5,2,0,    8)

## Create response and add normally distributed error
  sim2$response <- as.numeric(X %*% sim2_coeff)
```

```
set.seed(328781)
r <- rnorm(n = 60, mean = 0, sd = 1.1)
sim2$response <- round(sim2$response + r, digits = 3)
sim2$AV_theta <- NULL



####################################################################################################
####################################################################################################
## sim2

###########################
## Analyse the sim2 dataset

## Load the sim2 data
  data(sim2)
## View the first five entries
  head(sim2)
## Explore the variables in sim2
  str(sim2)

## Check that the proportions sum to 1 (required for DI models)
## p1 to p4 are in the 3rd to 6th columns in sim2
  sim2sums <- rowSums(sim2[3:6])
  summary(sim2sums)

## Check characteristics of sim2
  hist(sim2$response)
  summary(sim2$response)
  plot(sim2$p1, sim2$response)
  plot(sim2$p2, sim2$response)
  plot(sim2$p3, sim2$response)
  plot(sim2$p4, sim2$response)

## Find the best DI model using autoDI and F-test selection
 auto1 <- autoDI(y = "response", prop = c("p1", "p2", "p3", "p4"), block = "block", data = sim2,
                 selection = "Ftest")
  summary(auto1)

  ## Fit the average pairwise model, including theta, using DI and the AV tag
  m1 <- DI(y = "response", prop = c("p1","p2","p3","p4"), block = "block", DImodel = "AV",
           estimate_theta = TRUE, data = sim2)
  summary(m1)
  CI_95 <- theta_CI(m1, conf = .95)
  CI_95
  plot(m1)
  library(hnp)


## Check goodness-of-fit using a half-normal plot with a simulated envelope
  library(hnp)
  hnp(m1)
```

```
## Graph the profile likelihood
  library(ggplot2)
  ggplot(m1$profile_loglik, aes(x = grid, y = prof)) +
    theme_bw() +
    geom_line() +
    xlim(0,1.5) +
    xlab(expression(theta)) +
    ylab("Log-likelihood") +
    geom_vline(xintercept = CI_95, lty = 3) +
    labs(title = "   Log-likelihood versus theta",
      caption = "dotted vertical lines are upper and lower bounds of 95% CI for theta")

## Fit the average pairwise model, including theta, using DI and custom_formula
## A value of theta must be 'chosen'. Take: 0.4533437 from m1. The 'estimate_theta' option is not
##  available with custom_formula.
  newlist <- DI_data_E_AV(prop = c(3:6), data = sim2, theta = 0.4533437)
  sim2a <- data.frame(sim2, "AV_theta" = newlist$AV)
 m2 <- DI(y = "response", custom_formula = response ~ 0 + p1 + p2 + p3 + p4 + AV_theta + block,
          data = sim2a)
## This will adjust the standard errors in m2 for the 'estimation' of theta
  m2$df.residual <- m2$df.residual - 1
  summary(m2)
## This will adjust the AIC in m2 for the 'estimation' of theta
  m2$aic <- m2$aic + 2
  summary(m2)
```

---

sim3 *The Simulated "sim3" Dataset*

Description

The `sim3` dataset was simulated. There are two treatments and nine species that vary in proportions (`p1` -`p9`). It is assumed that species 1 to 5 come from functional group 1, species 6 and 7 from functional group 2 and species 8 and 9 from functional group 3. The response was simulated assuming that there were species identity effects and functional group specific interaction effects.

Usage

```
data(sim3)
```

Format

A data frame with 412 observations on the following 13 variables:

`community` A numeric vector identifying each unique community, i.e., two rows with the same community value also share the same set of p1 to p9 values.

`richness` A numeric vector identifying the number of species in the initial composition.

`treatment` A factor with levels `A` or `B`.

`p1` A numeric vector indicating the initial proportion of species 1.

`p2` A numeric vector indicating the initial proportion of species 2.

`p3` A numeric vector indicating the initial proportion of species 3.

`p4` A numeric vector indicating the initial proportion of species 4.

`p5` A numeric vector indicating the initial proportion of species 5.

`p6` A numeric vector indicating the initial proportion of species 6.

`p7` A numeric vector indicating the initial proportion of species 7.

`p8` A numeric vector indicating the initial proportion of species 8.

`p9` A numeric vector indicating the initial proportion of species 9.

`response` A numeric vector giving the simulated response variable.

Details

What are Diversity-Interactions (DI) models?

Diversity-Interactions (DI) models (Kirwan et al 2009) are a set of tools for analysing and interpreting data from experiments that explore the effects of species diversity on community-level responses. We strongly recommend that users read the short introduction to Diversity-Interactions models (available at: `DImodels`). Further information on Diversity-Interactions models is also available in Kirwan et al 2009 and Connolly et al 2013.

Parameter values for the simulation

DI models take the general form of:

$$y = Identities + Interactions + Structures + \epsilon$$

where y is a community-level response, the *Identities* are the effects of species identities and enter the model as individual species proportions at the beginning of the time period, the *Interactions* are the interactions among the species proportions, while *Structures* include other experimental structures such as blocks, treatments or density.

The dataset `sim3` was simulated with:

- identity effects for the nine species with values = 10, 9, 8, 7, 11, 6, 5, 8, 9
- treatment effects = 3, 0
- functional group specific interact effects; assume functional groups are labelled FG1, FG2 and FG3, then the interaction parameter values are: between FG1 and FG2 = 4, between FG1 and FG3 = 9, between FG2 and FG3 = 3, within FG1 = 2, within FG2 = 3 and within FG3 = 1
- theta = 1 (where θ is a non-linear parameter included as a power on each $pipj$ product within interaction variables, see Connolly et al 2013 for details)
- ϵ assumed normally distributed with mean 0 and standard deviation 1.2.

References

Connolly J, T Bell, T Bolger, C Brophy, T Carnus, JA Finn, L Kirwan, F Isbell, J Levine, A Lüscher, V Picasso, C Roscher, MT Sebastia, M Suter and A Weigelt (2013) An improved model to predict the effects of changing biodiversity levels on ecosystem function. Journal of Ecology, 101, 344-355.

Kirwan L, J Connolly, JA Finn, C Brophy, A Lüscher, D Nyfeler and MT Sebastia (2009) Diversity-interaction modelling - estimating contributions of species identities and interactions to ecosystem function. Ecology, 90, 2032-2038.

Examples

```
####################################
## Code to simulate the sim3 dataset



## Simulate dataset sim3 with 9 species, three functional groups and two levels of a treatment.
## The species 1-5 are FG1, species 6-7 are FG2 and species 8-9 are FG3.
## Assume ID effects and the FG interaction model, with a treatment (factor with two levels).

## Set up proportions
  data("design_a")
  sim3a <- design_a

# Replicate the design over two treatments
  sim3b <- sim3a[rep(seq_len(nrow(sim3a)), each = 2), ]
  sim3c <- data.frame(treatment = factor(rep(c("A","B"), times = 206)))
  sim3 <- data.frame(sim3b[,1:2], sim3c, sim3b[,3:11])
  row.names(sim3) <- NULL

## Create the functional group interaction variables
 newlist <- DI_data_FG(prop = 4:12, FG = c("FG1","FG1","FG1","FG1","FG1","FG2","FG2","FG3","FG3"),
                        data = sim3)
  sim3 <- data.frame(sim3, newlist$FG)

## To simulate the response, first create a matrix of predictors that includes p1-p9, the treatment
##  and the interaction variables.
  X <- model.matrix(~ p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9 + treatment
                    + bfg_FG1_FG2 + bfg_FG1_FG3 + bfg_FG2_FG3
                    + wfg_FG1 + wfg_FG2 + wfg_FG3 -1, data=sim3)

## Create a vector of 'known' parameter values for simulating the response.
## The first nine are the p1-p9 parameters, and the second set of two are the treatment effects
##   and the third set of six are the interaction parameters.
  sim3_coeff <- c(10,9,8,7,11, 6,5, 8,9,   3,0,      4,9,3, 2,3,1)

## Create response and add normally distributed error
  sim3$response <- as.numeric(X %*% sim3_coeff)
  set.seed(1657914)
  r <- rnorm(n = 412, mean = 0, sd = 1.2)
  sim3$response <- round(sim3$response + r, digits = 3)
```

```
  sim3[,13:18] <- NULL



###########################
## Analyse the sim3 dataset

## Load the sim3 data
  data(sim3)
## View the first five entries
  head(sim3)
## Explore the variables in sim3
  str(sim3)

## Check characteristics of sim3
  hist(sim3$response)
  summary(sim3$response)
  plot(sim3$richness, sim3$response)
  plot(sim3$p1, sim3$response)
  plot(sim3$p2, sim3$response)
  plot(sim3$p3, sim3$response)
  plot(sim3$p4, sim3$response)
  plot(sim3$p5, sim3$response)
  plot(sim3$p6, sim3$response)
  plot(sim3$p7, sim3$response)
  plot(sim3$p8, sim3$response)
  plot(sim3$p9, sim3$response)


## What model fits best? Selection using F-test in autoDI
  auto1 <- autoDI(y = "response", prop = 4:12, treat = "treatment",
             FG = c("FG1","FG1","FG1","FG1","FG1","FG2","FG2","FG3","FG3"), data = sim3,
                 selection = "Ftest")
  summary(auto1)


## Fit the functional group model, with treatment, using DI and the FG tag
  m1 <- DI(y = "response", prop = 4:12,
       FG = c("FG1","FG1","FG1","FG1","FG1","FG2","FG2","FG3","FG3"), treat = "treatment",
           DImodel = "FG", data = sim3)
  summary(m1)
  plot(m1)


## Check goodness-of-fit using a half-normal plot with a simulated envelope
  library(hnp)
  hnp(m1)


## Create the functional group interactions and store them in a new dataset
 newlist <- DI_data_FG(prop = 4:12, FG = c("FG1","FG1","FG1","FG1","FG1","FG2","FG2","FG3","FG3"),
```

```
                      data = sim3)
  sim3a <- data.frame(sim3, newlist$FG)

## Test if the FG interaction variables interact with treatment using 'extra_formula'
  m2 <- DI(y = "response", prop = 4:12,
           FG = c("FG1","FG1","FG1","FG1","FG1","FG2","FG2","FG3","FG3"),
           treat = "treatment", DImodel = "FG", extra_formula = ~ bfg_FG1_FG2:treatment
       + bfg_FG1_FG3:treatment + bfg_FG2_FG3:treatment + wfg_FG1:treatment + wfg_FG2:treatment
           + wfg_FG3:treatment, data = sim3a)
  summary(m2)

## Fit the functional group model using DI and custom_formula
## Set up a dummy variable for treatment first (required).
  sim3a$treatmentA <- as.numeric(sim3a$treatment=="A")
 m3 <- DI(y = "response", custom_formula = response ~ 0 + p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8
          + p9 + treatmentA + bfg_FG1_FG2 + bfg_FG1_FG3 + bfg_FG2_FG3 + wfg_FG1 + wfg_FG2
           + wfg_FG3, data = sim3a)
  summary(m3)
```

---

sim4 *The Simulated "sim4" Dataset*

Description

The sim4 dataset was simulated. There is a covariate treatment and six species that vary in proportions (p1 -p6). It is assumed that species 1 and 2 come from functional group 1, species 3 and 4 from functional group 2 and species 5 and 6 from functional group 3. The response was simulated assuming that there were species identity effects, separate pairwise interaction effects and a covariate effect.

Usage

```
data(sim4)
```

Format

A data frame with 141 observations on the following nine variables:

- `richness` A numeric vector identifying the number of species in the initial composition.
- `treatment` A covariate taking values 50, 150 or 250.
- `p1` A numeric vector indicating the initial proportion of species 1.
- `p2` A numeric vector indicating the initial proportion of species 2.
- `p3` A numeric vector indicating the initial proportion of species 3.
- `p4` A numeric vector indicating the initial proportion of species 4.
- `p5` A numeric vector indicating the initial proportion of species 5.
- `p6` A numeric vector indicating the initial proportion of species 6.
- `response` A numeric vector giving the simulated response variable.

Details

What are Diversity-Interactions (DI) models?

Diversity-Interactions (DI) models (Kirwan et al 2009) are a set of tools for analysing and interpreting data from experiments that explore the effects of species diversity on community-level responses. We strongly recommend that users read the short introduction to Diversity-Interactions models (available at: DImodels). Further information on Diversity-Interactions models is also available in Kirwan et al 2009 and Connolly et al 2013.

Parameter values for the simulation

DI models take the general form of:

$$y = Identities + Interactions + Structures + \epsilon$$

where *y* is a community-level response, the *Identities* are the effects of species identities and enter the model as individual species proportions at the beginning of the time period, the *Interactions* are the interactions among the species proportions, while *Structures* include other experimental structures such as blocks, treatments or density.

The dataset `sim4` was simulated with:

- identity effects for the six species with values = 25, 16, 18, 20, 10, 12
- a covariate effect = 0.03
- all 15 pairwise interaction effects with values: 30, 27, 20, 15, 10, 9, 14, 18, 36, 17, 26, 32, 9, 21, 16 (for pairs of species 1-2, 1-3, 1-4, 1-5, 1-6, 2-3, 2-4, ... , 5-6 respectively).
- theta = 1 (where θ is a non-linear parameter included as a power on each $pipj$ product within interaction variables, see Connolly et al 2013 for details)
- ϵ assumed normally distributed with mean 0 and standard deviation 2.

References

Connolly J, T Bell, T Bolger, C Brophy, T Carnus, JA Finn, L Kirwan, F Isbell, J Levine, A Lüscher, V Picasso, C Roscher, MT Sebastia, M Suter and A Weigelt (2013) An improved model to predict the effects of changing biodiversity levels on ecosystem function. Journal of Ecology, 101, 344-355.

Kirwan L, J Connolly, JA Finn, C Brophy, A Lüscher, D Nyfeler and MT Sebastia (2009) Diversity-interaction modelling - estimating contributions of species identities and interactions to ecosystem function. Ecology, 90, 2032-2038.

Examples

```
####################################
## Code to simulate the sim4 dataset



## Simulate dataset sim4 with 6 species, three functional groups and three levels of a covariate
## The species 1-2 are FG1, species 3-4 are FG2 and species 5-6 are FG3.
## Assume ID effects and the full pairwise interaction model, with a covariate.
```

```
## Set up proportions
  data("design_b")
  sim4a <- design_b

# Replicate the design for three values of a covariate
  sim4b <- sim4a[rep(seq_len(nrow(sim4a)), times = 3), ]
  sim4c <- data.frame(treatment = rep(c(50, 150, 250), each = 47))
  sim4 <- data.frame(richness = sim4b[,1], sim4c, sim4b[,2:7])
  row.names(sim4) <- NULL

## To simulate the response, first create a matrix of predictors that includes p1-p6, the treatment
##  and all pairwise interaction variables
 X <- model.matrix(~ p1 + p2 + p3 + p4 + p5 + p6 + treatment + (p1 + p2 + p3 + p4 + p5 + p6)^2 -1,
                   data = sim4)

## Create a vector of 'known' parameter values for simulating the response.
## The first six are the p1-p6 parameters, and the second set of one is the treatment parameter
##  and the third set of 15 are the interaction parameters.
 sim4_coeff <- c(25,16,18,20,10,12,   0.03,    30,27,20,15,10,9,14,18,36,17,26,32,9,21,16)

## Create response and add normally distributed error
  sim4$response <- as.numeric(X %*% sim4_coeff)
  set.seed(34261)
  r <- rnorm(n = 141, mean = 0, sd = 2)
  sim4$response <- round(sim4$response + r, digits = 3)




###########################
## Analyse the sim4 dataset

## Load the sim4 data
  data(sim4)
## View the first five entries
  head(sim4)
## Explore the variables in sim4
  str(sim4)

## Check characteristics of sim4
  hist(sim4$response)
  summary(sim4$response)
  plot(sim4$richness, sim4$response)
 plot(sim4$richness[sim4$treatment==50], sim4$response[sim4$treatment==50], ylim=c(0,40))
 plot(sim4$richness[sim4$treatment==150], sim4$response[sim4$treatment==150], ylim=c(0,40))
 plot(sim4$richness[sim4$treatment==250], sim4$response[sim4$treatment==250], ylim=c(0,40))
  plot(sim4$p1, sim4$response)
  plot(sim4$p2, sim4$response)
  plot(sim4$p3, sim4$response)
  plot(sim4$p4, sim4$response)
  plot(sim4$p5, sim4$response)
  plot(sim4$p6, sim4$response)
```

```
## What model fits best? Selection using F-test
  auto1 <- autoDI(y = "response", prop = 3:8, treat = "treatment",
           FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), data = sim4, selection = "Ftest")
  summary(auto1)

## Ignore functional groups (will replace FG model with ADD model in Step 1 selection)
 auto2 <- autoDI(y = "response", prop = 3:8, treat = "treatment", data = sim4, selection = "Ftest")
  summary(auto2)


## Fit the functional group model using DI and the FG tag
  m1 <- DI(y = "response", prop = 3:8, treat = "treatment",
         FG = c("FG1","FG1","FG2","FG2","FG3","FG3"), DImodel = FG, data = sim4)
  summary(m1)

## Fit the additive species model using DI and the ADD tag
  m2 <- DI(y = "response", prop = 3:8, treat = "treatment", DImodel = ADD, data = sim4)
  summary(m2)

## Fit the full pairwise model using DI and the FULL tag
  m3 <- DI(y = "response", prop = 3:8, treat = "treatment", DImodel = FULL, data = sim4)
  summary(m3)
  plot(m3)


## Check goodness-of-fit using a half-normal plot with a simulated envelope
  library(hnp)
  hnp(m3)


## Create interaction variables and store them in a new dataset
 newlist <- DI_data_prepare(prop = 3:8, FG=c("FG1","FG1","FG2","FG2","FG3","FG3"), data = sim4)
  sim4a <- data.frame(newlist$newdata, newlist$FG)

## Fit the functional group model using DI and custom_formula
 m4 <- DI(custom_formula = response ~ 0 + p1 + p2 + p3 + p4 + p5 + p6 + treatment + bfg_FG1_FG2
          + bfg_FG1_FG3 + bfg_FG2_FG3 + wfg_FG1 + wfg_FG2 + wfg_FG3, data = sim4a)
  summary(m4)

## Fit the additive species model using DI and custom_formula
  m5 <- DI(custom_formula = response ~ 0 + p1 + p2 + p3 + p4 + p5 + p6 + treatment + p1_add
          + p2_add + p3_add + p4_add + p5_add + p6_add, data = sim4a)
  summary(m5)

## Fit the full pairwise model using DI and custom_formula
  m6 <- DI(custom_formula = response ~ 0 + p1 + p2 + p3 + p4 + p5 + p6 + treatment
          + (p1 + p2 + p3 + p4 + p5 + p6)^2, data = sim4a)
  summary(m6)

## Fit the full pairwise model using DI and the FULL tag,
##  and add in a treatment by average pairwise interaction term using extra_formula.
```

```
m7 <- DI(y = "response", prop = 3:8, treat = "treatment", DImodel = FULL,
         extra_formula = ~ AV:treatment, data = sim4a)
summary(m7)
```

---

sim5 *The Simulated "sim5" Dataset*

Description

The `sim5` dataset was simulated. There are nine species that vary in proportions (`p1` -`p9`). It is assumed that species 1 to 5 come from functional group 1, species 6 and 7 from functional group 2 and species 8 and 9 from functional group 3. The response was simulated assuming that there were species identity effects and functional group specific interaction effects, with theta = 0.7.

Usage

```
data(sim5)
```

Format

A data frame with 206 observations on the following 12 variables:

- `community` A numeric vector identifying each unique community, i.e., two rows with the same community value also share the same set of p1 to p9 values.
- `richness` A numeric vector identifying the number of species in the initial composition.
- `p1` A numeric vector indicating the initial proportion of species 1.
- `p2` A numeric vector indicating the initial proportion of species 2.
- `p3` A numeric vector indicating the initial proportion of species 3.
- `p4` A numeric vector indicating the initial proportion of species 4.
- `p5` A numeric vector indicating the initial proportion of species 5.
- `p6` A numeric vector indicating the initial proportion of species 6.
- `p7` A numeric vector indicating the initial proportion of species 7.
- `p8` A numeric vector indicating the initial proportion of species 8.
- `p9` A numeric vector indicating the initial proportion of species 9.
- `response` A numeric vector giving the simulated response variable.

Details

What are Diversity-Interactions (DI) models?

Diversity-Interactions (DI) models (Kirwan et al 2009) are a set of tools for analysing and interpreting data from experiments that explore the effects of species diversity on community-level responses. We strongly recommend that users read the short introduction to Diversity-Interactions models (available at: `DImodels`). Further information on Diversity-Interactions models is also available in Kirwan et al 2009 and Connolly et al 2013.

Parameter values for the simulation

DI models take the general form of:

$$y = Identities + Interactions + Structures + \epsilon$$

where *y* is a community-level response, the *Identities* are the effects of species identities and enter the model as individual species proportions at the beginning of the time period, the *Interactions* are the interactions among the species proportions, while *Structures* include other experimental structures such as blocks, treatments or density.

The dataset `sim5` was simulated with:

- identity effects for the nine species with values = 10, 9, 8, 7, 11, 6, 5, 8, 9
- functional group specific interaction effects; assume functional groups are labelled FG1, FG2 and FG3, then the interaction parameter values are: between FG1 and FG2 = 8, between FG1 and FG3 = 3, between FG2 and FG3 = 6, within FG1 = 6, within FG2 = 4 and within FG3 = 5
- theta = 0.7 (where θ is a non-linear parameter included as a power on each $pipj$ product within interaction variables, see Connolly et al 2013 for details)
- ϵ assumed normally distributed with mean 0 and standard deviation 1.2.

References

Connolly J, T Bell, T Bolger, C Brophy, T Carnus, JA Finn, L Kirwan, F Isbell, J Levine, A Lüscher, V Picasso, C Roscher, MT Sebastia, M Suter and A Weigelt (2013) An improved model to predict the effects of changing biodiversity levels on ecosystem function. Journal of Ecology, 101, 344-355.

Kirwan L, J Connolly, JA Finn, C Brophy, A Lüscher, D Nyfeler and MT Sebastia (2009) Diversity-interaction modelling - estimating contributions of species identities and interactions to ecosystem function. Ecology, 90, 2032-2038.

Examples

```
#####################################
## Code to simulate the sim5 dataset



## Simulate dataset sim5 with 9 species and three functional groups.
## The species 1-5 are FG1, species 6-7 are FG2 and species 8-9 are FG3.
## Assume ID effects and the FG interactions model, with theta = 0.7.

## Set up proportions
  data("design_a")
  sim5 <- design_a

## Create the functional group interaction variables, with theta = 0.7.
 newlist <- DI_data_FG(prop = 3:11, FG = c("FG1","FG1","FG1","FG1","FG1","FG2","FG2","FG3","FG3"),
                       data = sim5, theta = 0.7)
  sim5 <- data.frame(sim5, newlist$FG)
  names(sim5)[12:17] <- paste0(names(sim5)[12:17], "_theta")
```

```
## To simulate the response, first create a matrix of predictors that includes p1-p9, the
##  treatment and the interaction variables.
  X <- model.matrix(~ p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9
                    + bfg_FG1_FG2_theta + bfg_FG1_FG3_theta + bfg_FG2_FG3_theta
                    + wfg_FG1_theta + wfg_FG2_theta + wfg_FG3_theta -1, data = sim5)

## Create a vector of 'known' parameter values for simulating the response.
## The first nine are the p1-p9 parameters, and the second set of six are the interaction
##  parameters.
  sim5_coeff <- c(10,9,8,7,11, 6,5, 8,9,     8,3,6, 6,4,5)

##Create response and add normally distributed error
  sim5$response <- as.numeric(X %*% sim5_coeff)
  set.seed(35748)
  r <- rnorm(n = 206, mean = 0, sd = 1.2)
  sim5$response <- round(sim5$response + r, digits = 3)
  sim5[,12:17] <- NULL




###########################
## Analyse the sim5 dataset

## Load the sim5 data
  data(sim5)
## View the first five entries
  head(sim5)
## Explore the variables in sim5
  str(sim5)

## Check characteristics of sim5
  hist(sim5$response)
  summary(sim5$response)
  plot(sim5$richness, sim5$response)
  plot(sim5$p1, sim5$response)
  plot(sim5$p2, sim5$response)
  plot(sim5$p3, sim5$response)
  plot(sim5$p4, sim5$response)
  plot(sim5$p5, sim5$response)
  plot(sim5$p6, sim5$response)
  plot(sim5$p7, sim5$response)
  plot(sim5$p8, sim5$response)
  plot(sim5$p9, sim5$response)

## What model fits best? Selection using F-test in autoDI
  auto1 <- autoDI(y = "response", prop = 3:11,
                  FG = c("FG1","FG1","FG1","FG1","FG1","FG2","FG2","FG3","FG3"),
                  data = sim5, selection = "Ftest")
  summary(auto1)
```

```
## Fit the functional group model, with theta, using DI and the FG tag
  m1 <- DI(y = "response", prop = 3:11,
          FG = c("FG1","FG1","FG1","FG1","FG1","FG2","FG2","FG3","FG3"), DImodel = "FG",
           estimate_theta = TRUE, data = sim5)
  summary(m1)
  CI_95 <- theta_CI(m1, conf = .95)
  CI_95
  plot(m1)


## Check goodness-of-fit using a half-normal plot with a simulated envelope
  library(hnp)
  hnp(m1)


## Graph the profile likelihood
  library(ggplot2)
  ggplot(m1$profile_loglik, aes(x = grid, y = prof)) +
    theme_bw() +
    geom_line() +
    xlim(0,1.5) +
    xlab(expression(theta)) +
    ylab("Log-likelihood") +
    geom_vline(xintercept = CI_95, lty = 3) +
    labs(title = "   Log-likelihood versus theta",
         caption = "dotted vertical lines are upper and lower bounds of 95% CI for theta")

## Fit the functional group model, with theta set equal to the estimate from m1, and custom_formula.
## Note, it is not possible to estimate theta with custom_formula (only select a 'known' value).
## First, create the functional group interactions (theta value as estimated from m1),
##  store them in a new dataset and rename them with a theta indicator.
 newlist <- DI_data_FG(prop = 3:11, FG=c("FG1","FG1","FG1","FG1","FG1","FG2","FG2","FG3","FG3"),
                      theta = 0.7296887, data = sim5)
  sim5new <- data.frame(sim5, newlist$FG)
  names(sim5new)[13:18] <- paste0(names(sim5new)[13:18], "_theta")
  m2 <- DI(custom_formula = response ~ 0 + p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9 +
         bfg_FG1_FG2_theta + bfg_FG1_FG3_theta + bfg_FG2_FG3_theta
         + wfg_FG1_theta + wfg_FG2_theta + wfg_FG3_theta, data = sim5new)
## This will adjust the standard errors in m2 for the 'estimation' of theta
  m2$df.residual <- m2$df.residual - 1
  summary(m2)
## This will adjust the AIC in m2 for the 'estimation' of theta
  m2$aic <- m2$aic + 2
  summary(m2)
```

Switzerland *The "Switzerland" Dataset*

Description

This dataset comes from a grassland biodiversity experiment that was conducted in Switzerland as part of the "Agrodiversity Experiment" (Kirwan et al 2014). A total of 68 grassland plots were established across a gradient of species diversity, and two additional treatments (nitrogen fertiliser and total seed density) were also manipulated. The proportions of four species were varied across the plots: there were plots with 100% of a single species, and 2- and 4-species mixtures with varying proportions (e.g., (0.5, 0.5, 0, 0) and (0.7, 0.1, 0.1, 0.1)). Nitrogen fertiliser was either 50 or 100 kg N per annum and total seed density was either low or high. Total annual yield per plot was recorded for the first year after establishment. An analysis of the Switzerland dataset is presented in Kirwan et al 2009.

Usage

```
data("Switzerland")
```

Format

A data frame with 68 observations on the following 8 variables:

`plot` A numeric vector uniquely identifying each of the 68 plots.

`nitrogen` A factor with two levels: "50" or "150" to indicate the level of nitrogen fertiliser (kg N per annum) applied to the plot.

`density` A factor with two levels: "low" and "high" to indicate the level of total seed density used when sowing the plot.

`p1` A numeric vector indicating the proportion of species 1 in the plot. Species 1 was the grass species *Lolium perenne*.

`p2` A numeric vector indicating the proportion of species 2 in the plot. Species 2 was the grass species *Dactylis glomerata*.

`p3` A numeric vector indicating the proportion of species 3 in the plot. Species 3 was the legume species *Trifolium pratense*.

`p4` A numeric vector indicating the proportion of species 4 in the plot. Species 4 was the legume species *Trifolium repens*.

`yield` A numeric vector giving the total dry matter yield for the plot (tonnes per hectare per annum).

Details

What are Diversity-Interactions (DI) models?

Diversity-Interactions (DI) models (Kirwan et al 2009) are a set of tools for analysing and interpreting data from experiments that explore the effects of species diversity on community-level responses. We strongly recommend that users read the short introduction to Diversity-Interactions models (available at: `DImodels`). Further information on Diversity-Interactions models is also available in Kirwan et al 2009 and Connolly et al 2013.

Functional groups

In Ecology, species can be categorised into 'functional groups' based on their traits and functions. Here, the four species comprise two grasses (species 1 and 2) and two legumes (species 3 and 4); this is one possible 'functional group' categorisation of the four species.

Source

Kirwan L, J Connolly, C Brophy, O Baadshaug, G Belanger, A Black, T Carnus, R Collins, J Čop, I Delgado, A De Vliegher, A Elgersma, B Frankow-Lindberg, P Golinski, P Grieu, AM Gustavsson, Á Helgadóttir, M Höglind, O Huguenin-Elie, M Jørgensen, Ž Kadžiuliene, T Lunnan, A Lüscher, P Kurki, C Porqueddu, MT Sebastia, U Thumm, D Walmsley and JA Finn (2014) The Agrodiversity Experiment: three years of data from a multisite study in intensively managed grasslands. Ecology, 95, 2680.

References

Connolly J, T Bell, T Bolger, C Brophy, T Carnus, JA Finn, L Kirwan, F Isbell, J Levine, A Lüscher, V Picasso, C Roscher, MT Sebastia, M Suter and A Weigelt (2013) An improved model to predict the effects of changing biodiversity levels on ecosystem function. Journal of Ecology, 101, 344-355.

Kirwan L, J Connolly, JA Finn, C Brophy, A Lüscher, D Nyfeler and MT Sebastia (2009) Diversity-interaction modelling - estimating contributions of species identities and interactions to ecosystem function. Ecology, 90, 2032-2038.

Examples

```
## Load the Switzerland data
  data(Switzerland)
## View the first five entries
  head(Switzerland)
## Explore the variables in Switzerland
  str(Switzerland)

## Histogram of the response variable yield
  hist(Switzerland$yield)
## Explore the marginal relationship between yield and each predictor
  plot(Switzerland$p1, Switzerland$yield)
  plot(Switzerland$p2, Switzerland$yield)
  plot(Switzerland$p3, Switzerland$yield)
  plot(Switzerland$p4, Switzerland$yield)
  boxplot(yield ~ nitrogen, data = Switzerland)
  boxplot(yield ~ density, data = Switzerland)

## Check that the proportions sum to 1 (required for DI models)
## p1 to p4 are in the 4th to 7th columns in Switzerland
  Switzerlandsums <- rowSums(Switzerland[4:7])
  summary(Switzerlandsums)

## Model selection by F-test
  auto1 <- autoDI(y = "yield", density = "density", prop = c("p1","p2","p3","p4"),
                  treat = "nitrogen", FG = c("G","G","L","L"), data = Switzerland,
                  selection = "Ftest")
  summary(auto1)

## Fit the model chosen by autoDI using DI
 m1 <- DI(y = "yield", density = "density", prop = 4:7, DImodel = "FG", FG = c("G","G","L","L"),
          data = Switzerland)
```

```
  summary(m1)
  plot(m1)
  library(hnp)
  hnp(m1)

## Set up the functional group interactions and add to a new Switzerland2 dataset
  newlist <- DI_data_FG(prop = 4:7, FG = c("G","G","L","L"), data = Switzerland)
  Switzerland2 <- data.frame(Switzerland, newlist$FG)

## Additional model testing using DI to test for interactions with nitrogen
 m2 <- DI(y = "yield", block = "density", prop = 4:7, DImodel = "FG", FG = c("G","G","L","L"),
          data = Switzerland2, extra_formula = ~ nitrogen:bfg_G_L)
  summary(m2)
```

---

theta_CI *Compute Confidence Interval for Theta*

Description

This function allows the computation of a confidence interval for theta from a model object created from DI that includes the argument estimate_theta = TRUE, or from certain model objects created from autoDI.

A description of the non-linear parameter theta is available in Connolly et al 2013.

Usage

```
theta_CI(obj, conf = .95)
```

Arguments

obj	DI model object.
conf	Confidence level of the interval. The default is 0.95.

Details

The confidence interval calculated here is based on the values obtained when profiling the log-likelihood function for different values of theta. It is obtained in four steps:

1. define a grid of values for theta ranging from 0.01 to 2.5 of length 100

2. fit the DI model setting theta equal to each value in the grid and obtain the log-likelihood value corresponding to each value of theta

3. obtain linear interpolations between the log-likelihood (l) values (here we use approxfun)

4. calculate the lower and upper values of the CI by obtaining the values of theta corresponding to a log-likelihood value of $max_\theta(l) - 0.5 * \chi^2_(1 - \alpha; 1)$, where $max_\theta(l)$ is the maximum value of the profile log-likelihood obtained in the grid and $\chi^2_(conf; 1)$ is the $conf * 100\%$ percentile of the chi-squared distribution with 1 d.f.

When fitting any DI model setting `estimate_theta = TRUE`, steps 1 and 2 are automatically done within the `DI` function call. The `theta_CI` function performs steps 3 and 4 above to return the CI.

Note that when maximising the log-likelihood to find the estimate for theta, the parametric space is limited between 0.01 and 1.5. The larger grid (up to 2.5) is constructed to allow for obtaining the upper bound of the confidence interval in case the estimate of theta is close to 1.5.

Value

The function returns a named numeric vector with two values: the `lower` and `upper` limits of the `conf`*100% CI for theta.

Author(s)

Rafael A. Moral, John Connolly and Caroline Brophy

References

Connolly J, T Bell, T Bolger, C Brophy, T Carnus, JA Finn, L Kirwan, F Isbell, J Levine, A Lüscher, V Picasso, C Roscher, MT Sebastia, M Suter and A Weigelt (2013) An improved model to predict the effects of changing biodiversity levels on ecosystem function. Journal of Ecology, 101, 344-355.

See Also

`DI` `autoDI`

Other examples using the `theta_CI` function:
The `Bell` dataset examples.
The `sim2` dataset examples.
The `sim5` dataset examples.

Examples

```
## Load the sim5 data
  data(sim5)
## View the first five entries
  head(sim5)
## Explore the variables in sim5
  str(sim5)

## Fit the functional group model, with theta, using DI and the FG tag
  m1 <- DI(y = "response", prop = 3:11,
          FG = c("FG1","FG1","FG1","FG1","FG1","FG2","FG2","FG3","FG3"), DImodel = "FG",
           estimate_theta = TRUE, data = sim5)
  summary(m1)
  CI_95 <- theta_CI(m1, conf = .95)
  CI_95
## Graph the profile likelihood
  library(ggplot2)
  ggplot(m1$profile_loglik, aes(x = grid, y = prof)) +
```

```
theme_bw() +
geom_line() +
xlim(0,1.5) +
xlab(expression(theta)) +
ylab("Log-likelihood") +
geom_vline(xintercept = CI_95, lty = 3) +
labs(title = "   Log-likelihood versus theta",
    caption = "dotted vertical lines are upper and lower bounds of 95% CI for theta")
```

Index