Package 'evobiR'

January 27, 2015

Type Package

Title evobiR: evolutionary biology in R.

Version 1.0

Date 2013-10-08

Author Heath Blackmon

Maintainer Heath Blackmon <coleoguy@gmail.com>

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

Depends seqinr, ape, stringr, geiger, taxize

Description evobiR is a collection of tools for use in evolutionary biology

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2013-10-21 17:35:27

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:	1.0
Date:	2013-10-08
License:	GPL (>=2)

More information on **evobiR** is available at http://www.uta.edu/karyodb/evobiR/

A full list of functions in the **evobiR** package is accesible by typing library(help = evobiR)

Author(s)

Heath Blackmon

Maintainer: Heath Blackmon <coleoguy@gmail.com>

1.fasta *simulated SNP data*

Description

This file just contains some simulated SNP data

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

2.fasta *simulated SNP data*

Description

This file just contains some simulated SNP data

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

3.fasta *simulated SNP data*

Description

This file just contains some simulated SNP data

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

`AnalyzeAssembly` *Analyze Genome Assembly*

Description

This function reports basic statistics for a genome assembly.

Usage

```
AnalyzeAssembly(genome, max_N = 25, plot = F)
```

Arguments

`genome`	a list of vectors with each element being a single string of the class "SeqFastadna".
`max_N`	Maximum number of consecutive N symbols. Scaffolds will be broken into contigs when this number is exceeded.
`plot`	When True an accumulation plot will be returned as well as the statistics

Details

If a standard FASTA file is read in with the function read.fasta from the package seqinr the argument as.string should set to TRUE. The genome should also be all lower case which is the default setting for read.fasta.

Value

A dataframe with the following rows:

Number of Scaffolds
Assembly Size Based on Scaffolds
Number of Scaffolds over 1MB
N50 Scaffold Size
Number of Contigs
Assembly Size Based on Contigs
N50 Contig Size
Minimum Contig Size
Percent GC

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

Examples

```
## just a small simulated genome
data(genome)
## calculate summary statistics for the genome
AnalyzeAssembly(genome = genome, max_N = 25, plot = TRUE)
```

`assembly.fasta` *a sample of a genome*

Description

This is a small portion of Tribolium genome

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

`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", boot = F, replicate = 1000)

CalcPartD(alignment = "alignment.fasta", boot = F, replicate = 1000, alpha = 0.05)

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

Arguments

`alignment`	This is an alignment in fasta format
`boot`	This indicates whether or not bootstrapping should be performed to estimate variance
`replicate`	Number of replicates to be used in estimating variance
`alpha`	The desired alpha level, used in calculating significance of tests in CalcPartD after Bonferoni correction

Details

The functions CalcD and CalcPopD are implementations of the algorithm described in Durand et al. 2011. The function CalcPartD is an implementation of the extension that was reported by Eaton and Ree 2013 which extended the original ABBA-BABA test to include additional taxa providing the ability to identify the source population of introgression events when more than one extant lineage may have hybridized with the ingroup.

Value

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

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

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"), boot = TRUE, replicate=10)

CalcPartD(alignment = system.file("2.fasta", package = "evobiR"), boot = TRUE, 10)

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

CoalSim *Coalescence Simulations*

Description

This function simulates the coalescence of linages through time. At the begining of the simulation a number of unique lineages specified by the census size as we go back in time these coalesce until we have only a single lineage remaining.

Usage

```
CoalSim(census, lw, ln.col)
```

Arguments

census	the size of population to use in the simulation.
lw	line width factor passed to the plotting function.
ln.col	specifies the color to be used in plotting the coalesence history

Value

Returns a plot showing the result of the coalescence simulation.

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

Examples

```
CoalSim(census = 15, lw = 2, ln.col = 'blue')
```

---

data.taxonomy *taxonomic table*

Description

This file contains taxonomic information

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

DriftSim *Simulate Allele Frequencies*

Description

These functions simulate allele frequencies through time

Usage

```
SelSim(census, initial.freq, selection.coef, iter, generations)

DriftSim(census, initial.freq, iter, generations)
```

Arguments

census	The population size to be used for the simulation.
initial.freq	The proportion of individuals that should have the selected mutation at the begining of the simulation.
selection.coef	The selection coefficient. 1 is a neutral mutation while values greater or less than 1 will be beneficial or deleterious respectively.
iter	The number of times that the simulation should be repeated
generations	How many generations each simulation should last.

Value

These functions were created to help in teaching population genetics. Each of these functions plot the frequency of a new mutation at each generation. Each iteration of the simulation is plotted in a new color and the functions also reports the number of iterations that resulted in fixation, loss or intermediate frequencies for the mutation. The function DriftSim is just a special case of SelSim where the selection coefficient is equal to 1.

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

Examples

```
set.seed(5)
SelSim(census = 200, initial.freq = .5, selection.coef = 1.01, iter = 10, generations = 200)

set.seed(5)
DriftSim(census = 200, initial.freq = .5, iter = 25, generations = 200)
```

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 and Matt Pennell

References

http://www.uta.edu/karyodb/evobiR/

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)
```

genome *A small portion of the genome of Tribolium confusum*

Description

This is just little piece of the Tribolium confusum genome

Format

a list of vectors with each element being a single string of the class "SeqFastadna".

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

GenomeSym *Genome Simulation*

Description

THIS FUNCTION IS STILL IN THE DEVELOPMENT PHASE AND RESULTS FROM IT SHOULD NOT BE TRUSTED WITHOUT CONTACTING ME FIRST!!!

This function is designed to simulate the evolution of a diploid organism with chromosomal sex determination. It allows for the evolution of sexual antagonism, recombination supression, anneuploidy, haplosufficiency.

Usage

```
GenomeSym(model = "fragileY", generations = 100, census = 50,  gametes = 4,
          loci.a  = 100, loci.s = 50, prob.point.mut = .0010,
          DFE = (c(rnorm(1000, mean = 0.9, sd = 0.06), rgamma(1000, rate = 2,
         shape = 0.4))), sex.bias.mut = .85, haplosuff.mut = 1, crossovers = 2,
        recom.mu = 10, anneuploi.mut = c(0, .0002), distance.mut = c(0, .000002),
          achiasmat.mut = c(0, .00002), sex.ant = c(.5, .3, .2),
          fragile.fact = 20, reporting = "None")
```

Arguments

model	the type of simulation to be run
generations	integer indicating the number of generations that the simulation should be run
census	integer indicating total census size
gametes	integer indicating number of gametes drawn from each individual
loci.a	integer indicating loci on autsomes
loci.s	integer indicating loci on sex chromosomes
prob.point.mut	probability that an new allele will arrise
DFE	a vector of fitness effect for mutations
sex.bias.mut	propotion of mutations to arrise in males
haplosuff.mut	a value of less than 1 from which mutations in haplosufficiency will be drawn from a uniform distribution between 0 and the specified value.
crossovers	the average number of crossovers per chromosome per meiosis in autosomes
recom.mu	maximum number of loci N that can quit recombining in a single mutational step
anneuploi.mut	probability of a Y aneuploidy mutation drawn from a uniform dist between X and Y
distance.mut	prob of evolving dist pairing uniform dist from X to Y
achiasmat.mut	prob of evolving achiasmatic meiosis in males unirom from X to Y
sex.ant	numerical vector must sum to 1 indicating the probability of unbiased, male biased, female biased mutations
fragile.fact	this is the factor by which we increase aneuploidy based on par reduction
reporting	a character string indicating the desired type of insimulation reporting: 'PAR', 'XY.fitness', 'all.loci'.

Details

This function simulates a genome as a series of numerical vectors each chromosome in a genome has vectors to represent those genes fitness when in a female or when in a male additionally if they are sex chromosomes they have vectors describing the recombination characteristics and haplosufficiency data. All of these genome characteristics are allowed to evolve during the simulation through a variety of mutational mechanisms. The genomes perform in silico gametogenesis with recombination. The probability of gametes being used to create the next generation is in proportion to their parents relative fitness.

default values for the function are as follows:

model = "fragileY"

generations = 100

census = 50

gametes = 4

loci.a = 100

loci.s = 50

prob.point.mut = .3

DFE = (c(rnorm(1000,mean=.9,sd=.06),rgamma(1000, rate=2, shape=.4)))

sex.bias.mut = .85

haplosuff.mut = 1

crossovers = 2

recom.mu = 10

anneuploi.mut = c(0, .0002)

distance.mut = c(0, .000002)

achiasmat.mut = c(0, .00002)

sex.ant = c(.5, .3, .2)

fragile.fact = 20

reporting ="None"

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

Examples

```
## just a small simulated genome
Genome <- GenomeSym(census = 20, generations = 10, reporting = 'all.loci')
```

`GetTaxonomy` *Create a taxonomy table*

Description

This function is an extension of the tax_name function from the package taxize. The input is a tree and the function will return a table with a description of the taxonomy of all species on the tree

Usage

```
GetTaxonomy(tree, database = "ncbi")
```

Arguments

`tree`	an object of class "phylo"
`database`	character string "ncbi or "itis" indicating the database that should be used

Value

this function saves a csv file named: tree.taxonomy.csv

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

Examples

```
# data(hym.tree)
# GetTaxonomy(tree = hym.tree, database = 'ncbi')
```

HetLevels *Calculates heterozygosity based on ambigous IUPAC nucleotide codes*

Description

Some assemblers like AllPaths will use IUPAC codes to indicate polymorphic sites in an assembly. This function uses these to estimate the level of heterozygosity along a sequence.

Usage

```
HetLevels(fasta)
```

Arguments

fasta	A fasta formatted file that may contain 1 or more sequences.

Details

Ambiguity codes of "r", "y", "s", "w", "k", "m","b", "d", "h", "v" are counted as polymorphic sites with an "n" are not counted since they are often indicative of scaffolding not polymorphism.

Value

A matrix is returned listing the length of each sequence the number of polymorphic sites and the number of "n" sites.

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

Examples

```
HetLevels(fasta = system.file("assembly.fasta", package = "evobiR"))
```

`HighLevelTree` *Prune a phylogeny to a higher taxonomic levels.*

Description

This function takes a species level tree and prunes it so only a single representative of each clade, e.g. genus or family, is present.

Usage

```
HighLevelTree(taxa.table, tree, cur.tips, new.tips)
```

Arguments

`taxa.table`	a character matrix with columns containing the current tips of the tree and additional columns containing their assigned genera, family, tribe, etc.
`tree`	the current species level tree
`cur.tips`	a number indicating which column contains the current tip names
`new.tips`	a number indicating which column contains the names for the taxonomic level that you would like to prune the tree to.

Value

returns and object of the class "phylo"

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

See Also

GetTaxonomy

Examples

```
data(species.tree)
data(taxa.table)

family.tree <- HighLevelTree(taxa.table, species.tree, 1, 2)
genus.tree <- HighLevelTree(taxa.table, species.tree, 1, 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

A 5 column by 100 row double matrix from a 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.

`mcmc3` *mcmc log file*

Description

A 10 column by 100 row double matrix from a 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.data` *cytogenetic mite data*

Description

This file contains cytogenetic mite data available at the tree of sex database

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

mite.tree *phylogenetic tree of mites*

Description

This file contains a phylogenetic tree of mites generated from phylota data by Nate Hardy

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

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 genertng the rate estimates
MCMC	MCMC states bring 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://www.uta.edu/karyodb/evobiR/

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)
```

ReplTable *Maximize Retained Tips*

Description

These functions attempt to retain as many datapoints as possible during a comparative analysis.

Usage

```
ReplTable(tree, data.taxonomy, tree.taxonomy, levels=c("Genus", "Family"), verbose=TRUE)

MaxTips(tree, data, repl.table)
```

Arguments

tree	a phylogenetic tree of the class "phylo".
data.taxonomy	a datframe containing taxonomic information for all species in the dataset see function GetTaxonomy
tree.taxonomy	a datframe containing taxonomic information for all species in the dataset see function GetTaxonomy
data	a dataframe containg the data you would like to use in your comparative analysis
repl.table	the output of ReplTable and one of the inputs of MaxTips
levels	currently not implemented
verbose	turns on reports to the screen.

Details

The goal of these two functions is to maximize the number of tips that are retained in a comparative phylogenetic analysis. This is done by searching for higher taxanomic level matches. This is accomplished by keeping one example of any available genera not matched at the species level and one available member of each family not matched at the genus or species level.

The taxonomy tables should have columns name of "Family", "Genus", and "Binomial". The dataframes (tree.taxonomy, data.taxonomy, and data) should all have row names correspoinding to the binomial.

Value

ReplTable produces a dataframe that is used as one of the inputs for MaxTips. It lists the tips on the tree that will be replaced and what taxonomic level matches have been found at. MaxTips returns a pruned dataset and tree.

Author(s)

Heath Blackmon and Matt Pennell

References

http://www.uta.edu/karyodb/evobiR/

See Also

GetTaxonomy

Examples

```
## Not run:
data('mite.data')
data('mite.tree')
data('data.taxonomy')
data('tree.taxonomy')

replacement.table <- ReplTable(tree = mite.tree, data.taxonomy = data.taxonomy,
                               tree.taxonomy = tree.taxonomy, levels=c("Genus", "Family"),
                                 verbose=TRUE)
new.data <- MaxTips(mite.tree, mite.data, replacement.table)
## End(Not run)
```

`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://www.uta.edu/karyodb/evobiR/

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://www.uta.edu/karyodb/evobiR/

Examples

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

species.tree *Species Phylogenetic tree*

Description

This is a simulated phylogenetic trees with 200 tips.

taxa.table *A table with taxonomic information*

Description

This character matrix contains species in one column followed by the genera and the family membership of each species

tree.taxonomy *taxonomic table*

Description

This file contains taxonomic information

Author(s)

Heath Blackmon

References

http://www.uta.edu/karyodb/evobiR/

trees *10 Phylogenetic trees*

Description

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

trees.nex *100 Phylogenetic trees*

Description

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

Index