trait.flex.anova

Description

This function decomposes the variation in community trait composition into two sources – the intraspecific trait variability and the variability due to species turnover. Their covariation is also separated. This decomposition is computed for the whole variation in the trait values and, depending on the formula specified, across the contribution of various explanatory variables considered in the model.

Usage

`trait.flex.anova(formula, specif.avg, const.avg, ...)`

Arguments

`formula`	A formula specifying the model
`specif.avg`	Vector with community trait composition values for a single trait. It is calculated from trait values specific to each community (i.e. trait values for individual species are ‘specific’ to each plot, or habitat, where the species is found)
`const.avg`	Vector with community trait composition values for a single trait. It is calculated from average (fixed) trait values of individual species (i.e. fixed trait value for individual species used for all habitats where the species is found)
`...`	Optional additional arguments are passed to calls to `aov` function made within the function. This could be specifications of `contrasts` or `weights` or `data` parameters

Details

The `formula` parameter must be a one-sided formula, i.e. starting with a tilde (~) character. The response variable is specified by the next two arguments, `specif.avg` and `const.avg`.

Value

An object of the `trait.flex` class. There are `print` and `plot` methods available for it. The object contains decomposition of sum of squares into intraspecific variation component, compositional variation component, their covariation and total in a `SumSq` element. This is a data frame with multiple rows if predictors were specified in `formula` argument. The `RelSumSq` element contains the same table relativized to unit row totals. Finally, the `anova.turnover`, `anova.total`, and `anova.diff` elements contain the three aov objects used to decompose the variation.

See also

`print.trait.flex`, `plot.trait.flex`

Example

```
# create example data frame
height.trait<- data.frame(MOWING=as.factor(c(0, 0, 1, 1, 1, 1, 0, 0, 1, 1,
0, 0)))
```

```
height.trait$FERTIL <- as.factor(c(0,1,0,1,1,0,1,0,1,0,1,0))
height.trait$specific <- c( 58.16354, 62.34342, 31.43701, 62.14333,
 51.98859, 29.95968, 55.48009, 50.68146, 51.75618, 31.13289,
 47.53024, 56.44128)
height.trait$nonspec<- c( 47.93985, 57.09998, 43.06760, 51.58106,
 44.52435, 40.85160, 50.85945, 44.48371, 43.20859, 43.92655,
 45.15222, 47.83641)

# calculate decomposition of total trait variance
x1<-trait.flex.anova(~1, specific, nonspec, data=height.trait)
print(x1)
par(ask=TRUE)
plot(x1)

# calculate decomposition of the factorial model
x2<-trait.flex.anova(~MOWING*FERTIL, specific, nonspec, data=height.trait)
print(x2)
plot(x2)
plot(x2, legend.pos="none")
plot(x2, plot.total=TRUE)
plot(x2, plot.covar=TRUE)

# and now the relative values ...
plot(x1, use.percentage=T)
plot(x2, use.percentage=T)
plot(x2, legend.pos="none", use.percentage=T)
plot(x2, plot.total=TRUE, use.percentage=T)
plot(x2, plot.covar=TRUE, use.percentage=T, main="Height")
par(ask=FALSE)
```

Output of the first *print* command for *x1* above looks like this:

```
Decomposing trait sum of squares into composition turnover
 effect, intraspecific trait variability, and their covariation:
      Turnover Intraspec. Covariation  Total
Total   231.43     861.59      450.76 1543.8

 Relative contributions:
      Turnover Intraspec. Covariation Total
Total   0.1499     0.5581       0.292     1
```

Output of the *print* command for *x2* above looks like this:

```
Decomposing trait sum of squares into composition turnover
 effect, intraspecific trait variability, and their covariation:
               Turnover Intraspec. Covariation   Total
MOWING         57.25518    176.414     201.003  434.67
FERTIL         49.28825    200.950     199.043  449.28
MOWING:FERTIL   0.15949    464.690     -17.218  447.63
Residuals     124.73075     19.532      67.929  212.19
Total         231.43367    861.586     450.757 1543.78

 Relative contributions:
               Turnover Intraspec. Covariation  Total
MOWING        0.0370877    0.11427     0.13020 0.2816
FERTIL        0.0319270    0.13017     0.12893 0.2910
MOWING:FERTIL 0.0001033    0.30101    -0.01115 0.2900
Residuals     0.0807958    0.01265     0.04400 0.1374
Total         0.1499139    0.55810     0.29198 1.0000

 Significance of testable effects:
              Turnover Intraspec.     Total
MOWING        0.091636 2.8145e-05 0.0036935
FERTIL        0.113297 1.7473e-05 0.0033643
MOWING:FERTIL 0.921928 7.3587e-07 0.0033995
```

print.trait.flex

Description

Print the decomposition of trait variation, obtained with the `trait.flex.anova` function.

Usage

```
print.trait.flex( x)
```

Arguments

`x`	object of trait.flex class returned by the `trait.flex.anova` function.

Example

See `trait.flex.anova` example

plot.trait.flex

Description

Summarizes graphically the decomposition of trait variation, obtained with the `trait.flex.anova` function.

Usage

```
plot.trait.flex ( x, plot.total = FALSE, use.percentage = FALSE,
                  plot.covar = FALSE, legend.pos = if(plot.total)
                    "topleft" else "topright", ...)
```

Arguments

`x`	This is the only obligatory argument, representing an object returned from the `trait.flex.anova` function
`plot.total`	This argument (with default value `FALSE`) can be set to `TRUE`, to plot not only the individual components of variation, but also the total variation. This is useful particularly when the decomposition was done with non-trivial formula (i.e. with explanatory variables)
`use.percentage`	When this argument is `TRUE` (default value is `FALSE`), the individual plotted sources of trait variation are shown as percentages of the total variation, on 0-100 scale
`plot.covar`	When this argument is `TRUE` (the default is `FALSE`), the covariance between within-species trait variability and the variability due to species composition turnover is plotted as yet another category within the stacked bars. This is not possible, however, when the covariation component has a negative sign. If so, it is better to keep this parameter as `FALSE`, because then the covariation is not explicitly visualized, but can be deduced, including its sign, from the total variation, shown in this case as a horizontal line. The `plot.covar` argument is entirely ignored when plotting `trait.flex` object fitted with a formula without any predictor variables.

`legend.pos`	This argument allows you to specify the position of graph legend. Its meaning is identical to the first parameter of the `legend` function. Similarly to `plot.covar` argument, this one is also entirely ignored when plotting `trait.flex` object created with a formula without predictors.
`...`	Any additional arguments are passed to the `barplot` function creating the core of the plot and can be used to adjust the look of resulting graph.

Example

See `trait.flex.anova` example