

Package ‘gsg’

February 20, 2015

Type Package

Title Calculation of selection coefficients

Version 2.0

Date 2014-09-03

Author Michael Morrissey, Krzysztof Sakrejda

Maintainer Michael Morrissey <michael.morrissey@st-andrews.ac.uk>

Imports mgcv, mvtnorm, boot, numDeriv

Description gsg (gam selection gradients) provides a unified approach to the regression analysis of selection from longitudinal data collected from natural populations.

License GPL-2

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2014-10-17 00:15:03

R topics documented:

gsg-package	2
fitness.landscape	2
gam.gradients	4
gppr	6
gppr.gradients	8
humanNeonatal	9
moments.differentials	10
predict.gppr	11
SoayLambs	12
Index	14

gsg-package

Calculation of selection coefficients

Description

Functions to calculate selection gradients, especially `gam.gradients()` which allows selection gradients to be calculated from a generalized additive model-based characterization of a fitness function.

Details

Package: gsg
Type: Package
Version: 1.0
Date: 2012-02-23
License: GPL-2
LazyLoad: yes

Author(s)

Michael Morrissey

Maintainer: Michael Morrissey <michael.morrissey@st-andrews.ac.uk>

References

M.B. Morrissey and K. Sakrejda. 2013. Unification of regression-based methods for the analysis of natural selection. *Evolution* 67: 2094-2100.

M.B. Morrissey. in prep.

fitness.landscape

A function to estimate fitness landscapes, i.e. the relationship between population mean fitness and population mean phenotype.

Description

`fitness.landscape()` takes a fitness function, i.e., the relationship between fitness and individual phenotype, as characterized using the function `gam()` from the package `mgcv`, and calculates a corresponding fitness landscape in one or two dimensions

Usage

```
fitness.landscape(mod, phenotype, covariates = NULL, points = NULL,
  plt.density = 25, PI.method = "boot.para",
  PI.interval = c(0.25, 0.75), n.boot = 50, refit.smooth = FALSE,
  parallel = "no", ncpus = 1)
```

Arguments

<code>mod</code>	a gam object. Must include predictor variables specified by phenotype
<code>phenotype</code>	a vector of one or two character strings specifying predictor variables in <code>mod</code> that are the phenotypic traits with respect to which the fitness landscape is to be calculated
<code>covariates</code>	a character vector listing any covariates in the model <code>mod</code> to be excluded from characterization of the fitness landscape
<code>points</code>	(optional) a matrix, array, or data frame with traits in columns and points in rows, specifying the series of points at which to evaluate the estimated fitness landscape. If missing, the fitness landscape is evaluated at points from one standard deviation below, to one standard deviation above the mean values, at as many points as specified by <code>plt.density</code>
<code>plt.density</code>	(optional) the number of points (per phenotypic trait) at which to evaluate the fitness landscape. only used if <code>points</code> is NULL
<code>PI.method</code>	the method by which to obtain a prediction interval for the fitness landscape. Options are 'n' for none, 'boot.para' for parametric bootstrapping, and 'boot.case' for case bootstrapping.
<code>PI.interval</code>	the upper and lower bounds of the prediction interval. Defaults to <code>c(0.5, 0.75)</code> in order to generate a prediction interval that is interpretable similarly to a standard error.
<code>n.boot</code>	number of bootstrap replicates for evaluating the prediction interval of the fitness landscape
<code>refit.smooth</code>	whether or not to re-estimate smoothing parameters when gam objects are refitted in bootstrapping and permutation algorithms
<code>parallel</code>	whether or not to use parallel processing to speed up computation of bootstrap prediction intervals. Default in 'no' for no parallel computing. Under linux only, <code>parallel='multicore'</code> allows parallel processing, using the number of processors specified by <code>ncpus</code>
<code>ncpus</code>	the number of cpus to be used for parallel processing of bootstrap prediction intervals. Only used under linux.

Value

<code>\$points</code>	the points at which the fitness landscape was evaluated
<code>\$Wbar</code>	population mean absolute fitness at points
<code>\$WbarPI</code>	bounds of the prediction interval of <code>Wbar</code> at points

Note

Fitness landscapes are hypothetical constructs used to provide a visual representation of the magnitude of selection. Care must be taken in the interpretation of these landscapes as the increase population mean fitness that would actually be realized from a given amount of selection. This is because there are many assumptions, such as independence of evolution of the mean and variance, which are in fact likely to scale together, lack of density dependence of population growth (i.e. fitness), the assumption that evolution of the trait itself does not change the environment, etc.

Author(s)

Michael Morrissey <michael.morrissey@st-andrews.ac.uk>

References

M.B. Morrissey and K. Sakrejda. 2013. Unification of regression-based methods for the analysis of natural selection. *Evolution* 67: 2094-2100.

See Also

[gam.gradients](#), [moments.differentials](#), [gam](#)

Examples

```
# simulated data (stabilizing selection)
z<-rnorm(200,0,2)
W<-rpois(200,exp(1-1*z^2))
d<-as.data.frame(list(W=W,z=z))

# characterize the fitness function
library(mgcv)
ff<-gam(W~s(z),family='poisson',data=d)

# characterize fitness landscape
fl<-fitness.landscape(mod=ff,phenotype="z",PI.method='n')

# (not run) plotting
# plot(fl$points[,1],fl$Wbar,type='l')
# lines(fl$points[,1],fl$WbarPI[1,],lty='dashed')
# lines(fl$points[,1],fl$WbarPI[2,],lty='dashed')
```

gam.gradients

A function to calculate selection gradients from generalized additive model-based characterizations of fitness functions.

Description

gam.gradients uses numerical approximations to the first and second order partial derivatives of population mean fitness with respect to population mean phenotype to obtain directional and quadratic selection gradients.

Usage

```
gam.gradients(mod, phenotype, covariates = NULL, standardized = FALSE,
              se.method = "boot.para", n.boot = 1000,
              parallel = "no", ncpus = 1, refit.smooth = FALSE)
```

Arguments

mod	a gam object. Must include predictor variables specified by phenotype
phenotype	a vector of one or two character strings specifying predictor variables in selection gradients are to be calculated
covariates	a character vector listing any covariates in the model mod for which selection gradients are not to be calculated
se.method	the method by which to obtain standard errors and P-values of the selection gradients. Options are 'n' for none, 'boot.para' for parametric bootstrapping (default), 'boot.case' for case bootstrapping, 'posterior' for an algorithm based on simulation from the multivariate normal approximation to the posterior distribution of the model parameters, and 'permute' for permutation-based P-values (no SEs).
n.boot	number of bootstrap replicates for evaluating statistical uncertainty in the selection gradients.
standardized	whether or not to standardize to unit variance to obtain selection gradient estimates that are comparable across traits and populations
parallel	whether or not to use parallel processing to speed up computation of bootstrap standard errors. Default in 'no' for no parallel computing. Under linux only, parallel='multicore' allows parallel processing, using the number of processors specified by ncpus
ncpus	the number of cpus to be used for parallel processing of bootstrap standard errors. Only used under linux.
refit.smooth	whether or not to re-estimate smoothing parameters when gam objects are refitted in bootstrapping and permutation algorithms

Value

A list. Element 1 is a table of estimates, standard errors and P-values, and element 2 contains the bootstrap or permutation values.

Author(s)

Michael Morrissey <michael.morrissey@st-andrews.ac.uk>

References

M.B. Morrissey and K. Sakrejda. 2013. Unification of regression-based methods for the analysis of natural selection. *Evolution* 67: 2094-2100.

See Also

[gam.gradients](#), [moments.differentials](#), [gam](#)

Examples

```
# simulated data (stabilizing selection)
z<-rnorm(200,0,2)
W<-rpois(200,exp(1-0.3*z^2))
d<-as.data.frame(list(W=W,z=z))

# characterize the fitness function
library(mgcv)
ff<-gam(W~s(z),family='poisson',data=d)

# derive selection gradients
gam.gradients(mod=ff,phenotype="z",se.method='n',standardized=FALSE)$ests

# or gam.gradients() can be used to do the
# equivalent of a basic Lande and Arnold
# 1983 regression

LA<-gam(W~z+I(z^2),family='gaussian',data=d)
gam.gradients(mod=LA,phenotype="z",se.method='n',standardize=FALSE)$ests
```

gppr

Generalised projection pursuit regression

Description

Performs generalized projection pursuit regression for binary and Poisson response variables. Wraps ppr from package stats in an iterative re-weighting algorithm. Included in package gsg to support inference of the form of natural selection, and for inference of selection gradients from projection pursuit regression-based inference of fitness functions.

Usage

```
gppr(y, xterms, data, nterms = 1, tol = 0.001, gcvcpen = 1,
      maxit = 50, family = 'binomial', max.terms = 2)
```

Arguments

y	response variable, e.g., absolute fitness
xterms	names of predictor variables in data
data	a data frame containing columns with names matching xterms
nterms	number of terms to include in the final model
tol	maximum sum of relative changes in linear predictors between iterations

gcvpen	penalty used for each degree of freedom in GCV selection for spline ridge functions
maxit	maximum number of iterations of the iterative re-weighting procedure
family	distribution of the response variable, currently 'binomial' and 'Poisson' are supported
max.terms	the maximum number of terms to choose from when building the model
...	other parameters passed to ppr

Value

A gppr object, which contains a ppr object, plus information pertaining to the iterative re-weighting procedure. Most of these repeat arguments passed to gppr(), also included are:

iterations	the number of iterations of the iterative re-weighting procedure that were performed
f	the formula passed to ppr, constructed from xterms

Author(s)

Michael Morrissey <michael.morrissey@st-andrews.ac.uk>

References

M.B. Morrissey. in prep.

See Also

[predict.gppr](#), [ppr](#), [gppr.gradients](#)

Examples

```
# simulated data (two traits, stabilizing selection on trait 1)
n<-250
z<-cbind(rnorm(n,0,1),rnorm(n,0,1))
W<-rpois(n,exp(2-0.6*z[,1]^2))
d<-as.data.frame(cbind(W,z))
names(d)<-c("W", "z1", "z2")

fit.func<-gppr(y="W",xterms=c("z1","z2"),data=d,family="poisson",
  nterms=2,max.terms=2)

# direction of axes, and their loadings
fit.func$alpha
fit.func$beta

## not run - plot ridge functions (linear predictor scale)
# par(mfrow=c(1,2))
# plot(fit.func)
```

gppr.gradients	<i>A function to calculate selection gradients from generalized additive model-based characterizations of fitness functions.</i>
----------------	--

Description

gam.gradients uses numerical approximations to the first and second order partial derivatives of population mean fitness with respect to population mean phenotype to obtain directional and quadratic selection gradients.

Usage

```
gppr.gradients(mod, phenotype, covariates = NULL, standardized = FALSE,
               se.method = "boot.para", n.boot = 1000,
               parallel = "no", ncpus = 1)
```

Arguments

mod	a fitted object produced by gppr . Must include predictor variables specified by phenotype
phenotype	a vector of one or two character strings specifying predictor variables in selection gradients are to be calculated
covariates	a character vector listing any covariates in the model mod for which selection gradients are not to be calculated
se.method	the method by which to obtain standard errors and P-values of the selection gradients. Options are 'n' for none, 'boot.para' for parametric bootstrapping (default), 'boot.case' for case bootstrapping, 'posterior' for an algorithm based on simulation from the multivariate normal approximation to the posterior distribution of the model parameters, and 'permute' for permutation-based P-values (no SEs).
n.boot	number of bootstrap replicates for evaluating statistical uncertainty in the selection gradients.
standardized	whether or not to standardize to unit variance to obtain selection gradient estimates that are comparable across traits and populations
parallel	whether or not to use parallel processing to speed up computation of bootstrap standard errors. Default in 'no' for no parallel computing. Under linux only, parallel='multicore' allows parallel processing, using the number of processors specified by ncpus
ncpus	the number of cpus to be used for parallel processing of bootstrap standard errors. Only used under linux.

Value

A list. Element 1 is a table of estimates, standard errors and P-values, and element 2 contains the bootstrap or permutation values.

Author(s)

Michael Morrissey <michael.morrissey@st-andrews.ac.uk>

References

M.B. Morrissey. in prep.

See Also

[gam.gradients](#), [gppr](#)

Examples

```
# simulated data (two traits, stabilizing selection on trait 1)
n<-250
z<-cbind(rnorm(n,0,1),rnorm(n,0,1))
W<-rpois(n,exp(2-0.6*z[,1]^2))
d<-as.data.frame(cbind(W,z))
names(d)<-c("W", "z1", "z2")

fit.func<-gppr(y="W",xterms=c("z1", "z2"),data=d,family="poisson",
  nterms=2,max.terms=2)

gppr.gradients(mod= fit.func,phenotype=c("z1", "z2"),se.method='n',standardize=FALSE)
```

humanNeonatal

Human male birth mass, gestation length, and neonatal survival

Description

Karn and Penrose's (1951) dataset on the survival of human male infants, their birth weights in 0.23 kg (half pound) increments, and their gestation lengths in 5 day increments.

Usage

```
humansNeonatal
```

Format

a data frame

Source

Transcribed by Dolph Schluter from Karn and Penrose 1951.

References

- Karn, M. N., and L. S. Penrose. 1951. Birth weight and gestation time in relation to maternal age, parity and infant survival. *Annals fo Eugenics* 16: 147-164.
- Schluter, D. 1988. Estimating the form of natural selection on a quantitative trait. *Evolution* 42: 849-861.
- Schluter, D., and D. Nychka. 1994. Exploring fitness surfaces. *The american Naturalist* 143: 597-616.
- M.B. Morrissey and K. Sakrejda. 2013. Unification of regression-based methods for the analysis of natural selection. *Evolution* 67: 2094-2100.

`moments.differentials` *Calculates selection differentials directly from the differences in phenotypic moments before and after selection.*

Description

The calculations are based on the difference between the fitness-weighted moments (means, variances, covariances) of the phenotypic distribution and the unweighted moments. For viability selection, this is simply the moments after minus the moments before, as there are two fitness classes, zero and one.

Usage

```
moments.differentials(z, W, n.boot = 1000, standardized = FALSE)
```

Arguments

<code>z</code>	phenotypic traits, a data frame, table, or array with individuals on rows and traits in columns
<code>W</code>	a vector of fitness (relative or absolute) of the same length as the number of individuals represented in <code>z</code>
<code>n.boot</code>	number of bootstrap replicates to use for calculation of standard errors and P-values
<code>standardized</code>	whether or not to calculate variance-standardized selection differentials

Value

A table containing the estimated directional and quadratic selection gradients, and if bootstrap-based standard errors are generated, columns containing those standard errors and associated P-values

Author(s)

Michael Morrissey <michael.morrissey@st-andrews.ac.uk>

References

M.B. Morrissey and K. Sakrejda. 2013. Unification of regression-based methods for the analysis of natural selection. *Evolution* 67: 2094-2100.

See Also

[gam.gradients](#)

Examples

```
# simulated data (stabilizing selection)
z<-rnorm(200,0,2)
W<-rpois(200,exp(1-0.3*z^2))
d<-as.data.frame(list(W=W,z=z))

# estimate selection coefficients by least-squares regression
moments.differentials(z=z,W=W)
# note that standardized differentials and gradients are trivially
# the same in a univariate analysis
```

predict.gppr

Generalised projection pursuit regression

Description

Obtains linear predictor- or data-scale predictions for a fitted gppr object

Usage

```
predict.gppr(object,newdata,type="link")
```

Arguments

object	a fitted gppr object
newdata	an optional data frame containing new predictor variable values. If 'newdata' is omitted, then the predictions are made for the original data from which the gppr model was fitted.
type	the type of prediction required. The default is the scale of the linear predictor, alternatives 'type="response"' gives expected values.

Value

A vector of predictions.

Author(s)

Michael Morrissey <michael.morrissey@st-andrews.ac.uk>

References

M.B. Morrissey. in prep.

See Also

[gppr](#), [ppr](#), [gppr.gradients](#)

Examples

```
# simulated data (two traits, stabilizing selection on trait 1)
n<-250
z<-cbind(rnorm(n,0,1),rnorm(n,0,1))
W<-rpois(n,exp(2-0.6*z[,1]^2))
d<-as.data.frame(cbind(W,z))
names(d)<-c("W", "z1", "z2")

fit.func<-gppr(y="W",xterms=c("z1", "z2"),data=d,family="poisson",
  nterms=2,max.terms=2)

# hist(predict(fit.func))
```

SoayLambs

Soay sheep lamb phenotypic data and first year winter survival

Description

Data for mass, hind leg length, horn length, and (log) ked count of Soay sheep measured in August. Fitness (W) is first-winter survival. Data are subsetting to females with normal horns in cohorts that did not experience crash conditions in their first winter. Phenotypic data are mean-centred and variance-standardized.

Usage

SoayLambs

Format

a data frame

Source

Data provided by Profs. J.M. Pemberton and L.E.B. Kruuk, University of Edinburgh. The data were collected primarily by Jill Pemberton and Andrew MacColl with the help of many volunteers, with logistical support from the National Trust for Scotland and QinetQ, and with funding from NERC, Royal Society, and the Leverhulme Trust.

References

Clutton-Brock, T.H., and J.M. Pemberton. 2004. *Soay sheep dynamics and selection in an island population*. Cambridge University Press, Cambridge.

Morrissey, M. B. in prep.

Index

*Topic **\textasciitildekwd1**

fitness.landscape, 2

gam.gradients, 4

gppr.gradients, 8

*Topic **\textasciitildekwd2**

fitness.landscape, 2

gam.gradients, 4

gppr.gradients, 8

*Topic **datasets**

humanNeonatal, 9

SoayLambs, 12

*Topic **package**

gsg-package, 2

fitness.landscape, 2

gam, 4, 6

gam.gradients, 4, 4, 6, 9, 11

gppr, 6, 8, 9, 12

gppr.gradients, 7, 8, 12

gsg (gsg-package), 2

gsg-package, 2

humanNeonatal, 9

moments.differentials, 4, 6, 10

ppr, 7, 12

predict.gppr, 7, 11

SoayLambs, 12