Package ‘perspectev’

May 13, 2015

Type Package
Title Permutation of Species During Turnover Events
Version 1.0
Date 2015-05-02
Maintainer Kenneth B. Hoehn <perspectev@gmail.com>
Depends R (>= 2.10), ggplot2, foreach, ape, doParallel, parallel
Imports mapproj, boot, sp
Description Provides a robust framework for analyzing the extent to which differential survival with respect to higher level trait variation is reducible to lower level variation. In addition to its primary test, it also provides functions for simulation-based power analysis, reading in common data set formats, and visualizing results. Temporarily contains an edited version of function hr.mcp() from package ‘wild1’, written by Glen Sargeant.
License GPL-3
NeedsCompilation no
Author Kenneth B. Hoehn [aut, cre],
Glen A. Sargeant [ctb]
Repository CRAN
Date/Publication 2015-05-13 19:53:39

R topics documented:

perspectev-package ..................................................... 2
hr.mcp ................................................................. 2
perspectev-Datasets .................................................... 3
perspectev.calc ......................................................... 3
perspectev.plot ......................................................... 4
perspectev.read ......................................................... 5
perspectev.simulate ................................................... 6
perspectev.test ......................................................... 7

Index 9
Description

Provides a robust framework for analyzing the reducibility of differential survival with respect to higher level trait variation. In addition to its primary test, it also provides functions for simulation-based power analysis, reading in common data set formats, and visualizing results.

Details

Table:

- Package: perspectev
- Type: Package
- Version: 1.0
- Date: 2015-05-02
- License: GPL-2

Author(s)

Kenneth B. Hoehn <perspectev@gmail.com>

hr.mcp

Home range estimation

Description

Compute and plot a minimum convex polygon estimate of "home range." Source code borrowed from wild1 package.

Usage

hr.mcp(x, y=NULL, n.min=50, plot=TRUE, add=FALSE, ID=NULL, ...)

Arguments

- x: A vector, matrix, or data frame of coordinates.
- y: A numeric vector (if x is a vector) or NULL
- n.min: If x includes fewer than n.min points, the function will return NA and a warning.
- plot: Parameter nonfunctional for perspectev implementation of this function.
- add: Logical. Add to existing plot on current output device? Default is FALSE.
- ID: Required argument from Polygons
- ...: Optional arguments for plot or points (see also par).
Details
See plot.Polygons for another way of plotting polygon objects.

Value
Returns an object of class Polygons (sp).

Author(s)
Glen A. Sargeant
U.S. Geological Survey
Northern Prairie Wildlife Research Center
<glen_sargeant@usgs.gov>
Edited by Kenneth B. Hoehn
<perspectev@gmail.com>

perspectev-Datasets Supplied Data Sets

Description
Occurrence data used in perspectev tutorial, and test data set used for checking.

Details
occurrence data: See online tutorial for further information.

perspectev.calc Calculate upper and lower level trait values and survivorship

Description
Calculates upper and lower level trait values and survivorship given a data set and trait functions

Usage
perspectev.calc(data,traitfun,vlist=list())

Arguments
data Dataframe in perspectev format (see below).
traitfun Function for calculating trait value at each level.
vlist Optional variable list for trait function.
Value

upper  Table of survivorship and trait value at upper level (scaled to mean 0 and variance 1)
lower  Table of survivorship and trait value at lower level (scaled to mean 0 and variance 1)
stats  [1]: correlation between upper level trait and survivorship, [2-4]: quantiles of upper level trait values (unscaled)

Author(s)

Kenneth B. Hoehn <perspectev@gmail.com>

perspectev.plot  Plotting function for permutation analysis

Description

Make plots to summarize permutation results and power analysis results

Usage

perspectev.plot( observed, simulated, names, title )

Arguments

observed Object returned from perspectev.test().
simulated List of up to eight objects obtained from perspectev.simulate().
names Names of each object in the list provided to "simulated" in order.
title Title of the plot. Optional.

Author(s)

Kenneth B. Hoehn <perspectev@gmail.com>
**Description**

Convert occurrence data (each row a separate occurrence) into standardized perspectev format.

**Usage**

```r
perspectev.read(data, extinctionAge, occurrenceAge, upper, lower, t1, t2,
                 traitfun = mcpRange, vlist = NULL, trim = TRUE, projection = FALSE)
```

**Arguments**

- `data`: Occurrence data.
- `occurrenceAge`: Column designation for the age of the occurrence.
- `extinctionAge`: Age of the extinction. E.g. 65.5 for K-Pg mass extinction if occurrenceAge is in mya.
- `upper`: Column designation for upper level names.
- `lower`: Column designation for lower level names.
- `t1`: Column designation for trait-data 1.
- `t2`: Column designation for trait-data 2.
- `trim`: Remove lower levels that give NA's from trait function (recommended)?
- `projection`: Convert coordinates to a Mollweide projection?
- `traitfun`: Function for calculating trait values at each level. Not used if trim==FALSE.
- `vlist`: Optional variable list for trait function.

**Details**

See tutorial for details. Note that this will assign any lower level entity found both before/during and after the extinctionAge parameter will be listed as a survivor, while any lower level entity only found in the beginning will be listed as a victim. It is important to ensure your data set does not span too far back, or many of the species that went extinct before the extinction will be listed as victims. This is not good.

If survivorship is to be given probabilistically rather than as a binary, run this function with extinctionAge=0, and input the survivorship probabilities into the Survivorship column of the resulting data frame, which lists each unique species on a separate row.

**Author(s)**

Kenneth B. Hoehn <perspectev@gmail.com>
Examples

data(testData)

data = perspectev.read(testData,extinctionAge=5,occurrenceAge="Age",upper="Genus",lower="Species",t1="Lat",t2="Long",traitfun=mcpRange,projection=FALSE)

perspectev.simulate  Simulate species/genus level selection at different intensities and observe resulting p values.

Description

Perform simulations of natural selection on a data set, and observe what p values result. Perform linear transformation of trait vs. survivorship (using provided or fitted slope and intercept) and then transform into survivorship probabilities using inv.logit function.

Usage

perspectev.simulate(data,simulations,cores,traitfun=mcpRange,vlist=NULL,binary=NA,intercept=NA,slope=NA,level=NA,noise=0,fit=FALSE)

Arguments

data  Dataframe in perspectev format. See ?perspectev.read for more information.
simulations  Number of simulations to perform. At least 1000 is recommended but can be slow.
cores  Number of cores over which to parallelize the script.
traitfun  Function for calculating trait value at each level.
vlist  Optional variable list for trait function.
intercept  Intercept of the regression between range and survivorship. Can be manually provided or, if not provided, is fit to the data set. Will attempt to set automatically if unspecified.
slope  Slope of the regression between range and survivorship. Can be manually provided or, if not provided, is fit to the data set. Will attempt to set automatically if unspecified.
level  Simulate either upper or lower level selection. Accepts either "lower" or "upper" respectively. Will set to ‘upper’ if unspecified.
noise  Extra stochasticity to the simulation.
fit  Fit simulation to observed parameters? Will do automatically if intercept or slope are not specified.
binary  Is survivorship binary or continuous? Will attempt to set automatically if unspecified.
Value

- **correlation_simulated**: Correlation between trait and survivorship from permuted groups at each simulation.
- **correlation_observed**: Observed correlation between trait and survivorship from simulated groups.
- **pvalue**: p value observed under simulation of selection.
- **level**: Level of simulated selection.
- **intercept**: Intercept of simulation.
- **slope**: Slope of the simulation.
- **noise**: Noise level selected for simulation.
- **fitted_model**: Was the model fitted to the data set?

Note

See original publication and tutorial for more information on simulation.

Author(s)

Kenneth B. Hoehn <perspectev@gmail.com>

---

**perspectev.test**

Test for irreducibility of relationship between higher level traits and survivorship

Description

Performs permutation tests by permuting upper level labels between lower levels, recalculating upper trait value, and taking the correlation between upper level trait and survivorship. This process is repeated until a null distribution is generated. This is then compared against observed covariance to give a p value for the null hypothesis that a relationship between trait and survivorship is explainable by random aggregations of lower level traits.

Usage

```r
perspectev.test(data, iterations=1000, cores=1, traitfun=mcpRange, vlist=NULL)
```

Arguments

data: Dataframe in perspectev format (see below).
iterations: Number of iterations to perform. At least 1000 is recommended, though can be slow.
cores: Number of cores over which to parallelize the test.
traitfun: Function for calculating trait values at each level.
vlist: Optional variable list for trait function.
Value

- **correlation_permuted**: Correlations between trait and survivorship obtained from permuted upper levels (Si)
- **correlation_observed**: Observed correlation between upper level trait and survivorship (Ri)
- **pvalue**: Portion of permuted genus correlations (S) larger than observed value (R)
- **permuted_quantiles**: Matrix of interquartile trait values obtained from each upper level permutation

Author(s)

Kenneth B. Hoehn <perspectev@gmail.com>

Examples

```r
data(testData)

data = perspectev.read(testData, extinctionAge=5, occurrenceAge="Age", upper="Genus", lower="Species", t1="Lat", t2="Long", traitfun=mcpRange, projection=FALSE)

# 4 iterations chosen out of convenience - use more!
mcptest = perspectev.test(data, 4, 1, traitfun=mcpRange)
mcpsim = perspectev.simulate(data, 4, 1, traitfun=mcpRange)
perspectev.plot(mcptest, list(mcpsim), c("S1", "Test"))
```
Index

*Topic aplot
  hr.mcp, 2
*Topic data
  perspectev-Datasets, 3
*Topic hplot
  hr.mcp, 2
*Topic package
  perspectev-package, 2

hr.mcp, 2

occurrence_data (perspectev-Datasets), 3

par, 2
perspectev (perspectev-package), 2
perspectev-Datasets, 3
perspectev-package, 2
perspectev.calc, 3
perspectev.plot, 4
perspectev.read, 5
perspectev.simulate, 6
perspectev.test, 7
plot, 2
points, 2

testData (perspectev-Datasets), 3