

Package: **gravy** (via **r-universe**)

September 18, 2024

Title Gradient Analysis of Vegetation

Version 0.4-0

Author Jari Oksanen

Maintainer Jari Oksanen <jhoksane@gmail.com>

Imports lattice

Description Experimental tools for gradient analysis of community data. Most of the functionality is now included in package 'eHOF', but there are still some divergent features.

License MIT + file LICENCE

URL <https://github.com/jarioksa/gravy/>

Repository <https://jarioksa.r-universe.dev>

RemoteUrl <https://github.com/jarioksa/gravy>

RemoteRef HEAD

RemoteSha 8178ad0cad9ca203144571b453dbffb40df5a976

Contents

betadiversity	2
boxgradient	3
GaussPara	4
gradscale	6
hillscale	7
HOF	9
mtf01	11
nichelap	11
plotGrad	13

Index	15
--------------	-----------

betadiversity	<i>Estimates Beta Diversity as the instantaneous rate of change at any gradient point.</i>
---------------	--

Description

The function `betadiversity` estimates the beta diversity as the instantaneous rate of change at any gradient point.

Usage

```
betadiversity(object, x, ...)  
## S3 method for class 'betadiversity'  
plot(x, type="b", xlab, ylab, splines = FALSE, ...)
```

Arguments

<code>object</code>	A response frame object: Fitted models for species.
<code>x</code>	Gradient values.
<code>type</code>	Type of graph.
<code>xlab, ylab</code>	Axis labels; defaults provided if these are missing.
<code>splines</code>	Use interpolating spline to draw smooth lines.
<code>...</code>	Other parameters.

Details

The function finds the instantaneous rate of change along the gradient from the fitted response functions for species.

Currently the function is implemented for [HOF](#) models only.

The function has a `plot` method.

Value

The function returns an object of class `"betadiversity"` with the following items:

<code>x</code>	Used gradient values.
<code>beta</code>	Beta diversity at the gradient points.

...

Author(s)

Jari Oksanen

References

Oksanen, J. & Tonteri, T. (1995). Rate of compositional turnover along gradients and total gradient length. *Journal of Vegetation Science* 6, 815-824.

See Also

[HOF](#).

Examples

```
data(mtf01)
data(mtf.alt)
attach(mtf.alt)
mod <- HOF(mtf01, Altitude, 1)
x <- seq(min(Altitude), max(Altitude), len=101)
beta <- betadiversity(mod, x)
plot(beta)
```

boxgradient

Display Species Response along Gradients as Boxplots or Gaussian Models

Description

The function display species responses along ecological gradients using either [boxplot](#) or a [polygon](#) displaying fitted Gaussian responses.

Usage

```
boxgradient(x, grad, horizontal = TRUE, xlab, freq.lim = 5,
  cex.species= 0.7, axes = TRUE, ...)
gaussgradient(x, grad, family = poisson, xlab, freq.lim = 5,
  cex.species = 0.7, axes = TRUE, ...)
```

Arguments

x	Community data
grad	Gradient vector
horizontal	Use horizontal boxplot .
family	Error family used in fitting the Gaussian model using glm . The link function will always be log, and some families cannot be fitted with this link.
xlab	Label of the gradient. Variable name used as default.
freq.lim	Frequency of rarest species displayed.
cex.species	Size multiplier for species labels. Species labels are printed horizontally, and typically you must reduce their size with this parameter or increase the margin (see mar in par), or abbreviate names.
axes	Draw axis with labels.
...	Other parameters passed to underlying functions boxplot or polygon .

Details

The functions are intended for simultaneous display of species responses along an ecological gradient.

Function `boxgradient` will draw `boxplots` of species presences. In addition, it puts points at weighted averages. The boxplots are based on presence data only, but weighted averages are based on the original quantitative information. Boxes are arranged by medians, but breaking ties by weighted averages. The function also adds lines of quartiles and median for the whole gradient, and adds a `rug` of gradient values.

Function `gaussgradient` draws fitted Gaussian response models. The curves are adjusted to the same area so that narrower responses will look higher, but the heights have no relation to the original abundances. The responses are fitted using `glm`, and species with failed fitting (“inverted” responses) will not be displayed. The fitted responses are ordered by the location of the top (‘optimum’).

Value

The functions are used to draw plots. Function `boxpgradient` returns invisibly the object returned by `boxplot`. Function `gaussgradient` returns invisibly a list of fitted optima and tolerances.

Author(s)

Jari Oksanen

See Also

`boxplot`, `polygon`, `axis`, `par`.

Examples

```
data(mtf01)
data(mtf.alt)
attach(mtf.alt)
op <- par(no.readonly=TRUE)
par(mfrow=c(2,1))
par(mar=c(4,6,1,1))
boxgradient(mtf01, Altitude, col="pink", border="blue", notch=TRUE)
gaussgradient(mtf01, Altitude, col="pink", border="blue")
par(op)
```

GaussPara

Finds Gaussian Parameters of a Response Model.

Description

`GaussPara` is a generic function to find Gaussian parameters in any response function. In Gaussian function, these will be the real Gaussian parameters, in other models similar parameters for the height at optimum (“top”), location of the optimum (“opt”) and the width of the response at height $\exp(-0.5)*top$. For unsymmetric responses, parameter `tol` gives the width to the left and `tol.right` the width to the right of `opt`.

Usage

```
GaussPara(resp, ...)
```

Arguments

<code>resp</code>	Response function for a single species or a list of responses from a data frame method.
<code>...</code>	Other parameters. See individual response models for possible other parameters.

Details

See individual response functions for more details.

Value

Returns an object with (at least) items `top` for the height at the optimum, `opt` for the location of the optimum), `tol` for the width to the left of the optimum, `tol.right` width to the right of the optimum or text `symmetric` for symmetric responses. Individual response functions may add some items to these.

Note

Currently implemented only for [HOF](#).

Author(s)

Jari Oksanen

References

Lawesson, J.E. & Oksanen, J. (2002) Niche characteristics of Danish woody species as derived from coenoclines. *Journal of Vegetation Science* 13: 279-290. (For HOF models.)

See Also

[HOF](#) is the only implementation. If the parameters cannot be found in closed form, [optimize](#) is used for `opt` and [uniroot](#) for `tol` and `tol.right`.

Examples

```
data(mtf01)
data(mtf.alt)
attach(mtf.alt)
mods <- HOF(mtf01, Altitude, 1)
GaussPara(mods)
```

gradscale	<i>Rescales an ecological gradient to a constant rate of change of community composition.</i>
-----------	---

Description

Function `gradscale` tries to scale an ecological gradient to a constant rate of compositional change. Usually this is not strictly possible.

Usage

```
gradscale(resp, grad, ...)
```

Arguments

<code>resp</code>	A data frame of fitted species response models.
<code>grad</code>	The gradient used for responses <code>resp</code> .
<code>...</code>	Other parameters.

Details

Function `gradscale` tries to rescale an ecological gradient to the constant rate of compositional change as measured by [betadiversity](#). Usually this is not strictly possible, because the responses fitted after rescaling indicate local aberrations, although the average change is close to constant.

The function `gradscale` makes the distance of ordered gradient points equal to sum of change in species abundance. In most cases, this is equal to change in fitted values, but if the species has its optimum between the gradient values, `gradscale` will estimate the accumulated change through that optimum point.

Value

The function returns the rescaled gradient at the point given by `grad`.

Author(s)

Jari Oksanen.

References

Oksanen, J. & Tonteri, T. (1995). Rate of compositional turnover along gradients and total gradient length. *Journal of Vegetation Science* 6, 815-824.

See Also

[betadiversity](#).

Examples

```

data(mtf01)
data(mtf.alt)
attach(mtf.alt)
mod <- HOF(mtf01, Altitude, M=1)
beta <- betadiversity(mod, Altitude)
plot(beta)
Alt.s <- gradscale(mod, Altitude, M=1)
mod.s <- HOF(mtf01, Alt.s, M=1)
plot(betadiversity(mod.s, Alt.s))

```

hillscale	<i>Hill indices of Beta diversity and gradient scaling to constant Hill index 2</i>
-----------	---

Description

The function finds two kinds of Hill indices of Beta diversity and tries to scale the gradient to constant Hill index 2 by segments.

Usage

```

hillscale(veg, grad, cycles = 4, freq.lim = 1)
betahill(veg, grad, freq.lim = 1)
## S3 method for class 'hillscale'
plot(x, which=c(1,2), xlab, ...)

```

Arguments

veg	Community data matrix.
grad	Environmental gradient.
cycles	Number of Hill scaling cycles in rescaling.
freq.lim	Frequency limit for including species.
which	Plots for indices 1 and 2.
xlab	Label for graphs. If missing, gradient name used.
x	A "hillscale" result object.
...	Other graphical parameters.

Details

Mark Hill (1979) suggested two indices of Beta diversity:

1. Mean width of species responses, measured as weighted standard deviation of gradient values.
2. Weighted variance of species scores within a site.

Hill & Gauch (1980) discuss only the former index, but the program [decorana](#) uses only the second index. Function `betahill` calculates both indices for all sample plots. In addition, the function smooths these on 20 segments along the gradient, using the same algorithm as [decorana](#) (Hill 1979, Hill & Gauch 1980).

Function `hillscale` rescales the gradient by segments using Hill index 2 (weighted variance of species scores) following as faithfully as possible the rescaling algorithm in [decorana](#). However, the function evaluates Hill index 1 (mean weighted sd of species), unlike [decorana](#).

The major difference is that [decorana](#) scales a correspondence analysis axis where site scores are direct weighted averages of species scores. Function `hillscale` uses original gradient values, but finds the species scores as weighted averages of gradient values, and expands the species scores that they have the same weighted variance as the species scores would have in correspondence analysis. If a correspondence analysis axis is given as a gradient, same species scores will be found as in [decorana](#). Another major difference is that [decorana](#) never rescales site scores. It rescales species scores instead, and always finds the site scores as direct weighted averages of species scores. Function `hillscale` rescales gradients. This difference is so significant that a rescaled correspondence analysis axis will be different in `hillscale` and [decorana](#).

Value

Both functions return an object of class "hillscale" with following items:

<code>grad</code>	Rescaled gradient in <code>hillscale</code> or the original gradient in <code>betahill</code>
<code>Hill.1</code>	Hill index 1 (mean weighted sd).
<code>Hill.2</code>	Hill index 2 (weighted variance of species scores).
<code>zv1</code>	The smoothed values of <code>Hill.1</code> on 20 segments.
<code>zv2</code>	The smoothed values of <code>Hill.2</code> on 20 segments.
<code>rug</code>	21 rug ticks equally distributed on the original gradient.
<code>cycles</code>	The number of rescaling cycles.
<code>gradname</code>	The name of the gradient variable.
<code>Call</code>	The function call.

Author(s)

Jari Oksanen

References

- Hill, M.O. (1979) *DECORANA: a FORTRAN program for detrended correspondence analysis and reciprocal averaging*. Cornell University, Ithaca, NY.
- Hill, M.O. & Gauch, J.G. (1980) Detrended correspondence analysis: an improved ordination technique. *Vegetatio* 42, 47-58.

See Also

[betadiversity](#), [gradscale](#), [decorana](#).

Examples

```
## None yet (no suitable data in the package)
```

HOF	<i>Fits Huisman-Olff-Fresco (HOF) Response Models for Species along an Ecological Gradient.</i>
-----	---

Description

Huisman-Olff-Fresco or HOF models are a series of five nested species response models which define skewed, symmetric, plateau, monotone and flat responses along ecological gradients.

Usage

```
## Default S3 method:
HOF(spec, grad, M, y.name, family = binomial, ...)
## S3 method for class 'data.frame'
HOF(veg, grad, M, freq.limit = 10, ...)
## S3 method for class 'HOF'
plot(x, ...)
## S3 method for class 'HOF.frame'
plot(x, level = 0.95, test = "F", species, ...)
## S3 method for class 'HOF'
fitted(object, model, ...)
## S3 method for class 'HOF'
residuals(object, type = c("deviance", "working",
  "response", "pearson"), model, ...)
## S3 method for class 'HOF'
predict(object, newdata, model, ...)
## S3 method for class 'HOF'
GaussPara(resp, model, ...)
```

Arguments

spec	Species data vector.
veg	Vegetation data frame.
grad	Gradient data vector.
M	Maximum attainable value in the HOF model, similar to binomial denominator.
y.name	Name of the species (stupid, but I used this in loops).
family	Error distribution. Alternatives are binomial, poisson and gaussian.
freq.limit	Lowest frequency of species analysed.
level	Probability for model selection (1-P).
test	Test for model selection. Alternatives are "F", "Chisq", "AIC", "AICc" and "BIC".

type	the type of residuals which should be returned (see residuals.glm).
species	Names of the species displayed in graphs.
x, object	An object from HOF (spec, ...).
newdata	Vector of gradient values for prediction.
resp	Fitted response models.
model	Specific HOF model used, if not selected automatically.
...	Other parameters

Details

Not yet written.

Function `fit` returns the fitted values for the used `grad`, and `predict` for any values in `newdata`.

Value

Function `HOF.fit` returns an object of class "HOF" which contains the fitting results and other useful information.

Note

These functions are at their alpha stage: proceed with caution.

Author(s)

Jari Oksanen

References

Huisman, J., Olf, H. & Fresco, L.F.M. (1993). A hierarchical set of models for species response analysis. *Journal of Vegetation Science* 4, 37-46.

Oksanen, J. & Minchin, P.R. (2002). Continuum theory revisited: what shape are species responses along ecological gradients? *Ecological Modelling* 157, 119-129.

See Also

[plotGrad](#)

Examples

```
data(mtf01)
data(mtf.alt)
attach(mtf01)
attach(mtf.alt)
mod <- HOF(BAUERUBI,Altitude, M=1)
mod
plot(mod)
mod <- HOF(mtf01, Altitude, 1)
plot(mod)
mod
```

`mtf01`*Species Data and Altitude from Mt. Field, Tasmania.*

Description

The `mtf01` data frame has 167 sites (rows) and 5 species (columns). The data are a subset of well drained sites from a more extensive data set. Data frame `mtf.alt` has only one variable: Altitude above sea level (in meters) for each site.

Usage

```
data(mtf01)
data(mtf.alt)
```

Format

The species data frame contains the following species:

EPACSERP a numeric vector
CYATPETI a numeric vector
NOTHCUNN a numeric vector
POA.GUNN a numeric vector
BAUERUBI a numeric vector

References

Minchin, P.R. (1989). Montane vegetation of the Mt. Field massif, Tasmania: a test of some hypotheses about properties of community patterns. *Vegetatio* 83, 97.110.

Examples

```
data(mtf01)
data(mtf.alt)
```

`nichelap`*Overlap of Two Species Responses along a Gradient.*

Description

Functions estimates the amount of overlap of two fitted species responses along a single gradient.

Usage

```
## S3 method for class 'HOF'
nichelap(sp1, sp2, test = "BIC", ...)
## S3 method for class 'HOF.frame'
nichelap(df, test = "BIC", ...)
## S3 method for class 'nichelap.HOF.frame'
as.matrix(x, ...)
```

Arguments

sp1, sp2	Fitted response models for two species.
df	Fitted responses for a data frame of several species.
test	The test used to select the HOF model.
x	The result of nichelap for a data frame of several species.
...	Other arguments passed to functions.

Details

The function finds the niche overlap as an overlap of fitted response curves (Lawesson & Oksanen 2002). The input can either consist of fitted responses for two species, or of responses fitted to several species in a data frame. If the input is a frame of several fitted responses, overlaps will be found for each pair of species.

For each pair of species, the function will return a vector with total areas for each species, the total area of the overlap, the proportion of overlap from the total covered jointly by two species, and the proportion of each species covered by the other species. The total area is defined so that the the maximum response height and total gradient range is unity. Consequently, the reported areas are proportions of the maximum attainable area (species occurs at its attainable maximum over the whole gradient range). Function `as.matrix` returns the last two entries in a matrix similar to Lawesson & Oksanen (2002), table 4.

The functions are based on the standard R\ function [integrate](#).

Value

Function returns a vector of niche statistic for each pair of species. For a response frame, these vectors are each an item in a list.

Note

At the moment, the function knows only HOF models.

Author(s)

Jari Oksanen

References

Lawesson, J.E. & Oksanen, J. (2002). Niche characteristics of Danish woody species as derived from coenoclines. *Journal of Vegetation Science* 13, 279–290.

See Also

[integrate](#), [HOF](#).

Examples

```
data(mtf01)
data(mtf.alt)
attach(mtf.alt)
mods <- HOF(mtf01, Altitude, M=1)
lap <- nichelap(mods)
lap
mat <- as.matrix(lap)
mat
# Similar formatting as in Lawesson & Oksanen (2002)
round(100*mat)
```

plotGrad

Plots Species gainst an Environmental Gradient

Description

Plots each species in a data frame against an environmental gradient using trellis graphics.

Usage

```
plotGrad(x, grad, freq.limit = 0, ...)
```

Arguments

x	Data frame with species data.
grad	Gradient vector.
freq.limit	Lowest frequency for species selected for plotting
...	Other parameters to Lattice functions.

Details

Function plotGrad is intended for preliminary inspection of the data. It plots each species in a data frame against an environmental gradient using [Lattice](#) functions. If grad is numeric, it uses scatter plots ([xyplot](#)) and if grad is a factor, it uses boxplots ([bwplot](#)).

All species are plotted with equal vertical scale, making scarce species very flat. Users can change this behaviour by transforming or standardizing species data before analysis (for instance, [decostand](#)) has options "max" for making all species to equal scale).

The function uses trellis graphics in package [lattice](#), and transfers all parameters to underlying functions [xyplot](#) and [bwplot](#). Moreover, many graphical parameters can be changed after producing the graph (see [trellis.par.set](#)).

Value

Function returns a "trellis" object.

Author(s)

Jari Oksanen

See Also

See [Lattice](#) basics of Trellis graphics, [xyplot](#) and [bwplot](#) for underlying plotting functions, and [trellis.par.set](#) for setting graphical parameters.

Examples

```
data(mtf01, mtf.alt)
attach(mtf.alt)
fig <- plotGrad(mtf01, Altitude)
fig
library(lattice)
trellis.par.set(theme = col.whitebg())
fig
```

Index

- * **datasets**
 - mtf01, 11
- * **dplot**
 - boxgradient, 3
 - plotGrad, 13
- * **models**
 - betadiversity, 2
 - boxgradient, 3
 - GaussPara, 4
 - gradscale, 6
 - hillscale, 7
 - HOF, 9
 - nichelap, 11
- * **nonlinear**
 - HOF, 9
- * **regression**
 - GaussPara, 4
 - HOF, 9
- abbreviate, 3
- as.matrix.nichelap.HOF.frame
(nichelap), 11
- axis, 4
- betadiversity, 2, 6, 8
- betahill (hillscale), 7
- boxgradient, 3
- boxplot, 3, 4
- bwplot, 13, 14
- decorana, 8
- decostand, 13
- family, 3
- fitted.HOF (HOF), 9
- gaussgradient (boxgradient), 3
- GaussPara, 4
- GaussPara.HOF (HOF), 9
- glm, 3, 4
- gradscale, 6, 8
- hillscale, 7
- HOF, 2, 3, 5, 9, 13
- integrate, 12, 13
- Lattice, 13, 14
- mtf.alt (mtf01), 11
- mtf01, 11
- nichelap, 11
- optimize, 5
- par, 3, 4
- plot.betadiversity (betadiversity), 2
- plot.hillscale (hillscale), 7
- plot.HOF (HOF), 9
- plotGrad, 10, 13
- polygon, 3, 4
- predict.HOF (HOF), 9
- residuals.glm, 10
- residuals.HOF (HOF), 9
- rug, 4
- spline, 2
- trellis.par.set, 13, 14
- uniroot, 5
- xyplot, 13, 14