Package ‘DImodelsMulti’

July 21, 2025

Type Package

Title Fit Multivariate Diversity-Interactions Models with Repeated Measures

Version 1.1.1

Maintainer Laura Byrne <ByrneL54@tcd.ie>

Description An add-on package to 'DImodels' for the fitting of biodiversity and ecosystem function relationship study data with multiple ecosystem function responses and/or time points. This package uses the multivariate and repeated measures Diversity-Interactions (DI) methods developed by Kirwan et al. (2009) <doi:10.1890/08-1684.1>, Finn et al. (2013) <doi:10.1111/1365-2664.12041>, and Dooley et al. (2015) <doi:10.1111/ele.12504>.

License GPL-2 | GPL-3

Encoding UTF-8

URL https://dimodels.com,
https://github.com/DI-LauraByrne/DImodelsMulti,

LazyData true

Imports DImodels, plyr, dplyr, nlme, stats, reshape2, knitr, Matrix

Suggests cli, fansi, crayon, rmarkdown, ggplot2, MASS, testthat (>= 3.0.0)

Depends R (>= 3.6.0)

RoxygenNote 7.2.3

VignetteBuilder knitr, rmarkdown

Config/testthat/edition 3

NeedsCompilation no

Author Laura Byrne [aut, cre] (ORCID: <https://orcid.org/0000-0003-2420-4019>),
Rishabh Vishwakarma [aut],
Rafael de Andrade Moral [aut],
Caroline Brophy [aut]

Repository CRAN

Date/Publication 2024-05-08 07:50:07 UTC

Contents

`AICc` *AICc*

Description

Calculate Second-order Akaike Information Criterion for a fitted DImulti object

Usage

```
AICc(model)
```

Arguments

`model`	an object of class DImulti

Value

A numeric value of the AICc value corresponding to the model object

`BICc` *BICc*

Description

Calculate Second-order Bayesian Information Criterion for a fitted DImulti obect

Usage

```
BICc(model)
```

Arguments

`model`	an object of class DImulti

Value

A numeric value of the BICc value corresponding to the model object

`dataBEL` *The Belgium dataset*

Description

Single site dataset containing thirty experimental units (plots), with four species seeded at two density levels, representing fifteen communities. Three ecosystem function responses are recorded for each plot at a single time point. Data is in a stacked format

Usage

```
data("dataBEL")
```

Format

A data frame with 90 observations on the following 9 variables.

`Plot` a numeric vector indicating the ID of the experimental unit from which the observation was recorded

`G1` a numeric vector ranging from 0 to 1, the proportion of the species G1

`G2` a numeric vector ranging from 0 to 1, the proportion of the species G2

`L1` a numeric vector ranging from 0 to 1, the proportion of the species L1

`L2` a numeric vector ranging from 0 to 1, the proportion of the species L2

`Density` a vector of factors with two levels, -1 or 1, representing the seeding density of the plot

`Var` a character vector indicating the ecosystem function recorded

`VarNum` a vector of factors with three levels, 1, 2, or 3 indicating the ecosystem function recorded

`Y` a numeric vector indicating the value of the ecosystem function recorded

Details

Data comes from a single site from a wider agrosdiversity experiment conducted in Belgium, established in 2002.
The four species used were Lolium perenne (G1), Phleum pratense (G2), Trifolium pratense (L1), and Trifolium repens (L2). There are two recommended functional groupings: grouping grasses (G1, G2) and legumes (L1, L2), or grouping fast-establishing species (G1, L1) and temporally persistent species (G2, L2).
Three ecosystem functions were recorded by summing recordings from four harvests over the first year of the experiment: (1) aboveground biomass of sown species (Sown) (t DM ha-1), (2) aboveground biomass of weed species (Weed) (t DM ha-1) and (3) the total annual yield of nitrogen in harvested aboveground biomass (N) (t DM ha-1).

Source

Dooley, A., Isbell, F., Kirwan, L., Connolly, J., Finn, J.A. and Brophy, C., 2015.
Testing the effects of diversity on ecosystem multifunctionality using a multivariate model.
Ecology Letters, 18(11), pp.1242-1251.

Kirwan, L., Connolly, J., Brophy, C., Baadshaug, O.H., Belanger, G., Black, A., Carnus, T., Collins, R.P., Cop, J., Delgado, I. and De Vliegher, A., 2014.
The Agrodiversity Experiment: three years of data from a multisite study in intensively managed grasslands.

References

Dooley, A., Isbell, F., Kirwan, L., Connolly, J., Finn, J.A. and Brophy, C., 2015.
Testing the effects of diversity on ecosystem multifunctionality using a multivariate model.
Ecology Letters, 18(11), pp.1242-1251.

Kirwan, L., Connolly, J., Brophy, C., Baadshaug, O.H., Belanger, G., Black, A., Carnus, T., Collins, R.P., Cop, J., Delgado, I. and De Vliegher, A., 2014.
The Agrodiversity Experiment: three years of data from a multisite study in intensively managed grasslands.

Finn, J.A., Kirwan, L., Connolly, J., Sebastia, M.T., Helgadottir, A., Baadshaug, O.H., Belanger, G., Black, A., Brophy, C., Collins, R.P. and Cop, J., 2013.
Ecosystem function enhanced by combining four functional types of plant species in intensively managed grassland mixtures: a 3-year continental-scale field experiment.
Journal of Applied Ecology, 50(2), pp.365-375 .

Kirwan, L., Connolly, J., Finn, J.A., Brophy, C., Luscher, A., Nyfeler, D. and Sebastia, M.T., 2009.
Diversity-interaction modeling: estimating contributions of species identities and interactions to ecosystem function.
Ecology, 90(8), pp.2032-2038.

Examples

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

## Modelling Example

# For a more thorough example of the workflow of this package, please see vignette
# DImulti_workflow using the following code:

# vignette("DImulti_workflow")

#
# In this example, we repeat the analysis in Dooley et al. 2015.

head(dataBEL)

# We begin with the main function DImulti(), passing the initial species proportions column names
# through 'prop' and the response value column name through 'y'.
# As the data is in a long or stacked format, we specify the ecosystem function type through the
# first index of the 'eco_func' parameter, along with specifying that we want an unstructured
# Sigma for these response types.
# The experimental unit ID is stored in the column "Plot" and so we pass this to 'unit_IDs'.
# Rather than estimating the nonlinear parameter theta, we opt to provide a value for each
# ecosystem function type through the parameter 'theta', which will be applied to the
# interaction terms as a power. In this case, we use functional group (FG) interactions, which
# requires an additional argument FG to be provided, specifying which group each species present
# in 'prop' belongs to. In this case, we group the grasses and the legumes.
# We include the treatment Density as an additional fixed effect.
# We opt to use the REML estimation method as we will not be doing any model comparisons.
# Finally, we specify the data object, dataBEL, through 'data'.

belModel <- DImulti(prop = c("G1", "G2", "L1", "L2"), y = "Y", eco_func = c("Var", "un"),
                     unit_IDs = "Plot", theta = c(0.5, 1, 1.2), DImodel = "FG",
                   FG = c("Grass", "Grass", "Legume", "Legume"), extra_fixed = ~ Density,
                     method = "REML", data = dataBEL)

# We can now print the output from our model, stored in belModel with class DImulti.

print(belModel)

# We can also retrieve the variance covariance matrix information from this object.

summary(belModel$corr[[1]])
nlme::getVarCov(belModel)
```

---

dataSWE *The Sweden dataset*

Description

Single site dataset containing 48 experimental units (plots), with four species seeded at two density levels, representing fifteen communities. One ecosystem function response is recorded for each plot at a three time points.

Usage

```
data("dataSWE")
```

Format

A data frame with 432 observations on the following 9 variables.

`YEARN` a numeric vector indicating the time point (year) that the ecosystem function recording was measured at

`PLOT` a numeric vector indicating the ID of the experimental unit from which the observation was recorded

`TREAT` a vector of factors with three levels, 1, or 2 indicating the treatment applied to the plot

`G1` a numeric vector ranging from 0 to 1, the proportion of the species G1

`G2` a numeric vector ranging from 0 to 1, the proportion of the species G2

`L1` a numeric vector ranging from 0 to 1, the proportion of the species L1

`L2` a numeric vector ranging from 0 to 1, the proportion of the species L2

`DENS` a character vector representing the seeding density of the plot (-1 or 1)

`YIELD` a numeric vector indicating the value of the harvest recorded

Details

Data comes from a single site from a wider agrosdiversity experiment conducted in Belgium, established in 2002.
The four species used were Lolium perenne (G1), Phleum pratense (G2), Trifolium pratense (L1), and Trifolium repens (L2). There are two recommended functional groupings: grouping grasses (G1, G2) and legumes (L1, L2), or grouping fast-establishing species (G1, L1) and temporally persistent species (G2, L2).
One ecosystem function (aboveground biomass (t DM ha-1)) was recorded by summing recordings from four harvests over each year, for three year.

Source

Kirwan, L., Connolly, J., Brophy, C., Baadshaug, O.H., Belanger, G., Black, A., Carnus, T., Collins, R.P., Cop, J., Delgado, I. and De Vliegher, A., 2014.
The Agrodiversity Experiment: three years of data from a multisite study in intensively managed grasslands.

References

Dooley, A., Isbell, F., Kirwan, L., Connolly, J., Finn, J.A. and Brophy, C., 2015.
Testing the effects of diversity on ecosystem multifunctionality using a multivariate model.
Ecology Letters, 18(11), pp.1242-1251.

Kirwan, L., Connolly, J., Brophy, C., Baadshaug, O.H., Belanger, G., Black, A., Carnus, T., Collins, R.P., Cop, J., Delgado, I. and De Vliegher, A., 2014.
The Agrodiversity Experiment: three years of data from a multisite study in intensively managed grasslands.

Finn, J.A., Kirwan, L., Connolly, J., Sebastia, M.T., Helgadottir, A., Baadshaug, O.H., Belanger, G., Black, A., Brophy, C., Collins, R.P. and Cop, J., 2013.
Ecosystem function enhanced by combining four functional types of plant species in intensively managed grassland mixtures: a 3-year continental-scale field experiment.
Journal of Applied Ecology, 50(2), pp.365-375 .

Kirwan, L., Connolly, J., Finn, J.A., Brophy, C., Luscher, A., Nyfeler, D. and Sebastia, M.T., 2009.
Diversity-interaction modeling: estimating contributions of species identities and interactions to ecosystem function.
Ecology, 90(8), pp.2032-2038.

Examples

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

## Modelling Example

# For a more thorough example of the workflow of this package, please see vignette
# DImulti_workflow using the following code:

# vignette("DImulti_workflow")


head(dataSWE)

# We begin by transforming the time identifier column to a factor, to better group coefficients

dataSWE$YEARN <- as.factor(dataSWE$YEARN)

# We fit a repeated measures DI model using the main function DImulti().
# We begin by passing the initial species proportions column names through 'prop' and the
# response value column name through 'y'.
# As the data contains multiple time points, we include the 'time' parameter, through which we
# specify the column name containing the repeated measure indicator and the autocorrelation
# structure we want to use, in this case, we use compound symmetry (CS).
# The experimental unit ID is stored in the column "PLOT" and so we pass this to 'unit_IDs'.
# We request that the method fits an average interaction structure and we include the treatments
# DENS and TREAT as additional fixed effects.
# We opt to use the ML estimation method to allow for model comparisons with different fixed
# effects.
# Finally, we specify the data object, dataSWE, through 'data'.

SWEmodel <- DImulti(y = "YIELD", time = c("YEARN", "CS"), unit_IDs = "PLOT",
                    prop = 5:8, data = dataSWE, DImodel = "AV",
                    extra_fixed = ~ 1:(DENS:TREAT),
                    method = "ML")

print(SWEmodel)

# We can now make any changes to the model's fixed effects and use a series of anovas (if the
# models are nested) or information criteria such as AIC, AICc, BIC, or BICc to select a final model
# We choose to change the interactions structure to functional groups, grouping the grasses and
# legumes, and simplify the extra_fixed terms, as an example, by only crossing the ID effect of G1
# with the treatments, and no longer crossing them with each other.

SWEmodel2 <- DImulti(y = "YIELD", time = c("YEARN", "CS"), unit_IDs = "PLOT",
                  prop = 5:8, data = dataSWE, DImodel = "FG", FG = c("G", "G", "L", "L"),
                    extra_fixed = ~ G1_ID:(DENS+TREAT),
                    method = "ML")

BICc(SWEmodel); BICc(SWEmodel2)

# We select the model with the lower information criteria, which is our functional group model, and
# refit it using the "REML" estimation method, for unbiased estimates.

SWEmodel2 <- DImulti(y = "YIELD", time = c("YEARN", "CS"), unit_IDs = "PLOT",
                  prop = 5:8, data = dataSWE, DImodel = "FG", FG = c("G", "G", "L", "L"),
                    extra_fixed = ~ G1_ID:(DENS+TREAT),
                    method = "REML")

# With this model, we can examine the coefficients

coef(SWEmodel2)

# or the variance and correlation structures

SWEmodel2$corr
nlme::getVarCov(SWEmodel2)
```

`DImodelsMulti` *Multivariate Diversity-Interactions (DI) Models with Repeated Measures*

Description

This package is an add-on to the **'DImodels'** package (Moral *et al.*, 2023). It enables the fitting of Diversity-Interactions (DI) models for (1) univariate repeated measures responses(Finn *et al.*, 2013), (2) multivariate responses at a single point in time (Dooley *et al.*, 2015), or (3) multivariate repeated measures responses. These responses will come from a biodiversity and ecosystem function (BEF) relationship study which aims to test the relationship between ecosystem functioning and ecosystem design, including species identities, interactions, and additional factors such as treatments and time. This data can be used to construct a regression model through the main function of this package `DImulti`.

Details

Data

This package is intended for use with data containing multiple ecosystem function responses and/or time points from a biodiversity and ecosystem function (BEF) relationship study. The dataset should contain a column for each species' proportion, so that each row of these columns sum to one. Each row of the data should also contain an identifier for the experimental unit being referenced, these identifiers must be unique to each experimental unit, but remain consistent over time and across functions. For each experimental unit included there must be recordings of either:
- a single community level ecosystem function response variable taken at multiple time points (repeated measures),
- multiple community level ecosystem function responses (multivariate), or
- multiple community level ecosystem function responses taken at multiple time points (multivariate repeated measures).
Ecosystem functions may be stored in a long or wide format while repeated measures must be stored in a long format.

Introduction to multivariate & repeated measures Diversity-Interactions models

Diversity-Interactions (DI) models are a regression based approach to modelling the biodiversity ecosystem function (BEF) relationship which assumes that the main driver behind changes in ecosystem functioning is the initial relative abundance (or proportions) of the species present. These models can be estimated using least squares estimation methods.
An example of a univariate DI model can be seen below,

$$y = \sum_{i=1}^{S} \beta_i p_i + \sum_{\substack{i,j=1 \\ i<j}}^{S} \delta_{ij} (p_i p_j)^{\theta} + \alpha A + \epsilon$$

where the response y represents the recorded ecosystem function, p_i represents the initial proportion of the i^{th} species, therefore the p values sum to 1 and form a simplex space, and scales the ID effect of the species, β_i; if no species interactions or treatments are required in the model, the response y is the weighted average of the species identity effects. S represents the number of unique species present in the study. Similarly to the ID effect, the interaction effect, δ, between species is scaled by some combination of the products of species proportions, which depends on the interaction structure chosen. The example above shows the full pairwise structure, which has a unique interaction term, δ_{ij}, per pair of species i & j. The nonlinear term θ (Connolly *et al.*, 2013; Vishwakarma *et al.*, 2023) is included in the model to allow the shape of the BEF relationship to change. This parameter can be estimated using profile log-likelihood optimisation (Brent, 1973) or can be assigned a set

value based on an *a priori* assumption/knowledge. A may include blocks or treatment terms, and α is a vector of the corresponding effect coefficients.
For further details of univariate DI modelling, see `?DImodels`, Kirwan *et al.*, 2009, and Moral *et al.*, 2023.

The multivariate DI model (Dooley *et al.*, 2015) extends the DI modelling framework to allow for the estimation of multiple ecosystem functions simultaneously, accounting for any existing covariance between functions through the error term. These models can be further extended through the introduction of repeated measures over multiple time points.
The structure for such models is:

$$y_{kmt} = \sum_{i=1}^{S} \beta_{ikt} p_{im} + \sum_{\substack{i,j=1 \\ i<j}}^{S} \delta_{ijkt} (p_{im} p_{jm})^{\theta_k} + \alpha_{kt} A + \epsilon_{kmt}$$

where y_{kmt} refers to the value of the k^{th} ecosystem function from the m^{th} experimental unit at a time point t. For an experimental unit m, β_{ikt} scaled by p_{im} is the expected contribution of the i^{th} species to the k^{th} response at time point t and is referred to as the i^{th} species' ID effect. The value of the nonlinear parameter θ is allowed to vary between ecosystem functions, in turn allowing the fixed effect structure to change across functions, in recognition that the nature of the species interactions could change between ecosystem functions.

In the case that a dataset contains only a single ecosystem function, the corresponding subscript k can simply be removed from the equation, the same can be said for the removal of the subscript t in the instance that a dataset contains a single time point.

The structure of the error term

For a univariate DI model, the error term is assumed to follow a normal distribution with mean 0 and variance σ^2.

$$\epsilon \sim N(0, \sigma^2)$$

When the model is extended to fit multivariate ($k > 1$) and/or repeated measures ($t > 1$) data, the error term is now assumed to follow a multivariate normal distribution with mean 0 and variance Σ^*.

$$\epsilon \sim MVN(0, \Sigma^*)$$

Σ^* is a block diagonal matrix, with one kt x kt block, Σ, for each experimental unit m. We refer to Σ as the variance covariance matrix for our ecosystem functions and time points. Typically, it includes a unique variance per combination of ecosystem functions and time points along the diagonal and a unique covariance between each pair of combinations on the off-diagonal. Autocorrelation structures may be implemented on the matrix Σ, either to simplify the estimation process or based on *a priori* knowledge. One structure is chosen for the ecosystem functions and another for repeated measures/time points, the two matrices are then estimated independently and combined using the Kronecker product ($\otimes$), a matrix multiplication method. In the case that the data is only multivariate ($k > 1$ & $t = 1$) or only has repeated measures ($k = 1$ & $t > 1$), only one autocorrelation structure needs to be chosen, with no multiplication necessary.
Three such structures are currently available in this package for repeated measures responses, and

two are available for multivariate responses:

1. **UN**: When each element of Σ is set to estimate independently, it is said to be unstructured or follow the general structure and is the preferred option in the case that there is no a priori information on the nature of these relationships. `corSymm`

2. **CS**: A simpler structure is compound symmetry, where it is assumed that each ecosystem function or time point has the same variance value σ^2 and each pair has the same covariance value $\sigma^2\rho$. This structure is not preferred for use with multiple ecosystem functions as it provides no meaningful interpretation, however it is allowed in this package if the model requires simplification. `corCompSymm`

3. **AR(1)**: An autocorrelation structure exclusive to repeated measures data is an autoregressive model of order one, which assumes that, each time point has the same variance, σ^2, and as the distance in pairs of time points increases, the covariance between them changes by a factor of ρ. `corAR1`

Author(s)

Laura Byrne [aut, cre], Rishabh Vishwakarma [aut], Rafael de Andrade Moral [aut], Caroline Brophy [aut]

Maintainer: Laura Byrne <byrnel54@tcd.ie>

References

Vishwakarma, R., Byrne, L., Connolly, J., de Andrade Moral, R. and Brophy, C., 2023.
Estimation of the non-linear parameter in Generalised Diversity-Interactions models is unaffected by change in structure of the interaction terms.
Environmental and Ecological Statistics, 30(3), pp.555-574.

Moral, R.A., Vishwakarma, R., Connolly, J., Byrne, L., Hurley, C., Finn, J.A. and Brophy, C., 2023.
Going beyond richness: Modelling the BEF relationship using species identity, evenness, richness and species interactions via the DImodels R package.
Methods in Ecology and Evolution, 14(9), pp.2250-2258.

Dooley, A., Isbell, F., Kirwan, L., Connolly, J., Finn, J.A. and Brophy, C., 2015.
Testing the effects of diversity on ecosystem multifunctionality using a multivariate model.
Ecology Letters, 18(11), pp.1242-1251.

Finn, J.A., Kirwan, L., Connolly, J., Sebastia, M.T., Helgadottir, A., Baadshaug, O.H., Belanger, G., Black, A., Brophy, C., Collins, R.P. and Cop, J., 2013.
Ecosystem function enhanced by combining four functional types of plant species in intensively managed grassland mixtures: a 3-year continental-scale field experiment.
Journal of Applied Ecology, 50(2), pp.365-375 .

Connolly, J., Bell, T., Bolger, T., Brophy, C., Carnus, T., Finn, J.A., Kirwan, L., Isbell, F., Levine, J., Luscher, A. and Picasso, V., 2013.
An improved model to predict the effects of changing biodiversity levels on ecosystem function.
Journal of Ecology, 101(2), pp.344-355.

Kirwan, L., Connolly, J., Finn, J.A., Brophy, C., Luscher, A., Nyfeler, D. and Sebastia, M.T., 2009.
Diversity-interaction modeling: estimating contributions of species identities and interactions to ecosystem function.
Ecology, 90(8), pp.2032-2038.

Brent, R.P., 1973.
Some efficient algorithms for solving systems of nonlinear equations.
SIAM Journal on Numerical Analysis, 10(2), pp.327-344.

See Also

This package: DImulti
Example datasets: dataBEL, dataSWE, simRM, simMV, simMVRM
Package family: DImodels, autoDI, DI, DI_data
Modelling package: nlme, gls

Examples

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


## Modelling Examples

# For a more thorough example of the workflow of this package, please see vignette
# DImulti_workflow using the following code:


vignette("DImulti_workflow")


## The simMVRM dataset
#
# This simulated dataset contains multiple ecosystem functions (k=3) and multiple time points
# (t=2). The dataset was
# simulated using unstructured Sigma matrices.
# The true values can be found in the help file for the data, ?simMVRM

data(simMVRM)
head(simMVRM)

# We will start the analysis with a call to the package's main function, DImulti().
# We begin the call by specifying the column indices holding the initial species proportions
# (p_i) through 'prop' and the columns which hold the ecosystem response values through 'y'.
# Since our data is multivariate, we include the 'eco_func' parameter, specifying "na" as our
# data is in a wide format (multiple columns in 'y') and "un" to fit an unstructured Sigma
# for our ecosystem functions.
# The data also contains repeated measures, so we include the 'time' parameter, specifying "time"
# as the column containing the time point indicator for each row and "ar1" to fit an AR(1)
# autocorrelation structure for our time points.
# Next, we specify that the experimental unit identifier is in column 1 through 'unit_IDs'. We
# indicate that we do not want to estimate the non-linear parameter theta, but do not provide
# any values, opting for the default value of 1.
# We specify a full pairwise (FULL) interaction structure through 'DImodel' and estimate the
# model using maximum likelihood (ML) as we may compare models.
# Finally, we provide the data object simMVRM through 'data'.

simModel_FULL <- DImulti(prop = 2:5, y = 6:8, eco_func = c("na", "un"), time = c("time", "ar1"),
                    unit_IDs = 1, estimate_theta = FALSE, DImodel = "FULL", method = "ML",
                          data = simMVRM)

# simModel_FULL is an object of custom class DImulti, which can be used with a number of S3
# methods and any method compatible with gls objects. We use summary() to examine the model fit,
# including fixed effect coefficients and the variance covariance matrix Sigma.

print(simModel_FULL)

# From this summary, we can see that there are many coefficients, a number of which are not
# statistically significant at an
# alpha level of 0.05, therefore this model may not be ideal for our data. We refit the model
# using an average interaction structure instead as it reduces the number of interaction terms
# to 1 per ecosystem function and time point.

simModel_AV <- DImulti(prop = 2:5, y = 6:8, eco_func = c("na", "un"), time = c("time", "ar1"),
                     unit_IDs = 1, estimate_theta = FALSE, DImodel = "AV", method = "ML",
                        data = simMVRM)
print(simModel_AV)

# This model looks like it is a better fit, with less insignificant terms, but we can test this
# formally using a likelihood ratio test, as the DImodels interaction structures are nested in
# nature (with the exception of "ADD" and "FG", which are on the same level in the nesting
# hierarchy).
# This will test the null hypothesis that the likelihood of the two models do not significantly
# differ, in other words, the added parameters of the more complex model are not worth it.

anova(simModel_FULL, simModel_AV)

# As the p-value is greater than our selected alpha value (0.05), we fail to reject the null
# hypothesis and so continue with the simpler model, simModel_AV.
# We can confirm this result by using information criteria such as AIC or BIC.
# We select the model with the lower value as it indicates a better fit.
# We use the second order versions of AIC and BIC (AICc and BICc) as we have a large number of
# terms, which can cause AIC and BIC to favour more complex models.

AICc(simModel_FULL); AICc(simModel_AV)
BICc(simModel_FULL); BICc(simModel_AV)

# We refit our chosen model using the REML estimation method to have unbiased estimates.
```

```
simModel_AV <- DImulti(prop = 2:5, y = 6:8, eco_func = c(”na”, ”un”), time = c(”time”, ”ar1”),
                    unit_IDs = 1, estimate_theta = FALSE, DImodel = ”AV”, method = ”REML”,
                       data = simMVRM)

# We can now examine the variance covariance matrix, Sigma, and the fixed effect coefficients,
# which can be retrieved from our initial summary() check or individually.

coef(simModel_AV)
simModel_AV$vcov

# An example of what we can infer from this is that ecosystem functions Y1 and Y2 have a positive
# covariance at time 1, while Y3 has a negative covariance with both Y1 and Y2 at the same time
# point. This means that maximising Y1 would not negatively impact Y2 but it would be at the cost
# of Y3. However, as our interaction term is positive and significant at this time point for all
# three ecosystem functions, we should be able to include a mixture of species that have positive
# ID effects for each response to help mitigate this trade-off.
# We can also predict from the model if we want to compare responses from different conditions,
# even those not included in the original data.

predict(simModel_AV, newdata = simMVRM[which(simMVRM$plot == 1:2), ])


##################################################################################################
## The Belgium dataset
#
# This real world dataset contains multiple ecosystem functions (k=3) at a single time point
# (t=1). The dataset also contains seeding density as a treatment in the form of a factor with
# two levels (1, -1). More detail can be found at ?dataBEL.

data(dataBEL)
head(dataBEL)

# We begin with the main function DImulti(), passing the initial species proportions column names
# through 'prop' and the response value column name through 'y'.
# As the data is in a long or stacked format, we specify the ecosystem function type through the
# first index of the 'eco_func' parameter, along with specifying that we want an unstructured
# Sigma for these response types.
# The experimental unit ID is stored in the column ”Plot” and so we pass this to 'unit_IDs'.
# Rather than estimating the nonlinear parameter theta, we opt to provide a value for each
# ecosystem function type through the parameter 'theta', which will be applied to the
# interaction terms as a power. In this case, we use functional group (FG) interactions, which
# requires an additional argument FG to be provided, specifying which group each species present
# in 'prop' belongs to. In this case, we group the grasses and the legumes.
# We include the treatment Density as an additional fixed effect.
# We opt to use the REML estimation method as we will not be doing any model comparisons.
# Finally, we specify the data object, dataBEL, through 'data'.

belModel <- DImulti(prop = c(”G1”, ”G2”, ”L1”, ”L2”), y = ”Y”, eco_func = c(”Var”, ”un”),
                    unit_IDs = ”Plot”, theta = c(0.5, 1, 1.2), DImodel = ”FG”,
                   FG = c(”Grass”, ”Grass”, ”Legume”, ”Legume”), extra_fixed = ~ Density,
                    method = ”REML”, data = dataBEL)
```

```
# We can now print the output from our model, stored in belModel with class DImulti.

print(belModel)


##################################################################################################
## The Sweden dataset
#
# This real-world dataset contains a single ecosystem function read at multiple time points
# (k=1 & t=3). The data contains two treatments, TREAT and DENS, each with two levels
# 1 & 2 and High & Low, respectively.
# More details can be found at ?dataSWE

data(dataSWE)
head(dataSWE)

# We transform the ”YEARN” column to factors to better act as groups in our models, giving us a
# coefficient per year number as opposed to acting as a continuous scaling factor.

dataSWE$YEARN <- as.factor(dataSWE$YEARN)

# We use the DImulti() function to fit a repeated measures DI model to this data.
# We specify the column indices 5 through 8 for our initial species proportions and the response
# value column name ”YIELD”.
# As there are multiple time points (repeated measures), we use the parameter 'time', providing
# the column name containing the time identifier through the first index and the desired Sigma
# structure (compound symmetry) through the second.
# The experimental unit ID is stored in column index two, which is passed through 'unit_IDs'.
# The interaction structure chosen is the average interaction term, ”AV”.
# We include both of the treatment terms, TREAT and DENS, as extra fixed effects crossed with
# each ID effect, the interaction effect, and with each other.
# We opt to use the REML
# estimation method for the model as we will not be doing any model comparisons.
# Finally, we specify the data object, dataSWE, through 'data'

SWEmodel <- DImulti(prop = 5:8, y = c(”YIELD”), time = c(”YEARN”, ”CS”),
                    unit_IDs = 2, DImodel = ”AV”, extra_fixed = ~ 1:(TREAT:DENS),
                    method = ”REML”, data = dataSWE)

print(SWEmodel)
```

DImulti *DImulti*

Description

A function to fit Diversity-Interactions models to data with multiple ecosystem functions and/or multiple time points

Usage

```
DImulti(
  y,
  eco_func = c("NA", "NA"),
  time = c("NA", "NA"),
  unit_IDs,
  prop,
  data,
  DImodel,
  FG = NULL,
  ID = NULL,
  extra_fixed = NULL,
  estimate_theta = FALSE,
  theta = 1,
  method = "REML"
)
```

Arguments

`y` If the dataset is in a wide format, this argument is a vector of k column names identifying the ecosystem function values recorded from each experimental unit in the dataset. For example, if the ecosystem function columns are labelled Y1 to Y4, then `y = c("Y1","Y2","Y3","Y4")`. Alternatively, the column numbers can be specified, for example, `y = 8:11`, where ecosystem function values are in the 8th to 11th columns. If the dataset is in the long/stacked format, this argument is the column name of the response value vector, for example, `y = "yield"`, alternatively, the column number can be supplied, `y = 8`

`eco_func` A vector of size two, with the first index holding the column name containing the character or factor indicator of which response was recorded (for use when stacked/long data passed through parameter `y`), otherwise it is the string `"NA"`, case insensitive. The second index should contain a string referring to the autocorrelation structure of the responses (for use in multivariate data case), options include `"un"` for unstructured/general and `"cs"` for compound symmetry. For example, `eco_func = c("Function", "cs")`, pertaining to multivariate data in a stacked format with a compound symmetry structure, or `eco_func = c("na", "un")`, meaning either univariate or wide multivariate data with an unstructured/general variance covariance matrix

`time` A vector of size two, with the first index holding the column name containing the repeated measures identifier (i.e., indicating which time point the corresponding response was recorded at), otherwise it is the string `"NA"`, case insensitive. The second index should contain a string referring to the autocorrelation structure of the repeated measures, options include `"un"` for unstructured/general, `"cs"` for compound symmetry, and `"ar1"` for an autoregressive model of order 1 (AR(1)). For example, `time = c("reading", "ar1")`, pertaining to repeated measures data with multiple readings on each unit with an AR(1) structure, or `time = c("na", "un")`, meaning multivariate data with no repeated measures (the correlation structure will be ignored) `unit_IDs` A vector of columns names/indices containing identifiers for the experimental units from which the observations (either multiple readings of the same response or a single reading of multiple responses) are taken, e.g., `unit_IDs = "Plot"`

`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

`data` A dataframe or tibble containing all previously input columns

`DImodel` A string, referring to the interaction structure of model to be fit from the full list: `"STR", "ID", "FULL", "E", "AV", "ADD", "FG"`

`FG` If species are classified by g functional groups, this argument takes a string vector (of length S) of the functional group to which each species referenced in the `prop` argument 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. This argument is optional but must be supplied if the `"FG"` interaction structure is specified in the `DImodel` argument

`ID` A text list (of length s) describing groupings for the identity of the effects of the species. These groupings will constrain some of the identity effects to be equal. For example, if there are four species and you wish to have two identity effect groups where species 1 and 3 and species 2 and 4 are grouped together: ID could be `ID = c("ID1", "ID2", "ID1", "ID2")`, where "ID1" and "ID2" are the names of the ID groups. This changes the identity component of the model from `'beta_1p_1 + beta_2p_2 + beta_3p_3 + beta_4p_4'` to `'beta_a(p_1 + p_3) + beta_b(p_2 + p_4)'`. This grouping does not affect the interaction terms

`extra_fixed` A formula expression for any additional fixed effect terms. For example, `extra_fixed = ~ Treatment` or `extra_fixed = ~ Treatment + Density`. Interactions across the regular formula can easily be included by beginning this argument with `1*` or `1:`, e.g. `extra_fixed = ~ 1:Density`. Any interactions included through this parameter will not be affected by `theta`. Any terms included will automatically be crossed with the column(s) specified through `eco_func` and/or `time`, therefore these interactions do not need to be specified here.

`estimate_theta` By default, θ (the power parameter on all `p_i * p_j` 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. A value of θ will be estimated for each ecosystem function present in the data, possibly changing the fixed effects across functions.

`theta` Users may specify a value of θ different than 1 to fit the DI model. Note that if `estimate_theta = TRUE`, then θ will be estimated via maximum profile log-likelihood and the value specified for `theta` will be overridden. Specify a vector of positive non-zero numerical values indicating the value for the non-linear parameter of the model for each ecosystem function (in alphabetical order, use `sort()` to find this) present in the dataset, changing the fixed effects across functions, or a single value to be used for all. For example, `theta = 0.5` or `theta = c(1, 0.5, 1)`

`method`	The string `"REML"` or `"ML"`, referring to the estimation method to be used. Defaults to "REML", which is suitable for model comparisons only if the fixed effects are held constant but provides unbiased estimates. If fixed effects are going to be tested between models, use "ML" for comparisons and then refit the chosen model using "REML"

Details

Data

This package is intended for use with data containing multiple ecosystem function responses and/or time points from a biodiversity and ecosystem function (BEF) relationship study. The dataset should contain a column for each species' proportion, so that each row of these columns sum to one. Each row of the data should also contain an identifier for the experimental unit being referenced, these identifiers must be unique to each experimental unit, but remain consistent over time and across functions. For each experimental unit included there must be recordings of either:
- a single community level ecosystem function response variable taken at multiple time points (repeated measures),
- multiple community level ecosystem function responses (multivariate), or
- multiple community level ecosystem function responses taken at multiple time points (multivariate repeated measures).
Ecosystem functions may be stored in a long or wide format while repeated measures must be stored in a long format.

Introduction to multivariate & repeated measures Diversity-Interactions models

Diversity-Interactions (DI) models are a regression based approach to modelling the biodiversity ecosystem function (BEF) relationship which assumes that the main driver behind changes in ecosystem functioning is the initial relative abundance (or proportions) of the species present. These models can be estimated using least squares estimation methods.
An example of a univariate DI model can be seen below,

$$y = \sum_{i=1}^{S} \beta_i p_i + \sum_{\substack{i,j=1 \\ i<j}}^{S} \delta_{ij} (p_i p_j)^{\theta} + \alpha A + \epsilon$$

where the response y represents the recorded ecosystem function, p_i represents the initial proportion of the i^{th} species, therefore the p values sum to 1 and form a simplex space, and scales the ID effect of the species, β_i; if no species interactions or treatments are required in the model, the response y is the weighted average of the species identity effects. S represents the number of unique species present in the study. Similarly to the ID effect, the interaction effect, δ, between species is scaled by some combination of the products of species proportions, which depends on the interaction structure chosen. The example above shows the full pairwise structure, which has a unique interaction term, δ_{ij}, per pair of species i & j. The nonlinear term θ (Connolly *et al.*, 2013; Vishwakarma *et al.*, 2023) is included in the model to allow the shape of the BEF relationship to change. This parameter can be estimated using profile log-likelihood optimisation (Brent, 1973) or can be assigned a set value based on an *a priori* assumption/knowledge. A may include blocks or treatment terms, and α is a vector of the corresponding effect coefficients.

For further details of univariate DI modelling, see `?DImodels`, Kirwan *et al.*, 2009, and Moral *et al.*, 2023.

The multivariate DI model (Dooley *et al.*, 2015) extends the DI modelling framework to allow for the estimation of multiple ecosystem functions simultaneously, accounting for any existing covariance between functions through the error term. These models can be further extended through the introduction of repeated measures over multiple time points.
The structure for such models is:

$$y_{kmt} = \sum_{i=1}^{S} \beta_{ikt} p_{im} + \sum_{\substack{i,j=1 \\ i<j}}^{S} \delta_{ijkt} (p_{im} p_{jm})^{\theta_k} + \alpha_{kt} A + \epsilon_{kmt}$$

where y_{kmt} refers to the value of the k^{th} ecosystem function from the m^{th} experimental unit at a time point t. For an experimental unit m, β_{ikt} scaled by p_{im} is the expected contribution of the i^{th} species to the k^{th} response at time point t and is referred to as the i^{th} species' ID effect. The value of the nonlinear parameter θ is allowed to vary between ecosystem functions, in turn allowing the fixed effect structure to change across functions, in recognition that the nature of the species interactions could change between ecosystem functions.

In the case that a dataset contains only a single ecosystem function, the corresponding subscript k can simply be removed from the equation, the same can be said for the removal of the subscript t in the instance that a dataset contains a single time point.

The structure of the error term

For a univariate DI model, the error term is assumed to follow a normal distribution with mean 0 and variance σ^2.

$$\epsilon \sim N(0, \sigma^2)$$

When the model is extended to fit multivariate ($k > 1$) and/or repeated measures ($t > 1$) data, the error term is now assumed to follow a multivariate normal distribution with mean 0 and variance Σ^*.

$$\epsilon \sim MVN(0, \Sigma^*)$$

Σ^* is a block diagonal matrix, with one kt x kt block, Σ, for each experimental unit m. We refer to Σ as the variance covariance matrix for our ecosystem functions and time points. Typically, it includes a unique variance per combination of ecosystem functions and time points along the diagonal and a unique covariance between each pair of combinations on the off-diagonal. Autocorrelation structures may be implemented on the matrix Σ, either to simplify the estimation process or based on *a priori* knowledge. One structure is chosen for the ecosystem functions and another for repeated measures/time points, the two matrices are then estimated independently and combined using the Kronecker product ($\otimes$), a matrix multiplication method. In the case that the data is only multivariate ($k > 1$ & $t = 1$) or only has repeated measures ($k = 1$ & $t > 1$), only one autocorrelation structure needs to be chosen, with no multiplication necessary.
Three such structures are currently available in this package for repeated measures responses, and two are available for multivariate responses:

1. **UN**: When each element of Σ is set to estimate independently, it is said to be unstructured or follow the general structure and is the preferred option in the case that there is no a priori information on the nature of these relationships. `corSymm`

2. **CS**: A simpler structure is compound symmetry, where it is assumed that each ecosystem function or time point has the same variance value σ^2 and each pair has the same covariance value $\sigma^2\rho$. This structure is not preferred for use with multiple ecosystem functions as it provides no meaningful interpretation, however it is allowed in this package if the model requires simplification. `corCompSymm`

3. **AR(1)**: An autocorrelation structure exclusive to repeated measures data is an autoregressive model of order one, which assumes that, each time point has the same variance, σ^2, and as the distance in pairs of time points increases, the covariance between them changes by a factor of ρ. `corAR1`

Value

`DImulti` - a custom class object containing the gls model fit with additional DI model attributes

Author(s)

Laura Byrne [aut, cre], Rishabh Vishwakarma [aut], Rafael de Andrade Moral [aut], Caroline Brophy [aut]
Maintainer: Laura Byrne <`byrnel54@tcd.ie`>

References

Vishwakarma, R., Byrne, L., Connolly, J., de Andrade Moral, R. and Brophy, C., 2023.
Estimation of the non-linear parameter in Generalised Diversity-Interactions models is unaffected by change in structure of the interaction terms.
Environmental and Ecological Statistics, 30(3), pp.555-574.

Moral, R.A., Vishwakarma, R., Connolly, J., Byrne, L., Hurley, C., Finn, J.A. and Brophy, C., 2023.
Going beyond richness: Modelling the BEF relationship using species identity, evenness, richness and species interactions via the DImodels R package.
Methods in Ecology and Evolution, 14(9), pp.2250-2258.

Dooley, A., Isbell, F., Kirwan, L., Connolly, J., Finn, J.A. and Brophy, C., 2015.
Testing the effects of diversity on ecosystem multifunctionality using a multivariate model.
Ecology Letters, 18(11), pp.1242-1251.

Finn, J.A., Kirwan, L., Connolly, J., Sebastia, M.T., Helgadottir, A., Baadshaug, O.H., Belanger, G., Black, A., Brophy, C., Collins, R.P. and Cop, J., 2013.
Ecosystem function enhanced by combining four functional types of plant species in intensively managed grassland mixtures: a 3-year continental-scale field experiment.
Journal of Applied Ecology, 50(2), pp.365-375 .

Connolly, J., Bell, T., Bolger, T., Brophy, C., Carnus, T., Finn, J.A., Kirwan, L., Isbell, F., Levine, J., Luscher, A. and Picasso, V., 2013.
An improved model to predict the effects of changing biodiversity levels on ecosystem function.
Journal of Ecology, 101(2), pp.344-355.

Kirwan, L., Connolly, J., Finn, J.A., Brophy, C., Luscher, A., Nyfeler, D. and Sebastia, M.T., 2009.
Diversity-interaction modeling: estimating contributions of species identities and interactions to ecosystem function.
Ecology, 90(8), pp.2032-2038.

Brent, R.P., 1973.
Some efficient algorithms for solving systems of nonlinear equations.
SIAM Journal on Numerical Analysis, 10(2), pp.327-344.

See Also

This package: `DImulti`
Example datasets: `dataBEL`, `dataSWE`, `simRM`, `simMV`, `simMVRM`
Package family: `DImodels`, `autoDI`, `DI`, `DI_data`
Modelling package: `nlme`, `gls`

Examples

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


## Modelling Examples

# For a more thorough example of the workflow of this package, please see vignette
# DImulti_workflow using the following code:


vignette("DImulti_workflow")


## The simMVRM dataset
#
# This simulated dataset contains multiple ecosystem functions (k=3) and multiple time points
# (t=2). The dataset was
# simulated using unstructured Sigma matrices.
# The true values can be found in the help file for the data, ?simMVRM

data(simMVRM)
head(simMVRM)

# We will start the analysis with a call to the package's main function, DImulti().
# We begin the call by specifying the column indices holding the initial species proportions
# (p_i) through 'prop' and the columns which hold the ecosystem response values through 'y'.
# Since our data is multivariate, we include the 'eco_func' parameter, specifying "na" as our
# data is in a wide format (multiple columns in 'y') and "un" to fit an unstructured Sigma
# for our ecosystem functions.
# The data also contains repeated measures, so we include the 'time' parameter, specifying "time"
# as the column containing the time point indicator for each row and "ar1" to fit an AR(1)
# autocorrelation structure for our time points.
# Next, we specify that the experimental unit identifier is in column 1 through 'unit_IDs'. We
# indicate that we do not want to estimate the non-linear parameter theta, but do not provide
# any values, opting for the default value of 1.
# We specify a full pairwise (FULL) interaction structure through 'DImodel' and estimate the
# model using maximum likelihood (ML) as we may compare models.
# Finally, we provide the data object simMVRM through 'data'.

simModel_FULL <- DImulti(prop = 2:5, y = 6:8, eco_func = c("na", "un"), time = c("time", "ar1"),
                    unit_IDs = 1, estimate_theta = FALSE, DImodel = "FULL", method = "ML",
                         data = simMVRM)

# simModel_FULL is an object of custom class DImulti, which can be used with a number of S3
# methods and any method compatible with gls objects. We use summary() to examine the model fit,
# including fixed effect coefficients and the variance covariance matrix Sigma.

print(simModel_FULL)

# From this summary, we can see that there are many coefficients, a number of which are not
# statistically significant at an
# alpha level of 0.05, therefore this model may not be ideal for our data. We refit the model
# using an average interaction structure instead as it reduces the number of interaction terms
# to 1 per ecosystem function and time point.

simModel_AV <- DImulti(prop = 2:5, y = 6:8, eco_func = c("na", "un"), time = c("time", "ar1"),
                     unit_IDs = 1, estimate_theta = FALSE, DImodel = "AV", method = "ML",
                       data = simMVRM)
print(simModel_AV)

# This model looks like it is a better fit, with less insignificant terms, but we can test this
# formally using a likelihood ratio test, as the DImodels interaction structures are nested in
# nature (with the exception of "ADD" and "FG", which are on the same level in the nesting
# hierarchy).
# This will test the null hypothesis that the likelihood of the two models do not significantly
# differ, in other words, the added parameters of the more complex model are not worth it.

anova(simModel_FULL, simModel_AV)

# As the p-value is greater than our selected alpha value (0.05), we fail to reject the null
# hypothesis and so continue with the simpler model, simModel_AV.
# We can confirm this result by using information criteria such as AIC or BIC.
# We select the model with the lower value as it indicates a better fit.
# We use the second order versions of AIC and BIC (AICc and BICc) as we have a large number of
# terms, which can cause AIC and BIC to favour more complex models.

AICc(simModel_FULL); AICc(simModel_AV)
BICc(simModel_FULL); BICc(simModel_AV)
```

```
# We refit our chosen model using the REML estimation method to have unbiased estimates.

simModel_AV <- DImulti(prop = 2:5, y = 6:8, eco_func = c("na", "un"), time = c("time", "ar1"),
                    unit_IDs = 1, estimate_theta = FALSE, DImodel = "AV", method = "REML",
                        data = simMVRM)

# We can now examine the variance covariance matrix, Sigma, and the fixed effect coefficients,
# which can be retrieved from our initial summary() check or individually.

coef(simModel_AV)
simModel_AV$vcov

# An example of what we can infer from this is that ecosystem functions Y1 and Y2 have a positive
# covariance at time 1, while Y3 has a negative covariance with both Y1 and Y2 at the same time
# point. This means that maximising Y1 would not negatively impact Y2 but it would be at the cost
# of Y3. However, as our interaction term is positive and significant at this time point for all
# three ecosystem functions, we should be able to include a mixture of species that have positive
# ID effects for each response to help mitigate this trade-off.
# We can also predict from the model if we want to compare responses from different conditions,
# even those not included in the original data.

predict(simModel_AV, newdata = simMVRM[which(simMVRM$plot == 1:2), ])


####################################################################################################
## The Belgium dataset
#
# This real world dataset contains multiple ecosystem functions (k=3) at a single time point
# (t=1). The dataset also contains seeding density as a treatment in the form of a factor with
# two levels (1, -1). More detail can be found at ?dataBEL.

data(dataBEL)
head(dataBEL)

# We begin with the main function DImulti(), passing the initial species proportions column names
# through 'prop' and the response value column name through 'y'.
# As the data is in a long or stacked format, we specify the ecosystem function type through the
# first index of the 'eco_func' parameter, along with specifying that we want an unstructured
# Sigma for these response types.
# The experimental unit ID is stored in the column "Plot" and so we pass this to 'unit_IDs'.
# Rather than estimating the nonlinear parameter theta, we opt to provide a value for each
# ecosystem function type through the parameter 'theta', which will be applied to the
# interaction terms as a power. In this case, we use functional group (FG) interactions, which
# requires an additional argument FG to be provided, specifying which group each species present
# in 'prop' belongs to. In this case, we group the grasses and the legumes.
# We include the treatment Density as an additional fixed effect.
# We opt to use the REML estimation method as we will not be doing any model comparisons.
# Finally, we specify the data object, dataBEL, through 'data'.

belModel <- DImulti(prop = c("G1", "G2", "L1", "L2"), y = "Y", eco_func = c("Var", "un"),
                     unit_IDs = "Plot", theta = c(0.5, 1, 1.2), DImodel = "FG",
                   FG = c("Grass", "Grass", "Legume", "Legume"), extra_fixed = ~ Density,
                     method = "REML", data = dataBEL)
```

```
# We can now print the output from our model, stored in belModel with class DImulti.

print(belModel)




#################################################################################################
## The Sweden dataset
#
# This real-world dataset contains a single ecosystem function read at multiple time points
# (k=1 & t=3). The data contains two treatments, TREAT and DENS, each with two levels
# 1 & 2 and High & Low, respectively.
# More details can be found at ?dataSWE

data(dataSWE)
head(dataSWE)

# We transform the "YEARN" column to factors to better act as groups in our models, giving us a
# coefficient per year number as opposed to acting as a continuous scaling factor.

dataSWE$YEARN <- as.factor(dataSWE$YEARN)

# We use the DImulti() function to fit a repeated measures DI model to this data.
# We specify the column indices 5 through 8 for our initial species proportions and the response
# value column name "YIELD".
# As there are multiple time points (repeated measures), we use the parameter 'time', providing
# the column name containing the time identifier through the first index and the desired Sigma
# structure (compound symmetry) through the second.
# The experimental unit ID is stored in column index two, which is passed through 'unit_IDs'.
# The interaction structure chosen is average interaction term, "AV".
# We include both of the treatment terms, TREAT and DENS, as extra fixed effects crossed with
# each ID effect, the interaction effect, and with each other.
# We opt to use the REML estimation method for the model as we will not be doing any model
# comparisons.
# Finally, we specify the data object, dataSWE, through 'data'

SWEmodel <- DImulti(prop = 5:8, y = c("YIELD"), time = c("YEARN", "CS"),
                    unit_IDs = 2, DImodel = "AV", extra_fixed = ~ 1:(TREAT:DENS),
                    method = "REML", data = dataSWE)

print(SWEmodel)
```

`getVarCov.DImulti` *getVarCov.DImulti*

Description

Extract the variance-covariance matrix from a fitted DImulti model

Usage

```
## S3 method for class 'DImulti'
getVarCov(obj, ...)
```

Arguments

`obj` an object of class DImulti

`...` some methods for this generic function require additional arguments. None are used in this method.

Value

A variance-covariance matrix or a named list of variance-covariance matrices.

See Also

`DImulti`

`predict.DImulti` *predict.DImulti*

Description

Predict from a multivariate repeated measures DI model

Usage

```
## S3 method for class 'DImulti'
predict(object, newdata = NULL, stacked = TRUE, ...)
```

Arguments

`object` an object of class DImulti

`newdata` an optional dataframe containing the communities from which to predict. If data is multivariate and in a wide format, to predict from a subset of ecosystem functions, as opposed to all, please include a column for each function with any numerical value, e.g. `newdata$Y2 <- 0`. If predicting from all functions, these columns may be included or left out.

`stacked` a logical value used to determine whether the output is in a wide or stacked format. Defaults to TRUE, meaning output is stacked/long.
If set to FALSE, non-unique groups of unit_IDs, ecosystem function, and time points will be aggregated upon widening using the mean function, please use unique unit_IDs values through newdata to avoid aggregation.

`...` some methods for this generic function require additional arguments. None are used in this method.

Value

The predictions from the supplied fitted DI models for the provided 'newdata', or the data used to fit the model if no 'newdata' is supplied. Predictions are returned in either a stacked or wide dataframe format.

See Also

`predict.gls` which this function wraps.

`print.DImulti` *print.DImulti*

Description

Print details of the fitted DI models supplied

Usage

```
## S3 method for class 'DImulti'
print(x, ...)
```

Arguments

`x`	an object of class DImulti
`...`	some methods for this generic function require additional arguments. None are used in this method.

Details

The appearance of the printed information will differ depending on whether a user has installed some combination of the suggested packages "crayon", "cli", and "fansi". These changes are mainly cosmetic, with crayon making the output easier to read, cli providing links to help files, and fansi enabling the reading of special characters in R markdown (Rmd) files. See 'Examples' below for suggested code to include in Rmd files.

Value

object x

See Also

`print` which this function wraps.

Examples

```
## Set up for R markdown for crayon and cli output if user has packages installed

if(requireNamespace("fansi", quietly = TRUE) &
   requireNamespace("crayon", quietly = TRUE) &
   requireNamespace("cli", quietly = TRUE))
{
 options(crayon.enabled = TRUE)

 ansi_aware_handler <- function(x, options)
 {
  paste0(
    "<pre class=\"r-output\"><code>",
    fansi::sgr_to_html(x = x, warn = FALSE, term.cap = "256"),
    "</code></pre>"
  )
 }

 old_hooks <- fansi::set_knit_hooks(knitr::knit_hooks,
                                    which = c("output", "message", "error", "warning"))

 knitr::knit_hooks$set(
   output = ansi_aware_handler,
   message = ansi_aware_handler,
   warning = ansi_aware_handler,
   error = ansi_aware_handler
  )
 }
##################################################################################################

## Usage

model <- DImulti(prop = c("G1", "G2", "L1", "L2"), y = "Y", eco_func = c("Var", "un"),
                 unit_IDs = "Plot", theta = c(0.5, 1, 1.2), DImodel = "FG",
                 FG = c("Grass", "Grass", "Legume", "Legume"), extra_fixed = ~ Density,
                 method = "REML", data = dataBEL)

print(model)
```

print.summary.DImulti *print.summary.DImulti*

Description

print the details of an object of class summary.DImulti

Usage

```
## S3 method for class 'summary.DImulti'
print(x, verbose = FALSE, digits = .Options$digits, ...)
```

Arguments

`x` an object of class DImulti

`verbose` an optional logical value used to control the amount of output when the object is printed. Defaults to FALSE.

`digits` the number of significant digits to use when printing

`...` some methods for this generic function require additional arguments. None are used in this method.

Value

object x

See Also

`print` which this function wraps.

`simMV` *The simulated multivariate "simMV" dataset*

Description

The `simMV` dataset was simulated from a multivariate DI model. It contains 192 plots comprising of six species that vary in proportions (`p1` - `p6`). Each plot was replicated once for testing a two-level factor `treat`, included at levels 0 and 1, resulting in a total of 384 plots. There are four simulated responses (`Y1` - `Y4`) recorded in a wide data format (one column per response). The data was simulated assuming that there were existing covariances between the responses, an additive treatment effect, and both species identity and species interaction effects were present.

Usage

```
data("simMV")
```

Format

A data frame with 384 observations on the following twelve variables.

`plot` A numeric vector identifying each unique experimental unit

`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

`treat` A two-level factor indicating whether a treatment was applied (1) or not (0)

`Y1` A numeric vector giving the simulated response for ecosystem function 1

`Y2` A numeric vector giving the simulated response for ecosystem function 2

`Y3` A numeric vector giving the simulated response for ecosystem function 3

`Y4` A numeric vector giving the simulated response for ecosystem function 4

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 multivariate DI model was developed by Dooley *et al.*, 2015.

Parameter values for the simulation

Multivariate DI models take the general form of:

$$y_{km} = Identities_{km} + Interactions_{km} + Structures_k + \epsilon_{km}$$

where y are the community-level responses, the $Identities$ are the effects of species identities for each response 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 `simMV` was simulated with:

- identity effects for the six species for response:
  - Y1 = 6.9, -0.3, 6.6, 1.7, -0.8, 4.3
  - Y2 = 4.9, 3.6, 4.4, 2.3, 4.3, 6.6
  - Y3 = -0.3, 4.6, 1.2, 6.8, 1.4, 6.9
  - Y4 = 4.1, 4.2, -0.5, 4.9, 6.7, -0.9, -1.0
- an average pairwise interaction effect for response:
  - Y1 = 1.8
  - Y2 = 6.8
  - Y3 = 1.4
  - Y4 = 0.3
- a teatment effect for response:
  - Y1 = 3.5
  - Y2 = -0.3
  - Y3 = 5.5
  - Y4 = -1.0
- ϵ assumed to have a multivaraite normal distribution with mean 0 and Sigma:

[1,] 3.87 -0.17 -0.23 0.31

[2,] -0.17 1.34 -0.11 0.49
[3,] -0.23 -0.11 2.95 -0.36
[4,] 0.31 0.49 -0.36 1.83

References

Dooley, A., Isbell, F., Kirwan, L., Connolly, J., Finn, J.A. and Brophy, C., 2015.
Testing the effects of diversity on ecosystem multifunctionality using a multivariate model.
Ecology Letters, 18(11), pp.1242-1251.

Finn, J.A., Kirwan, L., Connolly, J., Sebastia, M.T., Helgadottir, A., Baadshaug, O.H., Belanger, G., Black, A., Brophy, C., Collins, R.P. and Cop, J., 2013.
Ecosystem function enhanced by combining four functional types of plant species in intensively managed grassland mixtures: a 3-year continental-scale field experiment.
Journal of Applied Ecology, 50(2), pp.365-375 .

Kirwan, L., Connolly, J., Finn, J.A., Brophy, C., Luscher, A., Nyfeler, D. and Sebastia, M.T., 2009.
Diversity-interaction modeling: estimating contributions of species identities and interactions to ecosystem function.
Ecology, 90(8), pp.2032-2038.

Examples

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



## Modelling Example

# For a more thorough example of the workflow of this package, please see vignette
# DImulti_workflow using the following code:

# vignette(”DImulti_workflow”)

# We use head() to view the dataset
head(simMV)


# We can use the function DImulti() to fit a multivariate DI model, with an intercept for ”treat”
# for each ecosystem function. The dataset is wide, so the Y columns are all entered through 'y'
# and the first index of eco_func is ”NA”. We fit the average interaction structure and use ”ML”
# so that we can perform model comparisons with varying fixed effects
MVmodel <- DImulti(y = paste0(”Y”, 1:4), eco_func = c(”NA”, ”UN”), unit_IDs = 1,
              prop = paste0(”p”, 1:6), data = simMV, DImodel = ”AV”, extra_fixed = ~ treat,
                   method = ”ML”)
print(MVmodel)
```

```
# We can adjust the previous model to now cross "treat" with each other ID effect. We also specify
# different values of theta for each ecosystem function and use the "REML" method to get unbiased
# estimates.

MVmodel_theta <- DImulti(y = paste0("Y", 1:4), eco_func = c("NA", "UN"), unit_IDs = 1,
             prop = paste0("p", 1:6), data = simMV, DImodel = "AV", extra_fixed = ~ 1:treat,
                   theta = c(1, 0.5, 0.8, 0.6), method = "REML")

summary(MVmodel_theta)


# We can now use any S3 method compatible with a gls object, for example, predict()

predict(MVmodel_theta, newdata = simMV[which(simMV$plot == 1), ])


##################################################################################################
#
##################################################################################################
## Code to simulate data
#
#

set.seed(412)

props <- data.frame(plot = integer(),
                    p1 = integer(),
                    p2 = integer(),
                    p3 = integer(),
                    p4 = integer(),
                    p5 = integer(),
                    p6 = integer())

index <- 1 #row number

#Monocultures
for(i in 1:6) #6 species
{
  for(j in 1:2) #2 technical reps
  {
    props[index, i+1] <- 1
    index <- index + 1
  }
}


#Equal Mixtures
for(rich in sort(rep(2:6, 3))) #3 reps at each richness level
{
  sp <- sample(1:6, rich) #randomly pick species from pool

  for(j in 1:2) #2 technical reps
  {
    for(i in sp)
    {
      props[index, i+1] <- 1/rich #equal proportions
    }
    index <- index + 1
  }
}


#Unequal Mixtures
for(rich in sort(rep(c(2, 3, 4, 5, 6), 15))) #15 reps at each richness level
{
  sp <- sample(1:6, rich, replace = TRUE) #randomly pick species from pool

  for(j in 1:2) #2 technical reps
  {
    for(i in 1:6)
    {
      props[index, i+1] <- sum(sp==i)/rich #equal proportions
    }
    index <- index + 1
  }
}


props[is.na(props)] <- 0

mySimData <- props

mySimData$treat <- 0
mySimDataDupe <- mySimData
mySimDataDupe$treat <- 1
mySimData <- rbind(mySimData, mySimDataDupe)

mySimData$plot <- 1:nrow(mySimData)

mySimData$Y1 <- NA
mySimData$Y2 <- NA
mySimData$Y3 <- NA
mySimData$Y4 <- NA

ADDs <- DImodels::DI_data(prop=2:7, what=c("ADD"), data=mySimData)
mySimData <- cbind(mySimData, ADDs)

E_AV <- DImodels::DI_data(prop=2:7, what=c("E", "AV"), data=mySimData)
mySimData <- cbind(mySimData, E_AV)

n <- 4 #Number of Ys
p <- qr.Q(qr(matrix(stats::rnorm(n^2), n))) #Principal Components (make sure it's positive definite)
S <- crossprod(p, p*(n:1)) #Sigma
m <- stats::runif(n, -0.25, 1.5)

#runif(8, -1, 7) #decide on betas randomly
```

```
for(i in 1:nrow(mySimData))
{
  #Within subject error
  error <- MASS::mvrnorm(n=1, mu=m, Sigma=S)
  mySimData$Y1[i] <- 6.9*mySimData$p1[i] + -0.3*mySimData$p2[i] + 6.6*mySimData$p3[i] +
 1.7*mySimData$p4[i] + -0.8*mySimData$p5[i] + 4.3*mySimData$p6[i] + 3.5*mySimData$treat[i] +
  1.8*mySimData$AV[i] + error[1]
  mySimData$Y2[i] <- 4.9*mySimData$p1[i] + 3.6*mySimData$p2[i] + 4.4*mySimData$p3[i] +
 2.3*mySimData$p4[i] + 4.3*mySimData$p5[i] + 6.6*mySimData$p6[i] + -0.3*mySimData$treat[i] +
  6.8*mySimData$AV[i] + error[2]
  mySimData$Y3[i] <- -0.3*mySimData$p1[i] + 4.6*mySimData$p2[i] + 1.2*mySimData$p3[i] +
 6.8*mySimData$p4[i] + 1.4*mySimData$p5[i] + 6.9*mySimData$p6[i] + 5.5*mySimData$treat[i] +
  1.4*mySimData$AV[i] + error[3]
  mySimData$Y4[i] <- 4.1*mySimData$p1[i] + 4.2*mySimData$p2[i] + -0.5*mySimData$p3[i] +
 4.9*mySimData$p4[i] + 6.7*mySimData$p5[i] + -0.9*mySimData$p6[i] + -1.0*mySimData$treat[i] +
  0.3*mySimData$AV[i] + error[4]
}

mySimData$treat <- as.factor(mySimData$treat)
```

simMVRM *The simulated multivariate repeated measures "simMVRM" dataset*

Description

The simMVRM dataset was simulated from a multivariate repeated measures DI model. It contains 336 plots comprising of four species that vary in proportions (p1 - p4). There are three simulated responses (Y1, Y2, Y3), taken at two differing time points, recorded in a wide data format (one column per response type). The data was simulated assuming that there were existing covariances between the responses and between the time pointsand both species identity and species interaction effects were present.

Usage

```
data("simMVRM")
```

Format

A data frame with 672 observations on the following 9 variables.

plot a factor vector indicating the ID of the experimental unit

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

Y1 a numeric vector indicating the response of ecosystem function 1

`Y2` a numeric vector indicating the response of ecosystem function 2

`Y3` a numeric vector indicating the response of ecosystem function 3

`time` a factor with levels `1` `2`

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

Multivariate repeated measures DI models take the general form of:

$$y_{kmt} = Identities_{kmt} + Interactions_{kmt} + Structures_{kt} + \epsilon_{kmt}$$

where y are the community-level responses, the $Identities$ are the effects of species identities for each response and enter the model as individual species proportions measured 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 `simRM` was simulated with:

- identity effects for the four species for each time and ecosystem function:
  - Y1time1 = -1.0, 5.0, 2.8, -0.9
  - Y1time2 = 0.5, 5.4, 4.9, -2.1
  - Y2time1 = 0.1, 4.1, -0.5, 0.3
  - Y2time2 = 2.3, 3.2, -3.1, 2.1
  - Y3time1 = 0.9, 6.6, 3.5, 6.1
  - Y3time2 = -0.1, 7.0, 2.8, 4.0
- evenness interaction effect for each time and ecosystem function:
  - Y1time1 = -0.1
  - Y1time2 = 12.0
  - Y2time1 = 2.3
  - Y2time2 = 1.6
  - Y3time1 = 2.1
  - Y3time2 = 6.8
- ϵ assumed to have a multivaraite normal distribution with mean 0. An ecosystem function matrix Sigma:

  [1,] 2.36 0.71 -0.29
  [2,] 0.71 1.42 -0.39
  [3,] -0.29 -0.39 2.21

and a time matrix Sigma:

[1,] 1.69 0.46
[2,] 0.46 1.31

References

Dooley, A., Isbell, F., Kirwan, L., Connolly, J., Finn, J.A. and Brophy, C., 2015.
Testing the effects of diversity on ecosystem multifunctionality using a multivariate model.
Ecology Letters, 18(11), pp.1242-1251.

Finn, J.A., Kirwan, L., Connolly, J., Sebastia, M.T., Helgadottir, A., Baadshaug, O.H., Belanger, G., Black, A., Brophy, C., Collins, R.P. and Cop, J., 2013.
Ecosystem function enhanced by combining four functional types of plant species in intensively managed grassland mixtures: a 3-year continental-scale field experiment.
Journal of Applied Ecology, 50(2), pp.365-375 .

Kirwan, L., Connolly, J., Finn, J.A., Brophy, C., Luscher, A., Nyfeler, D. and Sebastia, M.T., 2009.
Diversity-interaction modeling: estimating contributions of species identities and interactions to ecosystem function.
Ecology, 90(8), pp.2032-2038.

Examples

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

## Modelling Example

# For a more thorough example of the workflow of this package, please see vignette
# DImulti_workflow using the following code:

# vignette("DImulti_workflow")

head(simMVRM)

# We call DImulti() to fit a series of models, with increasing complexity, and test whether the
# additional terms are worth keeping.

# We begin with an ID DI model, ensuring to use method = "ML" as we will be comparing fixed effects

MVRMmodel <- DImulti(y = 6:8, eco_func = c("Na", "un"), time = c("time", "CS"), unit_IDs = 1,
                     prop = 2:5, data = simMVRM, DImodel = "ID",
                     method = "ML")
print(MVRMmodel)

# Next, we include the simplest interaction structure available in this package, "AV", which adds
# a single extra term per ecosystem function and time point
```

```
MVRMmodel_AV <- DImulti(y = 6:8, eco_func = c("Na", "un"), time = c("time", "CS"), unit_IDs = 1,
                   prop = 2:5, data = simMVRM, DImodel = "AV",
                   method = "ML")

anova(MVRMmodel, MVRMmodel_AV)

# We select the more model with the lower AIC/BIC value or we use the p-value of the likelihood
# ratio test to determine if we reject the null hypothesis that the extra terms in the model are
# equal to zero, which in this case is lower than our alpha value of 0.05, so we do reject this
# hypothesis and continue with our more complex model.
#
# We can continue increasing the complexity of the interaction structure in the same fashion, this
# time we elect to use the additive interaction structure

MVRMmodel_ADD <- DImulti(y = 6:8, eco_func = c("Na", "un"), time = c("time", "CS"), unit_IDs = 1,
                   prop = 2:5, data = simMVRM, DImodel = "ADD",
                   method = "ML")

anova(MVRMmodel_AV, MVRMmodel_ADD)

# We fail to reject the null hypothesis and so we select the average interaction structure.
#
# Finally, we can also increase the model complexity via the inclusion of the non-linear parameter
# theta, which we can estimate, or select a value for. We also choose to estimate using the "REML"
# method as we will do no further fixed effect model comparisons


MVRMmodel_theta <- DImulti(y = 6:8, eco_func = c("Na", "un"), time = c("time", "CS"), unit_IDs = 1,
                   prop = 2:5, data = simMVRM, DImodel = "AV",
                   estimate_theta = TRUE, method = "REML")

print(MVRMmodel_theta)


#Finally, we can utilise this model for our interpretation and predictions

head(predict(MVRMmodel_theta))


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

## Code to simulate data

set.seed(746)

props <- data.frame(plot = integer(),
                    p1 = integer(),
                    p2 = integer(),
                    p3 = integer(),
                    p4 = integer())
```

```
index <- 1 #row number

#Monocultures
for(i in 1:4) #6 species
{
  for(j in 1:3) #2 technical reps
  {
    props[index, i+1] <- 1
    index <- index + 1
  }
}


#Equal Mixtures
for(rich in sort(rep(2:4, 4))) #3 reps at each richness level
{
  sp <- sample(1:4, rich) #randomly pick species from pool

  for(j in 1:2) #2 technical reps
  {
    for(i in sp)
    {
      props[index, i+1] <- 1/rich #equal proportions
    }
    index <- index + 1
  }
}


#Unequal Mixtures
for(rich in sort(rep(c(2, 3, 4), 50))) #15 reps at each richness level
{
  sp <- sample(1:4, rich, replace = TRUE) #randomly pick species from pool

  for(j in 1:2) #2 technical reps
  {
    for(i in 1:4)
    {
      props[index, i+1] <- sum(sp==i)/rich #equal proportions
    }
    index <- index + 1
  }
}


props[is.na(props)] <- 0

mySimData <- props

ADDs <- DImodels::DI_data(prop=2:5, what=c("ADD"), data=mySimData)
mySimData <- cbind(mySimData, ADDs)
```

```
E_AV <- DImodels::DI_data(prop=2:5, what=c("E", "AV"), data=mySimData)
mySimData <- cbind(mySimData, E_AV)

mySimData$plot <- 1:nrow(mySimData)

mySimData$Y1 <- NA
mySimData$Y2 <- NA
mySimData$Y3 <- NA

mySimData$time <- 1
mySimDataT1 <- mySimData
mySimDataT2 <- mySimData
mySimDataT2$time <- 2


nT <- 2 #Number of s
#Principal Components (make sure it's positive definite)
pT <- qr.Q(qr(matrix(stats::rnorm(nT^2), nT)))
ST <- crossprod(pT, pT*(nT:1)) #Sigma
mT <- stats::runif(nT, -0.25, 1.5)

nY <- 3 #Number of Ys
#Principal Components (make sure it's positive definite)
pY <- qr.Q(qr(matrix(stats::rnorm(nY^2), nY)))
SY <- crossprod(pY, pY*(nY:1)) #Sigma
mY <- stats::runif(nY, -0.25, 1.5)


#runif(7, -1, 7) #decide on betas randomly
for(i in 1:nrow(mySimData))
{
  #Within subject error
  errorT <- MASS::mvrnorm(n=1, mu=mT, Sigma=ST)
  errorY <- MASS::mvrnorm(n=1, mu=mY, Sigma=SY)
 mySimDataT1$Y1[i] <- -1.0*mySimDataT1$p1[i] + 5.0*mySimDataT1$p2[i] + 2.8*mySimDataT1$p3[i] +
  -0.9*mySimDataT1$p4[i] + -0.1*mySimDataT1$E[i] + errorT[1]*errorY[1]
 mySimDataT2$Y1[i] <- 0.5*mySimDataT2$p1[i] + 5.4*mySimDataT2$p2[i] + 4.9*mySimDataT2$p3[i] +
  -2.1*mySimDataT2$p4[i] + 12.0*mySimDataT1$E[i] + errorT[2]*errorY[1]

 mySimDataT1$Y2[i] <- 0.1*mySimDataT1$p1[i] + 4.1*mySimDataT1$p2[i] + -0.5*mySimDataT1$p3[i] +
  0.3*mySimDataT1$p4[i] + 2.3*mySimDataT1$E[i] + errorT[1]*errorY[2]
 mySimDataT2$Y2[i] <- 2.3*mySimDataT2$p1[i] + 3.2*mySimDataT2$p2[i] + -3.1*mySimDataT2$p3[i] +
  2.1*mySimDataT2$p4[i] + 1.6*mySimDataT2$E[i] + errorT[2]*errorY[2]

 mySimDataT1$Y3[i] <- 0.9*mySimDataT1$p1[i] + 6.6*mySimDataT1$p2[i] + 3.5*mySimDataT1$p3[i] +
  6.1*mySimDataT1$p4[i] + 2.1*mySimDataT1$E[i] + errorT[1]*errorY[3]
 mySimDataT2$Y3[i] <- -0.1*mySimDataT2$p1[i] + 7.0*mySimDataT2$p2[i] + 2.8*mySimDataT2$p3[i] +
  4.0*mySimDataT2$p4[i] + 6.8*mySimDataT2$E[i] + errorT[2]*errorY[3]
}

mySimData <- rbind(mySimDataT1, mySimDataT2)

mySimData$time <- as.factor(mySimData$time)
```

```
mySimData$plot <- as.factor(mySimData$plot)
```

simRM *The simulated repeated measures "simRM" dataset*

Description

The simRM dataset was simulated from a repeated measures DI model. It contains 116 plots comprising of four species that vary in proportions (p1 - p4). There is one simulated response (Y), taken at three differing time points, recorded in a stacked data format (one row per response value). The data was simulated assuming that there were existing covariances between the responses, an additive treatment effect, and both species identity and species interaction effects were present.

Usage

```
data("simRM")
```

Format

A data frame with 384 observations on the following seven variables.

plot A numeric vector identifying each unique experimental unit

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

Y A numeric vector giving the simulated response for an ecosystem function

time A three-level factor indicating the time point at which the response Y was recorded

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

Repeated measures DI models take the general form of:

$$y_{mt} = Identities_{mt} + Interactions_{mt} + Structures_t + \epsilon_{mt}$$

where y are the community-level responses, the $Identities$ are the effects of species identities for each response and enter the model as individual species proportions measured 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 simRM was simulated with:

- identity effects for the four species for each time:
  - time1 = 4.1, 2.1, 3.6, 4.8
  - time2 = 2.3, 2.4, 5.1, 5.0
  - time3 = 0.7, 2.3, 6.5, 5.9
- additive interaction effects for each time:
  - time1 = 3.3, 4.0, 1.3, 5.2
  - time2 = 3.6, 4.5, 0.5, 6.5
  - time3 = 4.5, 5.2, 0.6, 7.8
- ϵ assumed to have a multivaraite normal distribution with mean 0 and Sigma:

[1,] 2.01 -0.11 -0.08
[2,] -0.11 2.96 0.21
[3,] -0.08 0.21 1.03

References

Dooley, A., Isbell, F., Kirwan, L., Connolly, J., Finn, J.A. and Brophy, C., 2015.
Testing the effects of diversity on ecosystem multifunctionality using a multivariate model.
Ecology Letters, 18(11), pp.1242-1251.

Finn, J.A., Kirwan, L., Connolly, J., Sebastia, M.T., Helgadottir, A., Baadshaug, O.H., Belanger, G., Black, A., Brophy, C., Collins, R.P. and Cop, J., 2013.
Ecosystem function enhanced by combining four functional types of plant species in intensively managed grassland mixtures: a 3-year continental-scale field experiment.
Journal of Applied Ecology, 50(2), pp.365-375 .

Kirwan, L., Connolly, J., Finn, J.A., Brophy, C., Luscher, A., Nyfeler, D. and Sebastia, M.T., 2009.
Diversity-interaction modeling: estimating contributions of species identities and interactions to ecosystem function.
Ecology, 90(8), pp.2032-2038.

Examples

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

## Modelling Example

# For a more thorough example of the workflow of this package, please see vignette
# DImulti_workflow using the following code:
```

```
# vignette("DImulti_workflow")


head(simRM)

# We call DImulti() to fit a series of models, with increasing complexity, and test whether the
# additional terms are worth keeping.

# We begin with an ID DI model, ensuring to use method = "ML" as we will be comparing fixed effects

RMmodel <- DImulti(y = "Y", time = c("time", "AR1"), unit_IDs = "plot",
                   prop = paste0("p", 1:4), data = simRM, DImodel = "ID",
                   method = "ML")
print(RMmodel)

# We can simplify the above model by grouping the ID effect terms into two categories, G1 and G2

RMmodel_ID <- DImulti(y = "Y", time = c("time", "AR1"), unit_IDs = "plot",
                   prop = paste0("p", 1:4), data = simRM, DImodel = "ID",
                   ID = c("G1", "G1", "G2", "G2"),
                   method = "ML")

# We can compare the two models by calling the anova function to perform a likelihood ratio test

anova(RMmodel, RMmodel_ID)

# As the p-value < an alpha value of 0.05, we reject the null hypthesis that the extra terms are
# equal to zero, therefore we continue with the larger model.
# We can confirm this result by selecting the model with the lower AIC value.

AIC(RMmodel); AIC(RMmodel_ID)

# Next, we include the simplest interaction structure available in this package, "AV", which adds
# a single extra term per ecosystem function

RMmodel_AV <- DImulti(y = "Y", time = c("time", "AR1"), unit_IDs = "plot",
                   prop = paste0("p", 1:4), data = simRM, DImodel = "AV",
                   method = "ML")

anova(RMmodel, RMmodel_AV)

# Once again, we select the more complex model.
#
# We can continue increasing the complexity of the interaction structure in the same fashion, this
# time we elect to use the additive interaction structure

RMmodel_ADD <- DImulti(y = "Y", time = c("time", "AR1"), unit_IDs = "plot",
                   prop = paste0("p", 1:4), data = simRM, DImodel = "ADD",
                   method = "ML")

anova(RMmodel_AV, RMmodel_ADD)

# We continue with the additive interactions structure, the next interaction structure to test
# functional group interactions, is not nested with our additive model, so we compare th two using
# information criterion. In this case we choose to use AICc, to better penalise extra terms, as AIC
# becomes biased towards the more complex model as parameters numbers increase.
# The functional group strcuture requires an additional parameter, FG, to specify groups.

RMmodel_FG <- DImulti(y = "Y", time = c("time", "AR1"), unit_IDs = "plot",
                   prop = paste0("p", 1:4), data = simRM, DImodel = "FG",
                   FG = c("G1", "G1", "G2", "G2"),
                   method = "ML")

AICc(RMmodel_ADD); AICc(RMmodel_FG)

# In this case, we choose the lower valued model, which is the additive structure.
#
# The last interaction structure to test is the full pairwise model. Which we can see is not worth
# the extra terms. So our final interaction structure chosen is additive.

RMmodel_FULL <- DImulti(y = "Y", time = c("time", "AR1"), unit_IDs = "plot",
                   prop = paste0("p", 1:4), data = simRM, DImodel = "FULL",
                   method = "ML")

anova(RMmodel_ADD, RMmodel_FULL)

# Finally, we can also increase the model complexity via the inclusion of the non-linear parameter
# theta, which we can estimate, or select a value for. We also choose to estimate using the "REML"
# method as we will do no further fixed effect model comparisons

RMmodel_theta <- DImulti(y = "Y", time = c("time", "AR1"), unit_IDs = "plot",
                      prop = paste0("p", 1:4), data = simRM, DImodel = "ADD",
                      estimate_theta = TRUE, method = "REML")

print(RMmodel_theta)

# We can however test changes to the correlation structure, testing the AR(1) structure we have been
# using against the simpler CS strcuture.

RMmodel_CS <- DImulti(y = "Y", time = c("time", "CS"), unit_IDs = "plot",
                      prop = paste0("p", 1:4), data = simRM, DImodel = "ADD",
                      theta = 0.9991, method = "REML")

AICc(RMmodel_theta); AICc(RMmodel_CS)

# We see that they are practically identical, but the AR(1) strcuture is preffered.

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

## Code to simulate data

set.seed(523)

props <- data.frame(plot = integer(),
                    p1 = integer(),
                    p2 = integer(),
                    p3 = integer(),
                    p4 = integer())

index <- 1 #row number

#Monocultures
for(i in 1:4) #4 species
{
  for(j in 1:2) #2 technical reps
  {
    props[index, i+1] <- 1
    index <- index + 1
  }
}


#Equal Mixtures
for(rich in sort(rep(2:4, 3))) #3 reps at each richness level
{
  sp <- sample(1:4, rich) #randomly pick species from pool

  for(j in 1:2) #2 technical reps
  {
    for(i in sp)
    {
      props[index, i+1] <- 1/rich #equal proportions
    }
    index <- index + 1
  }
}


#Unequal Mixtures
for(rich in sort(rep(c(2, 3, 4), 15))) #15 reps at each richness level
{
  sp <- sample(1:4, rich, replace = TRUE) #randomly pick species from pool

  for(j in 1:2) #2 technical reps
  {
    for(i in 1:4)
    {
      props[index, i+1] <- sum(sp==i)/rich #equal proportions
    }
    index <- index + 1
  }
}


props[is.na(props)] <- 0

mySimData <- props
```

```
mySimData$Y <- NA

ADDs <- DImodels::DI_data(prop=2:5, what=c("ADD"), data=mySimData)
mySimData <- cbind(mySimData, ADDs)

E_AV <- DImodels::DI_data(prop=2:5, what=c("E", "AV"), data=mySimData)
mySimData <- cbind(mySimData, E_AV)

mySimData$plot <- 1:nrow(mySimData)

mySimData$time <- 1
mySimDataT1 <- mySimData
mySimDataT2 <- mySimData
mySimDataT3 <- mySimData
mySimDataT2$time <- 2
mySimDataT3$time <- 3


n <- 3 #Number of Ys
p <- qr.Q(qr(matrix(stats::rnorm(n^2), n))) #Principal Components (make sure it's positive definite)
S <- crossprod(p, p*(n:1)) #Sigma
m <- stats::runif(n, -0.25, 1.5)

#runif(11, -1, 7) #decide on betas randomly

for(i in 1:nrow(mySimData))
{
  #Within subject error
  error <- MASS::mvrnorm(n=1, mu=m, Sigma=S)
 mySimDataT1$Y[i] <- 4.1*mySimDataT1$p1[i] + 2.1*mySimDataT1$p2[i] + 3.6*mySimDataT1$p3[i] +
  4.8*mySimDataT1$p4[i] + 3.3*mySimDataT1$p1_add[i] + 4.0*mySimDataT1$p2_add[i] +
  1.3*mySimDataT1$p3_add[i] + 5.2*mySimDataT1$p4_add[i] + error[1]
 mySimDataT2$Y[i] <- 2.3*mySimDataT2$p1[i] + 2.4*mySimDataT2$p2[i] + 5.1*mySimDataT2$p3[i] +
  5.0*mySimDataT2$p4[i] + 3.6*mySimDataT2$p1_add[i] + 4.5*mySimDataT2$p2_add[i] +
  0.5*mySimDataT2$p3_add[i] + 6.5*mySimDataT2$p4_add[i] + error[2]
 mySimDataT3$Y[i] <- 0.7*mySimDataT3$p1[i] + 2.3*mySimDataT3$p2[i] + 6.5*mySimDataT3$p3[i] +
  5.9*mySimDataT3$p4[i] + 4.5*mySimDataT3$p1_add[i] + 5.2*mySimDataT3$p2_add[i] +
  0.6*mySimDataT3$p3_add[i] + 7.8*mySimDataT3$p4_add[i] + error[3]
}

mySimData <- rbind(mySimDataT1, mySimDataT2)
mySimData <- rbind(mySimData, mySimDataT3)

mySimData$time <- as.factor(mySimData$time)
mySimData$plot <- as.factor(mySimData$plot)
```

summary.DImulti *summary.DImulti*

Description

Print a summary of the fitted DI models supplied

Usage

```
## S3 method for class 'DImulti'
summary(object, verbose = FALSE, ...)
```

Arguments

`object`	an object inheriting from class DImulti, representing a generalized least squared fitted linear model using the Diversity-Interactions framework
`verbose`	an optional logical value used to control the amount of output when the object is printed. Defaults to FALSE.
`...`	some methods for this generic function require additional arguments. None are used in this method.

Value

an object inheriting from class summary.gls with all components included in object (see glsObject for a full description of the components) plus the following components:

corBeta, approximate correlation matrix for the coefficients estimates

tTable, a matrix with columns Value, Std. Error, t-value, and p-value representing respectively the coefficients estimates, their approximate standard errors, the ratios between the estimates and their standard errors, and the associated p-value under a t approximation. Rows correspond to the different coefficients.

residuals, if more than five observations are used in the gls fit, a vector with the minimum, first quartile, median, third quartile, and maximum of the residuals distribution; else the residuals.

AIC, the Akaike Information Criterion corresponding to object.

BIC, the Bayesian Information Criterion corresponding to object.

theta, the values of the non-linear parameter theta used in the model

See Also

`summary.gls` which this function wraps.

Index