Package 'phylowise'

August 21, 2026

Title Phylogenetic Pairwise Contrasts

Version 0.0.1

Maintainer Jordan Douglas <jordan.douglas@auckland.ac.nz>

Author Jordan Douglas [aut, cre],
Lindell Bromham [aut]

Description A phylogenetic comparative method for finding associations between biological traits and molecular evolutionary rates. The method samples pairs from a phylogeny such that each pair has non-overlapping edge paths, and can therefore be treated as statistically independent observations. Linear regression is performed on the pair contrasts. This approach is similar to phylogenetically independent contrasts (PIC) but without reconstructing the traits at internal nodes, and is better suited for finding trait-rate associations than phylogenetic generalised least squares (PGLS). Refer to Douglas and Bromham (2026) <doi:10.64898/2026.08.13.744736> for further details.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 8.0.0

URL https://github.com/jordandouglas/phylowise

Depends ape, BMA, phylotate

Imports Rcpp

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2026-08-21 13:40:02 UTC

Contents

`getDistanceMatrix` *Build a distance matrix from a tree (time tree or substitution tree)*

Description

Build a distance matrix from a tree (time tree or substitution tree)

Usage

```
getDistanceMatrix(tree)
```

Arguments

`tree`	a tree (phylo object)

Value

A distance matrix, where dij is the mean distance from leaf i and j to their ancestor

Examples

```
# Sample a coalescent tree with 10 taxa and then build a distance matrix
tree <- ape::rcoal(10)
dmat <- getDistanceMatrix(tree)
```

`plotPairs` *Plot a set of sampled pairs onto a tree*

Description

Plot a set of sampled pairs onto a tree

Usage

```
plotPairs(
  tree,
  pairs.df,
  edge.col = "red",
  show.tip.label = FALSE,
  edge.width = 1,
  edge.width.pairs = 3,
  label.cex = 1
)
```

Arguments

tree	the binary rooted tree used to get samples
pairs.df	a data frame of pairs, obtained using phylowise::sampleTaxonPairs
edge.col	the pairs will be highlighted in this colour
show.tip.label	display tip labels on the tree?
edge.width	edge line width of all branches on the tree
edge.width.pairs	edge line width of paired branches
label.cex	tree tip label font size, if show.tip.label=TRUE

Value

No return value, function is called to make a plot

Examples

```
# Sample a birth-death tree with 100 taxa
time.tree <- ape::rphylo(birth=10, death=5, n=100)
ntips = length(time.tree$tip.label)

# Simulate traits down the tree under Brownian motion
traits1 <- simulateTrait(time.tree)
traits1.leaf <- traits1[1:ntips]

# Simulate substitutions that have a positive association with traits
sim.result <- simulateSubstitutions(time.tree=time.tree,
         beta=1,
         theta=10,
         traits=traits1,
         method="TD",
         number.of.subst=10000)
subst.tree <- sim.result$subst.tree.est

# Sample taxon pairs within a window of (0.01, 0.2) time units
pairs.df <- sampleTaxonPairs(subst.tree=subst.tree,
       covariate=traits1.leaf,
       window.tree=time.tree,
       dist.min=0.01, dist.max=0.2)

# Plot the pairs onto the substitution tree
plotPairs(subst.tree, pairs.df, show.tip.label=TRUE)

# Plot the pairs onto the time tree
plotPairs(time.tree, pairs.df, show.tip.label=TRUE, edge.col="#008cba")
```

PPC.test *Phylogenetic pairwise contrast test on a series of taxon pairs, using ordinary least squares and Bayesian model averaging (BMA). This method does not do multivariate regression, but it returns the data frame that can be used for it.*

Description

Phylogenetic pairwise contrast test on a series of taxon pairs, using ordinary least squares and Bayesian model averaging (BMA). This method does not do multivariate regression, but it returns the data frame that can be used for it.

Usage

```
PPC.test(
  pairs.df,
  standardise = 2,
  logY = TRUE,
  extreme.value.threshold = 0,
  prior.weight = 0.5,
  OR = 1000,
  epsilon = 1e-06
)
```

Arguments

pairs.df	data frame of taxon pairs
standardise	should we standardise the evolutionary distance? 0 for no, 1 to standardise by sqrt(distance), 2 to standardise by sqrt(mrca age), 3 to standarise by mrca age. default and recommended setting: 2
logY	should we take the logarithm of the evolutionary distance? default and recommended setting: true
extreme.value.threshold	remove any observations more than this many standard deviations away from the mean response variable; set to zero for no extreme value removal
prior.weight	trait inclusion prior probbaility for Bayesian model averaging (see BMA::bic.glm)
OR	Occam's window for Bayesian model averaging (see BMA::bic.glm)
epsilon	precision of BMA probabilities for Bayes factor calculation: p < epsilon and p > 1-epsilon will be respectively set to epsilon and 1-epsilon to avoid NaN calculations

Value

A vector of p-values and Pearson correlations (one element per trait), a data frame of standardised and logged datapoints for doing regression, and posterior probabilities / Bayes factors from BMA analyses

Examples

```
# Sample a birth-death tree with 100 taxa
time.tree <- ape::rphylo(birth=10, death=5, n=100)
ntips = length(time.tree$tip.label)

# Simulate traits down the tree under Brownian motion
traits1 <- simulateTrait(time.tree)
traits1.leaf <- traits1[1:ntips]

# Simulate substitutions that have a positive association with traits
sim.result <- simulateSubstitutions(time.tree=time.tree,
         beta=2,
         theta=10,
         traits=traits1,
         method="TD",
         number.of.subst=10000)
subst.tree <- sim.result$subst.tree.est

# Sample taxon pairs within a window of (0.01, 0.2) time units
pairs.df <- sampleTaxonPairs(subst.tree=subst.tree,
       covariate=traits1.leaf,
       window.tree=time.tree,
       dist.min=0.01, dist.max=0.2)

# Perform linear regression on the pairs
ppc <- PPC.test(pairs.df)
p.value <- ppc$p.var
pearson <- ppc$rho.var
inclusion.prob <- ppc$bma.probs
bayes.factor <- ppc$bma.bf

# Plot the pairs. Should see a positive trend under these parameters
data.df <- ppc$data.df
plot(data.df$trait, data.df$distance.response, xlab="Trait contrast", ylab="Distance contrast")
```

`readBeastTrees` *Read a single summary tree or a posterior distribution of annotated nexus trees generated by the BEAST software suites, and apply burn-in. Built on top of the phylotate::read_annotated function.*

Description

Read a single summary tree or a posterior distribution of annotated nexus trees generated by the BEAST software suites, and apply burn-in. Built on top of the phylotate::read_annotated function.

Usage

```
readBeastTrees(nexus.file, burnin = 0.1)
```

Arguments

nexus.file	a nexus file
burnin	proportion of burn-in to apply to the file (default: 0.1)

Value

A list of trees

Examples

```
# Read in the small mammal.trees file generated by BEAST, available in the phylowise package
treefile <- system.file("extdata", "mammals.trees", package = "phylowise")
trees <- readBeastTrees(treefile, burnin=0)
tree1 <- trees[[1]]
plot(tree1)
print(tree1$node.comment)
```

`sampleTaxonPairs` *Sample a set of taxon pairs from the tree. These taxa will have non-overlapping edges between their paths and can this be treated as statistically independent. The algorithm iteratively searches for two random taxa that satisfy sampling requirements until there are no more valid pairs. Each pair must descend from an MRCA with dist.min <= tMRCA <= dist.max. c++ is used to speed up the runtime of this code.*

Description

Sample a set of taxon pairs from the tree. These taxa will have non-overlapping edges between their paths and can this be treated as statistically independent. The algorithm iteratively searches for two random taxa that satisfy sampling requirements until there are no more valid pairs. Each pair must descend from an MRCA with dist.min <= tMRCA <= dist.max. c++ is used to speed up the runtime of this code.

Usage

```
sampleTaxonPairs(
  subst.tree,
  covariate,
  dist.min,
  dist.max = Inf,
  response = NULL,
  nested = TRUE,
  maximise = TRUE,
  youngest = FALSE,
  window.tree = subst.tree,
  distance.matrix = NULL,
  verbose = FALSE
)
```

Arguments

subst.tree	a binary rooted tree, with branch lengths in units of change (phylo object)
covariate	a data frame of traits at the tips of the tree (rows are taxa, columns are traits); rows should be in the same order as tips in the tree
dist.min	minimum distance that two tips must be apart from their ancestor (mean of both distances), on window.tree
dist.max	maximum distance that two tips must be apart from their ancestor (mean of both distances), on window.tree
response	provide a vector as a response trait instead of genetic distances in the tree (optional)
nested	can taxon pairs be nested with each other?
maximise	should we maximise the trait difference?
youngest	take the youngest pair at each step (and therefore increase the number of pairs)?
window.tree	tree that is the basis for building the distance matrix, if it is not provided; should have same taxa as 'tree'
distance.matrix	provide an n x n distance matrix rather than recalculate from scratch
verbose	print some statements along the way

Value

A data frame, where each row is a pair of taxa.

Examples

```
# Sample a birth-death tree with 100 taxa
time.tree <- ape::rphylo(birth=10, death=5, n=100)
ntips = length(time.tree$tip.label)

# Simulate traits down the tree under Brownian motion
traits1 <- simulateTrait(time.tree)
traits1.leaf <- traits1[1:ntips]

# Simulate substitutions that have a positive association with traits
sim.result <- simulateSubstitutions(time.tree=time.tree,
         beta=1,
         theta=10,
         traits=traits1,
         method="TD",
         number.of.subst=10000)
subst.tree <- sim.result$subst.tree.est

# Sample taxon pairs within a window of (0.01, 0.2) time units
pairs.df <- sampleTaxonPairs(subst.tree=subst.tree,
       covariate=traits1.leaf,
       window.tree=time.tree,
       dist.min=0.01, dist.max=0.2)
```

```
# If iterating this, we can precompute the distance matrix to save time
dmat <- getDistanceMatrix(time.tree)
for (i in 1:100){
    pairs.df <- sampleTaxonPairs(subst.tree=subst.tree,
        covariate=traits1.leaf,
        distance.matrix=dmat,
        dist.min=0.01,
        dist.max=0.2)
}

# We can also do this with traits-vs-traits rather than rates-vs-traits
traits2 <- simulateTrait(time.tree) # Unassociated with traits1
traits2.leaf <- traits2[1:ntips]
pairs.df <- sampleTaxonPairs(subst.tree=subst.tree,
       response=traits1.leaf,
       covariate=traits2.leaf,
       window.tree=time.tree,
       dist.min=0.01,
       dist.max=0.2)
```

`simulateSubstitutions` *Simulate a substitution count down each branch of a tree. First, the branch rates are sampled from a correlated, uncorrelated or OU process that can be dependent on traits. Then the number of substitutions along each branch is sampled from a Poisson distribution.*

Description

Simulate a substitution count down each branch of a tree. First, the branch rates are sampled from a correlated, uncorrelated or OU process that can be dependent on traits. Then the number of substitutions along each branch is sampled from a Poisson distribution.

Usage

```
simulateSubstitutions(
  time.tree,
  nu = 0.5,
  sigma = 0.5,
  beta = 0,
  theta = 1,
  traits = NULL,
  number.of.subst = 1000,
  method = c("AC", "UCLN", "TD")
)
```

Arguments

time.tree	a binary rooted tree (phylo object)
nu	variance scale of autocorrelated clock or TD
sigma	standard deviation of UCLN
beta	effect size of traits on rates (if traits is not NULL)
theta	theta term for OU process in TD
traits	one trait per node; set to NULL if rates are conditionally independent of traits
number.of.subst	expected number of substitutions per unit of time
method	clock model may be uncorrelated lognormal (UCLN), autocorrelated lognormal (AC), or trait dependent (TD)

Value

Two trees, with branch lengths set to either subst. rates or counts, and two vectors, one of true branch rates and one of estimated branch rates (i.e., count divided by time, which can evaluate to zero)

Examples

```
# Sample a birth-death tree with 50 taxa
time.tree <- ape::rphylo(birth=10, death=5, n=50)

# Simulate traits down the tree under Brownian motion
traits <- simulateTrait(time.tree)

# Simulate substitutions that have a positive association with traits
sim.result <- simulateSubstitutions(time.tree=time.tree,
        beta=1,
        theta=10,
        sigma=0.5,
        traits=traits,
        method="TD",
        number.of.subst=10000)
subst.tree <- sim.result$subst.tree.est
node.rates <- sim.result$node.rates.est
plot(subst.tree)
axis(1)

# Simulate substitutions with no association with traits (Brownian)
sim.result.brownian <- simulateSubstitutions(time.tree=time.tree,
         sigma=0.5,
         method="AC",
         number.of.subst=10000)
```

`simulateTrait` *Simulate a trait down a tree using Brownian motion (BM)*

Description

Simulate a trait down a tree using Brownian motion (BM)

Usage

```
simulateTrait(time.tree, sigma = 1, root.value = 0)
```

Arguments

`time.tree`	a binary rooted tree (phylo object)
`sigma`	standard deviation for BM
`root.value`	trait value at the root of the tree

Value

A vector of traits, one element for each node in the tree, ordered by node number

Examples

```
# Sample a birth-death tree with 50 taxa
time.tree <- ape::rphylo(birth=10, death=5, n=50)

# Simulate traits down the tree under Brownian motion
traits <- simulateTrait(time.tree)
```

Index