Package: evobiR (via r-universe)

September 5, 2024

Type Package

Title Evolutionary Biology in R

Version 2.1

Date 2023-5-13

Author Michelle M. Jonika, Maximos Chin, Nathan Anderson, Richard H. Adams, Jeffery P. Demuth, and Heath Blackmon

Maintainer Heath Blackmon <coleoguy@gmail.com>

URL https://github.com/coleoguy/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.

License GPL (>=2)

Imports seqinr, ape, geiger, phytools, castor, methods, viridis, RColorBrewer, expm, igraph

Suggests knitr

VignetteBuilder knitr

Repository https://phylotastic.r-universe.dev

RemoteUrl https://github.com/coleoguy/evobir

RemoteRef HEAD

RemoteSha 81a898a1054ed356e43154f4f276825f466a3f66

Contents

`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:	2017-5-05
License:	GPL (>=2)

More information on **evobiR** is available at https://github.com/coleoguy/evobir

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/

AncCond *Calculate the mean of a continuous character at transitions in a binary charachter*

Description

This function uses ancestral state estimations for a discrete character based on stochastic mapping under an Mk2 model and ancestral state estimates for a continuous trait under a Brownian motion model to determine if transitions in the binary trait coincide with extreme values of a continuous trait.

Usage

```
AncCond(tree,
        data,
        mc = 1000,
        drop.state=NULL,
        mat=c(0,2,1,0),
        pi="equal",
        n.tails = 1,
        message = TRUE)
```

Arguments

tree	tree of class phylo
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. The binary trait should be coded as 1 and 2 if one is ancestral then it must be coded as 1.
mc	the number of iterations to use in building the null distribution.

drop.state	NULL or a numeric value of 1 or 2. If 1 or 2 are given then continuous data from species in the specified state will be dropped from the reconstruction of the continuous character.
mat	a vector describing the possible transition for the discrete trait. The default is equivelant to APE's "ARD" model c(0,2,1,0), other options are c(0,0,1,0) which would allow only for transition from state 1 to state 2 or c(0,1,1,0) which would allow for transition in either direction but at equal rates.
pi	The probabilities for the binary trait at the root of the tree. The values possible are "equal", "estimated", or a numeric vector of length 2 with probabilities for each state
n.tails	numeric value of 1 or 2 to indicate whether a 1 or 2 tailed p-value should be calculated
message	Logical value if TRUE then status messages will be printed to the console

Details

This function uses ancestral state estimates to determine if the transitions in the binary trait are associated with extreme values of the continuous trait. This test can incorporate the possibility that the derived state of a binary character may lead to correlated selection in the continuous trait. If this is desired then the drop.state argument should be used to specify the derived state of the binary character that should not be used in the ancestral state estimation of the continuous trait.

Value

Returns a list of length 4:

OriginatingVals	the mean value(s) for the continuous trait at the transition points of the binary character
NTrans	the number of transitions in the binary character
NullDist	the null distribution(s) produced by simulation
pval	pvalue

Author(s)

Nathan Anderson, Jeffery P. Demuth, Richard H. Adams, and Heath Blackmon

References

http://coleoguy.github.io/

Examples

```
## Not run:
data(mite.trait)
data(trees.mite)
AncCond(trees[[1]], mite.trait)

## 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", ambig="D",
block.size = 1000, replicate = 1000, align.format='fasta')

CalcPopD(alignment = "alignment.fasta", sig.test = "N", ambig="D",
block.size = 1000, replicate = 1000, align.format='fasta')
```

Arguments

alignment	This is an alignment in fasta format. Sequences should be in the order: P1, P2, P3, Outgroup. In the case of CalcPopD sequence from each populations should have identical names see file 3.fasta for an example
sig.test	This indicates whether or if to test for significance. Options are "B" bootstrap, "J" jackknife, or "N" none.
ambig	This flag indicate how to deal with sequence ambiguities. Options are "D" drop all ambigous loci, "R" resolve each biallelic ambiguity, or "I" ignore ambiguity and perform analysis without checking sequences. If the argument "R"" is chosen there is becomes a degree of stochasticity in the analysis and the user should run the analysis more than once. Also it would be wise to compare this result to setting the argument to "D".
block.size	The number of sites to be dropped in the jackknife approach
replicate	Number of replicates to be used in estimating variance
align.format	a character string specifying the format of the alignment file : mase, clustal, phylip, fasta or msf

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"), sig.test = "N")
```

`countTrees` *Calculate the number of times a set of topologies occur*

Description

This function counts the number of times that a set of topologies is present in a collection of trees.

Usage

```
countTrees(collection = NULL, ref = NULL, classes = T, verbose=T)
```

Arguments

`collection`	path to a collection of trees in a Newick format file
`ref`	path to a Newick format file with the topologies of interest
`classes`	if T then will return a vector with classification of each tree
`verbose`	returns intermediate progress messages if TRUE

Value

If classes is T returns a list with the first element being a numeric vector of the same length as the number of trees in the ref file. The elements of the returned vector correspond to the occurences of trees in the collection file that match the topologies supplied in the ref file. The second element is a vector the same length as the input tree collection and each tree is assigned a number based on the topology it matches.

Author(s)

Heath Blackmon

References

http://coleoguy.github.io/

`fix.simmap` *Fix a stochastic map with failed edges*

Description

This function facilitates automated resolution of failed edges in a modified stochastic map produced by `make.simmap2` through application of graph theory implemented in **igraph**.

Usage

```
fix.simmap(hists, tips, transition.matrix)
```

Arguments

`hists`	an object of class `simmap` or `multiSimmap` produced by `make.simmap2`
`tips`	two column dataframe with first column listing species name as shown in tree tips and second column listing model state for each species
`transition.matrix`	square matrix describing possible transitions between states

Value

A object of class `simmap` or `multiSimmap`, see `make.simmap`.

All edges which failed in the original stochastic maps should be resolved.

Author(s)

Maximos Chin, Matthew Marano, and Heath Blackmon

References

http://coleoguy.github.io/

See Also

`make.simmap2`

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)
```

getNe — *Calculate the variance effective population size*

Description

This function calculates the variance effective population size due to unequal sex ratio. Formula are available for both autosomal loci and X chromosome loci.

Usage

```
getNe(males, females, locus)
```

Arguments

males	number of males
females	number of females
locus	"A" or "X" to denote the population size of interest is for autosomal locus or X chromosome locus respectively.

Value

Returns a numeric vector of length 1 which contains the variance effective population size.

Author(s)

Heath Blackmon

References

http://coleoguy.github.io/

Examples

```
getNe(males=100, females=200)
```

GetTipRates — *Calculate the rate of evolution on the leaves of a phylogeny*

Description

This function calculates the rate of change from parent nodes to extant tips of a phylogeny.

Usage

```
GetTipRates(tree, Q, tip.states, hyper, p.mat)
```

Arguments

tree	an ultrametric tree of class phylo
Q	transition matrix containing estimated rates with column names
tip.states	An integer vector with a length equal to the number of species on the phylogeny. It should have values of 1 to N with N being the number of total states. Elements of the vector should match the tip names for the phylogeny.
hyper	logical vector of length one. TRUE indicates the model includes a binary hyperstate. Default is FALSE and indicates no binary hyperstate
p.mat	a probability matrix where each column represent a discrete state and each row is a species. Values in the matrix describe the probability of being in any of these states

Value

A named numeric vector with the rate for each tip in the phylogeny.

Author(s)

Michelle M. Jonika and Heath Blackmon

References

http://coleoguy.github.io/

---

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.

make.simmap2 *Modified stochastic mapping which is resistant to model failure*

Description

This function is an extension of the make.simmap function of **phytools** which allows users to monitor rejections at specific edges during the simulation proces and set an upper limit on the number of rejections permitted per edge.

Usage

```
make.simmap2(tree, x, model="SYM", nsim=1 ,
             monitor = FALSE, rejmax = NULL, rejint = 1000000,
             ...)
```

Arguments

tree	see make.simmap
x	see make.simmap
model	see make.simmap
nsim	see make.simmap
monitor	boolean describing whether or not to print number of rejections per edge to console. Defaults to FALSE
rejmax	int greater than one giving the maximum number of rejections permitted before an edge is skipped. Defaults to NULL (no upper limit on rejections)
rejint	int giving the interval after which rejection number is to be printed to console
...	optional arguments. see make.simmap

Value

A object of class simmap or multiSimmap, see make.simmap.

In addition to the states present in the model, an additional state fail in the maps and mapped.edge elements is assigned to edges which are skipped due to exceeding the rejection limit.

Author(s)

Matthew Marano, Maximos Chin, and Heath Blackmon

References

http://coleoguy.github.io/

See Also

make.simmap

mite.trait *phenotype data for mites*

Description

A dataframe of sexual system and chromosome number data for mites. The first column has species names, the second column has diploid numbers, and the third column contains 0 and 1 to indicate diplodiploidy or haplodiploidy sex determination.

Pfsa *Calculate the proportion of different classes of fusions as a fraction of all fusions*

Description

Thes three functions, Pfsa, Pfaa, and Pfss use sex chromosome systems and diploid autosome number to calculate the proportion of all fusions that are expected to be sex chromosome autosome fusions (Pfsa), autosome autosome fusions (Pfaa), and sex chromosome sex chromosome fusions (Pfss).

Usage

```
Pfsa(Da, scs, mud)
Pfaa(Da, scs, mud)
Pfss(Da, scs, mud)
```

Arguments

Da	the diploid autosome count
scs	a text string describing the sex chromosome system for instance: "XO", "XXO", "XY", "ZO"", or "ZWW"
mud	the proportion of fusions that originate in females.

Details

This approach assumes that all chromosomes have equal probability of being involved in a fusion and that X and Y (or Z and W) chromosomes are not able to fuse. It will provide accurate results for any sex chromosome system that has any number of one sex chromosomes and either zero or one of the other. For instance it will work for XO, XY, XXY, XYYYY, but not for XXYY. It is applicable to both male and female heterogameic systems.

Value

Returns a numeric vector of length 1 which contains the proportion of fusions expected to be of the specified type.

Author(s)

Nathan Anderson and Heath Blackmon

References

http://coleoguy.github.io/

Examples

```
Pfsa(Da = 26, scs = "XY", mud=0.4)
```

`plot` *Phylogenetic visualization of heterogenity in discrete character evolution*

Description

This function provides two methods for visualizing a `phyloscaled` tree produced by `scaleTreeRates`.

Usage

```
## S3 method for class 'phyloscaled'
plot(tree, method = "multiply", palette = "RdYlGn",
     edge.width = 1, cex = 1, show.tip.label = T)
```

Arguments

`tree`	a tree of class `phylo` and `phyloscaled`
`method`	a `string` describing the method to be used for visualization. Can either be "multiply" or "color", see **Details**
`palette`	a `string` giving the palette to be used for edge coloring, only taken into account if `method = "color"`. See **Details**
`edge.width`	numeric value that determines branch width
`cex`	numeric value for size
`show.tip.label`	logical indicating whether to print tip labels

Details

The two plotting methods currently supported are `"multiply"` and `"color"`.

`multiply` takes the length of each edge on the phylogeny and multiplies it by the scalar associated with the edge before plotting the scaled tree.

`color` assigns a color from a diverging palette to each edge depending on it's associated scalar. Currently supported palettes are any of the **RColorBrewer** palettes or the standard viridis palette (specified by string `"viridis"`). Because RColorBrewer sets the maximum number of distinct colors for divergent palettes to be 11, any phylogeny which has greater than 11 unique scalar bins represented within it's edges must use a viridis palette. The ultrametric phylogeny is then plotted with each edge colored accordingly

Value

A plotted phylogeny with edges either multiplied or colored by their associated scalars

Author(s)

Maximos Chin and Heath Blackmon

References

http://coleoguy.github.io/

See Also

scaleTreeRates

`ReOrderAlignment` *Re-order sequences based on starting position*

Description

This function re-orders aligned sequences with sequences based on the site of the alignment that they first have data. It also allows user to select reference sequences that will stay at the top of the alignmment regardless of starting position.

Usage

```
ReOrderAlignment(file, newfile, ref = "")
```

Arguments

`file`	the name of the fasta file which has sequences need to be re-ordered
`newfile`	name of the output file
`ref`	Either an integer or a characrter vector. Elements of the vector should match the sequences that should be kept at the top of the alignment.

Value

A fasta file

Author(s)

Sean Chien and 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')
```

scaleTreeRates *Phylogeneitc analysis of heterogeneity in discrete character evolution*

Description

This function performs the phylogenetic methods for analysis of heterogenity in rates of discrete character evolution described in Jonika et al. (2023).

Usage

```
scaleTreeRates(tree, tip.states, model, fixedQ = NULL,
               max.ratio = 2, nbins = 10, max.transition = 1,
               var.start = FALSE, pi = "fitzjohn")
```

Arguments

`tree`	a tree of class `phylo`
`tip.states`	a named vector of tip states for some discrete character which is associated with the phylogeny. Order can differ from order of tips on phylogeny
`model`	the model which should be used to perform likelihod calculations. This can either be a `string` which can be passed to the `fitMk` function of **phytools** or a symmetrical transition `matrix` which has transitions between states categorized into some number of distinct classes
`fixedQ`	optional argument to be used when Q-matrix with pre-estimated rates is available. Deafults to `NULL`
`max.ratio`	`num` or `int` greater than one descirbing the maximum ratio of scalar bins to one. Defaults to 2, i.e. scalar bins range between 0.5 and 2
`nbins`	`int` giving the number of scalar bins above and below 1. Defaults to 10, i.e. 10 bins below 1 and 10 bins above 1 for a total of 21 bins inclusive of 1
`max.transition`	`int` giving the maximum number of bins which the scalar associated with an edge can differ from the scalar associated with it's parent edge. Defaults to 1
`var.start`	`logical` whether or not to increment scalar values at the root of the tree. If `TRUE`, the analysis will be iterated across all pssible root scalar values and the best tree (highest likelihood) returned. If `FALSE` (default), root scalar is set to one and only a single iteration is performed
`pi`	`string` giving method to be used for estimating prior. Takes any option which can be passed to phytool's `fitMk`, defaults to `"fitzjohn"`

Value

A phylogeny of class `phylo` and `phyloscaled`. Phylogeny has all elements normally included in an object of class `phylo`, with an additional element:

`scalar`	a `numeric` vector of scalars equal in length to the number of edges in phylogeny. Ordering of scalars is identical to the ordering of edges

Author(s)

Maximos Chin and Heath Blackmon

References

http://coleoguy.github.io/

`SlidingWindow` *Sliding window analysis*

Description

Applies a function within a sliding window of a numeric vector or matrix. Both the step size and the window size can be set by the user. For the matrix impelentation the step size and window size is constrained to be the same in both the X and Y dimensions.

Usage

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

Arguments

`FUN`	a function to be applied within each window.
`data`	a numerical vector or matrix
`window`	an integer setting the size of the window
`step`	an integer setting the size of step between windows
`strict`	TRUE or FALSE indicating whether validation testing should be performed

Details

If the input data is a vector then returns a vector of numeric values representing the application of the selected function within each window. If the input data is a matrix then returns a matrix of numeric values representing the application fo the selected function within each window.

Author(s)

Heath Blackmon

References

http://coleoguy.github.io/

Examples

```
# vector example
x1 <- rnorm(100, sd=3)
z1 <- SlidingWindow(FUN="mean", data=x1, window=10, step=5, strict=TRUE)

# matrix example
x2 <- matrix(rnorm(10000),100,100)
z2 <- SlidingWindow(FUN="mean", data=x2, window=10, step=5, strict=TRUE)
```

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,
input = "", format.in = "f", format.out = "f", concatenate = 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
input	a regular expression determining which files will be read in for example "*.fasta" will read in all files which end in ".fasta". Default is blank and will result in all files in the working directory being read in.
format.in	A character string specifying the format of the alignments to be read in. The argument is passed to read.dna in APE: "interleaved", "sequential", "clustal", or "fasta", or abbreviations of these are available.
format.out	A character string specifying the format for the supermatrix to be saved to. The argument is passed to write.dna in APE: "interleaved", "sequential", "clustal", or "fasta", or abbreviations of these are available.
concatenate	logical value when TRUE sequences are concatenated into a single fasta file. When set to FALSE sequences are saved as individual fasta files but are expanded to include all taxa in combined dataset.

Details

This function reads all alignments in the working directory and constructs a single supermatrix that includes all taxa present in any of the 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 alignment 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, format = "f")

## End(Not run)
```

---

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.

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