Package 'evobiR'

October 13, 2022

Type Package

Title Comparative and Population Genetic Analyses

Version 1.1

Date 2015-8-25

Author Heath Blackmon and Richard H. Adams

Maintainer Heath Blackmon <coleoguy@gmail.com>

URL <http://www.uta.edu/karyodb/evobiR/>

Description Comparative analysis of continuous traits influencing discrete states, and utility tools to facilitate comparative analyses. Implementations of ABBA/BABA type statistics to test for introgression in genomic data. Wright-Fisher, phylogenetic tree, and statistical distribution Shiny interactive simulations for use in teaching.

License GPL (>= 2)

Imports seqinr, ape, geiger, shiny, phytools

NeedsCompilation no

Repository CRAN

Date/Publication 2015-09-06 19:30:55

R topics documented:

---

evobiR-package   *evobiR: Evolutionary Biology in R*

Description

evobiR is a collection of tools for use in evolutionary biology. Some of the functions manipulate data in a way not implemented by other functions while others calculate sequence statistics or perform simulations, either of data across trees or genetic and genomic simulations.

Details

Package:	evobiR
Type:	Package
Version:	1.1
Date:	2013-10-08
License:	GPL (>=2)

More information on **evobiR** is available at <http://coleoguy.github.io/software.html>

Author(s)

Heath Blackmon

Maintainer: Heath Blackmon <coleoguy@gmail.com>

1.fasta   *simulated SNP data*

Description

This file contains simulated SNP data

Author(s)

Heath Blackmon

References

http://coleoguy.github.io/

AICc   *Computes an AICc score*

Description

Supplied with a log likelihood, the number of model parameters, and sample size calculates the small sample size version of the AIC score.

Usage

```
AICc(loglik, K, N)
```

Arguments

loglik	log likelihood.
K	the number of parameters in the model
N	the sample size.

Details

Returns an AICc score.

Author(s)

Heath Blackmon

References

<http://coleoguy.github.io/>

Examples

```
AICc(-32, 3, 100)
```

AncCond   *Calculate the mean of a continuous character at the origin of derived*
                                        state of a binary charachter

---

Description

This function uses stochastic mapping and ancestral state reconstruction to determine if the derived
state of a binary trait originates when a continuous trait has an extreme value.

Usage

```
AncCond(trees, data, derived.state, iterations=1000)
```

Arguments

trees	tree(s) of class phylo or multiPhylo
data	a dataframe with 3 columns. The first should match the taxa names in the tree, the second should have the continuous trait values and the third the states for the binary character
derived.state	the derived condition for the binary trait
iterations	the number of iterations to be used in estimating significance

Details

This function uses stochastic mapping and ancestral state reconstruction as implemented in phytools
to determine if the derived state of a binary trait originates when a continuous trait has an extreme
value. This test assumes that the derived state of the binary character may lead to correlated selec-
tion in the continuous trait. Because of this the ancestral state reconstruction of the continuous trait
is based only on data from species that remain in the ancestral condition for the binary trait

Value

Returns a plot of the null distribution and the observed data as well as empirical p-value for the
observed data.

Author(s)

Heath Blackmon and Richard H. Adams

References

http://coleoguy.github.io/

Examples

```
## Not run:
data(mite.trait)
data(trees.mite)
AncCond(trees, mite.trait, derived.state = "haplodiploidy", iterations=100)

## End(Not run)
```

CalcD   *Calculate Patterson's D-statistic*

Description

These functions calculate Patterson's D-statistic to compare the frequencies of discordant SNP genealogies. These tests assume equal substitution rates and unlinked loci, D-statistics significantly different from 0 suggest that introgression has occurred.

Usage

```
CalcD(alignment = "alignment.fasta", sig.test = "N", block.size = 1000, replicate = 1000)

CalcPopD(alignment = "alignment.fasta")
```

Arguments

alignment	This is an alignment in fasta format. Sequences should be in the order: P1, P2, P3, Outgroup.
sig.test	This indicates whether or if to test for significance. Options are "B" bootstrap, "J" jackknife, or "N" none.
block.size	The number of sites to be dropped in the jackknife approach
replicate	Number of replicates to be used in estimating variance

Details

The functions CalcD and CalcPopD are implementations of the algorithm described in Durand et al. 2011. Significance of the D-stat can be calculated either through bootstrapping or jackknifing. Bootstrapping is appropriate for datasets where SNPs are unlinked for instance unmapped RADSeq data. Jackknifing is the appropriate approach when SNPs are potentially in linkage for instance gene alignments or genome alignments.

Value

Returns the number of each type of site, Z scores and p-values

Author(s)

Heath Blackmon

References

http://coleoguy.github.io/

Durand, Eric Y., et al. Testing for ancient admixture between closely related populations. Molecular biology and evolution 28.8 (2011): 2239-2252.

Eaton, D. A. R., and R. H. Ree. 2013. Inferring phylogeny and introgression using RADseq data: An example from flowering plants (Pedicularis: Orobanchaceae). Syst. Biol. 62:689-706

Examples

```
CalcD(alignment = system.file("1.fasta", package = "evobiR"), sig.test = "N")

CalcPopD(alignment = system.file("3.fasta", package = "evobiR"))
```

Even   *Tests whether a number is even*

Description

Just a simple function that returns True if a number is even and False otherwise.

Usage

```
Even(x)
```

Arguments

x   a numerical vector.

Details

Returns a vector of logical values of the same length as the input vector. If the input value is not a number it will return an error message.

Author(s)

Heath Blackmon

References

<http://coleoguy.github.io/>

Examples

```
Even(c(1,2,3,4,5,6,2,5))
```

FuzzyMatch   *Find Close Matches in a tree and dataset*

Description

When assembling data from different sources typos can sometimes cause a loss of perfect matches between trees and datasets. This function helps you find these close matches that can be hand curated to keep as many species as possible in your analysis.

Usage

```
FuzzyMatch(tree, data, max.dist)
```

Arguments

tree	a phylogenetic tree of the class "phylo".
data	character vector with the names from your dataset.
max.dist	This is the maximum number of characters that can differ between your tree and data and still be recognized as a close match.

Value

A dataframe with the following rows:

Name in data
Name in tree
Number of differences

Author(s)

Heath Blackmon

References

http://coleoguy.github.io/

Examples

```
data(hym.tree)
names <- c("Pepsis_elegans", "Plagiolepis_alluaudi", "Pheidele_lucreti",
          "Meliturgula_scriptifronsi", "Andrena_afimbriat")
FuzzyMatch(tree = hym.tree, data = names, max.dist=3)
```

`horn.beetle.csv`	*Gnatocerus measurements*

Description

A csv file containing measurements of horn and body size for the beetle Gnatocerus cornutus.

`hym.tree`	*Phylogenetic tree*

Description

This is a phylogenetic tree with 5 species of hymenoptera.

`mcmc2`	*mcmc log file*

Description

an mcmc log file. The first column is the tree used during the iteration the remaining columns are the rate parameters of the Q matrix listed by column order.

`mite.trait`	*phenotype data for mites*

Description

dataframe of sexual system and chromosome number data for mites

---

Mode   *Calcualtes the mode of a numeric vector*

Description

R's base package function mode returns the type of object 'numeric', 'character' etc. This give the option of an easy to remember work around for that.

Usage

```
Mode(x)
```

Arguments

x   a numerical vector.

Details

Returns the most frequently occuring value in a vector. In the case of a tie it will return the mode which has the earliest initial occurence in the vector

Value

returns the most frequently occuring value in a series of numbers

Author(s)

Heath Blackmon

References

<http://coleoguy.github.io/>

Examples

```
Mode(c(1,2,3,4,5,6,2,5))
```

PPSDiscrete   *Create Simulated Datasets via PPS*

Description

This function performs posterior predictive simulations of discrete traits. The function is written to work with the output of bayesian programs that produce a collection of rate matrix parameter estimates based on either one or a collection of trees.

Usage

```
PPSDiscrete(trees, MCMC, states, N = 2)
```

Arguments

trees   an object of class "multiPhylo" or "phylo" containing the trees used in generting the rate estimates

MCMC   this will normally be a log file that is brought into R with read.csv the columns for a three state character should be: tree, qAA, qBA, qCA, qAB, qBB, qCB, qAC, qBC, qCC. If your analysis involves only a single tree then the tree column should be excluded.

states   a vector of root probabilities

N   the number of PPS datasets desired

Value

A matrix is returned with the rownames being the species names from the tree and each column containing a result of a single PPS.

Author(s)

Heath Blackmon

References

<http://coleoguy.github.io/>

Examples

```
data(trees)
data(mcmc2)
data(mcmc3)
# 1 tree 100 q-mats 3 states
PPSDiscrete(trees[[1]], MCMC=mcmc3[,2:10], states=c(.5,.2,.3), N=2)
# 10 trees 100 q-mats 3 states
PPSDiscrete(trees, MCMC=mcmc3, states=c(.5,.2,.3), N=10)
# 10 trees 100 q-mats 2 states
PPSDiscrete(trees, MCMC=mcmc2, states=c(.5,.5), N=10)
```

ReorderData	*Reorders trait data to match the order of tips in a tree*

Description

This function takes a vector, matrix, or dataframe and reorders the data to match the order of tips in a phylo object.

Usage

```
ReorderData(tree, data, taxa.names="row names")
```

Arguments

tree	a phylo object
data	a vector, matrix, dataframe set of taxa names as present in the tree and data must match. If data is a vector it should be a named vector. If the data is a matrix or dataframe the taxa names may be row names or present in a column.
taxa.names	If taxa names are present in a column the column number should be supplied. If taxa names are the row names the argument can be set to "row names" (default setting). If the data is being supplied in a vector this argument is not used.

Details

Returns data in the same format as supplied but reordered to match the order of tip labels.

Author(s)

Heath Blackmon

References

<http://coleoguy.github.io/>

ResSel	*Selection on Residuals*

Description

This function takes measurements of multiple traits and performs a linear regression and identifies those records with the largest and smallest residual. Originally it was written to perform a regression of horn size on body size allowing for high and low selection lines.

Usage

```
ResSel(data, traits, percent = 10, identifier = 1, model = "linear")
```

Arguments

data	this is a dataframe with subject identifiers and phenotypic trait values
traits	a numeric vector indicating the column containing the predictor and response variables in that order
percent	the percentage of highest and lowest residuals that should be identified
identifier	the column which contains the record numbers to identify individuals
model	currently this is not used

Value

This function returns a list

high line	the ID numbers for the individuals selected for the high line
low line	the ID numbers of the individuals selected for the low line

Author(s)

Heath Blackmon

References

<http://coleoguy.github.io/>

Examples

```
data <- read.csv(file = system.file("horn.beetle.csv", package = "evobiR"))
ResSel(data = data, traits = c(2,3), percent = 15, identifier = 1, model = "linear")
```

SampleTrees	*Select a random sample of trees*

Description

This function takes as its input a large collection of trees from a program like MrBayes or Beast and allows the user to select the number of randomly drawn trees they wish to retrieve

Usage

```
SampleTrees(trees, burnin, final.number, format, prefix)
```

Arguments

trees	a nexus format file containing trees that the user wants to sample from
burnin	the proportion of trees to remove as burnin
final.number	the number of trees desired
format	options are "new" or "nex" indicating to save the trees in newick format or nexus format
prefix	a text string to assing to the new treefile name

Value

an object of the class "multiPhylo" is returned

Author(s)

Heath Blackmon

References

<http://coleoguy.github.io/>

Examples

```
SampleTrees(trees = system.file("trees.nex", package = "evobiR"),
            burnin = .1, final.number = 20, format = 'new', prefix = 'sample')
```

SlidingWindow   *Sliding window analysis*

Description

Applies a function within a sliding window of a numeric vector. Both the step size and the window size can be set by the user.

Usage

```
SlidingWindow(FUN, data, window, step)
```

Arguments

FUN	a function to be applied within each window.
data	a numerical vector.
window	an integer setting the size of the window.
step	an integer setting the size of step between windows.

Details

Returns a vector of numeric values representing the applying the selected function within each window. The length will be unequal to the original data and will be determined primarily by the step size.

Author(s)

Heath Blackmon

References

<http://coleoguy.github.io/>

Examples

```
data <- c(1,2,1,2,10,2,1,2,1,2,3,4,5,6,2,5)
SlidingWindow("mean", data, 3, 1)
```

SuperMatrix   *creates a supermatrix from multiple gene alignments*

Description

combines all alignments in a folder into a single supermatrix

Usage

```
SuperMatrix(missing = "-", prefix = "concatenated", save = T)
```

Arguments

missing   the character to use when no data is available for a taxa

prefix   prefix for the resulting supermatrix

save   if True then supermatrix and partition file will be saved

Details

This function reads all fasta format alignments in the working directory and constructs a single supermatrix that includes all taxa present in any of the fasta files and inserts missing symbols for taxa that are missing sequences for some loci.

Value

A list with two elements is returned. The first element contains partition data that records the alignment positions of each input fasta file in the combined supermatrix. The second element is a dataframe version of the supermatrix. If the argument save is set to True then both of these files are also saved to the working directory.

Author(s)

Heath Blackmon

References

<http://coleoguy.github.io/>

Examples

```
## Not run:
SuperMatrix(missing = "N", prefix = "DATASET2", save = T)

## End(Not run)
```

---

trees	*10 Phylogenetic trees*

Description

This is a collection of 10 simulated phylogenetic trees with 200 tips each.

---

trees.mite	*10 Phylogenetic trees*

Description

These are trees from a previously published work on mite sexual system evolution.

---

trees.nex	*100 Phylogenetic trees*

Description

This is a collection of 100 simulated phylogenetic trees with 10 tips each.

ViewEvo	*Learning Resources*

Description

This uses the shiny app to produce interactive pages.

Usage

```
ViewEvo(simulation)
```

Arguments

simulation   Text string indicating the application to run. Currently options are "wf.model",
                "bd.model", "dist.model"

Details

The wf.model was implemented to illustrate to students the effects of genetic drift. In particular the high likelihood of losing a beneficial allele when population size is finite. The bd.model will plot 4 phylogenetic trees based on a birth death model with a single set of parameters. This application was developed to illustrate the high variability of a birth death process as a generating model for phylogenies and the inherint difficulty in detecting differential diversification rates. Finally the dist.model was developed to help illustrate the relationship between common statistical distributions often used as priors and the way that parameters effect the density distribution.

Value

This function returns an interactive webpage.

Author(s)

Heath Blackmon

References

<http://coleoguy.github.io/>

Wright-Fisher Simulator: <https://evobir.shinyapps.io/wf_model/>

Birth-death Simulator: <https://evobir.shinyapps.io/bd_model>

Statistical Distribution: <https://evobir.shinyapps.io/dist_model>

Examples

```
## Not run:
ViewEvo("wf.model")
ViewEvo("bd.model")
ViewEvo("dist.model")

## End(Not run)
```

WinCalcD   *Calculate Patterson's D-statistic in sliding windows*

Description

This functions calculate Patterson's D-statistic in windows.

Usage

```
WinCalcD(alignment = "alignment.fasta", win.size = 100, step.size=50,
boot = F, replicate = 1000)
```

Arguments

`alignment`	This is an alignment in fasta format
`win.size`	This is the size of the window used
`step.size`	This is the size of steps in the sliding window
`boot`	This indicates whether or not bootstrapping should be performed to estimate variance
`replicate`	Number of replicates to be used in estimating variance

Details

This function is just an extension of CalcD and calculates D statistic for windows.

Value

Returns a table with the number of each type of site, Z scores and p-values for each window in the genome

Author(s)

Heath Blackmon

References

http://coleoguy.github.io/

Durand, Eric Y., et al. Testing for ancient admixture between closely related populations. Molecular biology and evolution 28.8 (2011): 2239-2252.

Eaton, D. A. R., and R. H. Ree. 2013. Inferring phylogeny and introgression using RADseq data: An example from flowering plants (Pedicularis: Orobanchaceae). Syst. Biol. 62:689-706

Examples

```
WinCalcD(alignment = system.file("1.fasta", package = "evobiR"),
         win.size=100, step.size=50, boot = TRUE, replicate=10)
```

Index