

Package ‘RERconverge’

August 14, 2021

Type Package

Title Analysis of Convergence Between Organismal Traits and
DNA/Protein Sequences

Version 0.1.0

Author Maria Chikina, Nathan Clark, Amanda Kowalczyk, Weiguang (Wayne) Mao, Wynn Meyer, and Raghavendran Partha

Maintainer Maria Chikina <mchikina@pitt.edu> / Nathan Clark <nclark@pitt.edu>

Description RERConverge is a set of software written in R that estimates the correlation between relative evolutionary rates of genes and the evolution of a convergent binary or continuous trait across a phylogeny.

License GNU General Public License v3.0

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 3.5.0), RColorBrewer, gplots, phytools, geiger, knitr,
Rcpp, RcppArmadillo, weights, phangorn, ggplot2, dplyr

Imports Rcpp (>= 0.11.0), ape

LinkingTo Rcpp, RcppArmadillo, ape

RemoteType github

RemoteHost api.github.com

RemoteRepo RERconverge

RemoteUsername nclark-lab

RemoteRef master

RemoteSha db53e0799070b30239768edfd8d596454dfb78d6

GithubRepo RERconverge

GithubUsername nclark-lab

GithubRef master

GithubSHA1 db53e0799070b30239768edfd8d596454dfb78d6

NeedsCompilation yes

R topics documented:

| | |
|---|----|
| calculateCorPermuted | 3 |
| calculatePermulatedPaths_apply | 4 |
| char2Paths | 4 |
| click_select_foreground_branches | 5 |
| combinePermData | 6 |
| correlateWithBinaryPhenotype | 6 |
| correlateWithContinuousPhenotype | 7 |
| edgeVars | 8 |
| estimatePhangornTree | 9 |
| estimatePhangornTreeAll | 10 |
| fastwilcoxGMTall | 11 |
| fixPseudoroot | 11 |
| foreground2Paths | 12 |
| foreground2Tree | 13 |
| foreground2TreeClades | 14 |
| generatePermulatedBinPhen | 14 |
| generatePermulatedBinPhenSSMBatched | 15 |
| getAllCor | 16 |
| getAllResiduals | 17 |
| getEnrichPerms | 18 |
| getForegroundInfoClades | 19 |
| getPermsBinary | 20 |
| getPermsContinuous | 21 |
| getStat | 22 |
| makeBinaryPaths | 23 |
| matchAllNodesClades | 23 |
| namePathsWSpecies | 24 |
| permpvalcor | 24 |
| permpvalenrich | 25 |
| permutevec | 25 |
| plotPositivesFromPermutations | 26 |
| plotRers | 26 |
| plotTreeHighlightBranches | 27 |
| pruneTree | 28 |
| read.gmt | 29 |
| readTrees | 29 |
| RERconverge | 30 |
| returnRersAsNewickStrings | 30 |
| returnRersAsTree | 31 |
| returnRersAsTreesAll | 32 |
| simBinPheno | 32 |
| simBinPhenoCC | 33 |
| simBinPhenoSSM | 34 |
| simBinPhenoVec | 34 |
| simpermvec | 35 |
| simulatevec | 36 |

| | |
|-----------------------------|-----------|
| <i>calculateCorPermuted</i> | 3 |
| tree2Paths | 36 |
| tree2PathsClades | 37 |
| treePlotNew | 38 |
| treePlotRers | 40 |
| Index | 41 |

| | |
|----------------------|---|
| calculateCorPermuted | <i>Calculate permutation correlation statistics</i> |
|----------------------|---|

Description

Calculate permutation correlation statistics

Usage

```
calculateCorPermuted(
  permulated.paths,
  RERmat,
  min.sp = 10,
  min.pos = 2,
  method = "k"
)
```

Arguments

| | |
|------------------|--|
| permulated.paths | A nested list of permulated paths (e.g., output of calculatePermulatedPaths_apply |
| RERmat | An RER matrix calculated using getAllResiduals . |
| min.sp | Minimum number of species that must be present for a gene |
| min.pos | Minimum number of species that must be present in the foreground (non-zero phenotype values) |
| method | Method used to compute correlations. Accepts the same arguments as cor . Set to "auto" to select automatically based on the number of unique values in charP. This will also auto set the winsorization for Pearson correlation. Set winsorizetrait=some number and winsorizeRER=some number to override |

Value

A nested list containing the correlation statistics for the permutations

```
calculatePermulatedPaths_apply
```

Calculates the paths for all permulated trees for a list of genes

Description

Calculates the paths for all permulated trees for a list of genes

Usage

```
calculatePermulatedPaths_apply(permulated.trees.list, map.list, treesObj)
```

Arguments

| | |
|------------------------------------|--|
| <code>permulated.trees.list</code> | A nested list of permulated phenotype trees for all the genes. The top layer lists the genes, and the nested layer lists the permulated binary trees (phylo objects) for each gene |
| <code>map.list</code> | A list of maps corresponding to the genes listed in <code>permulated.trees.list</code> , the output of <code>matchAllNodesClades</code> |
| <code>treesObj</code> | <code>treesObj</code> from readTrees |

Value

`permulated.paths.list` A nested list of permulated paths corresponding to the nested list of permulated phenotype trees

| | |
|-------------------------|---|
| <code>char2Paths</code> | <i>turns a named vector of characters into a paths vector to be used with getAllCor</i> |
|-------------------------|---|

Description

turns a named vector of characters into a paths vector to be used with [getAllCor](#)

Usage

```
char2Paths(
  tip.vals,
  treesObj,
  altMasterTree = NULL,
  metric = "diff",
  se.filter = -1,
  ...
)
```

Arguments

| | |
|------------------------|--|
| <code>tip.vals</code> | the trait/phenotype/character value at the tip, <code>names(tip.vals)</code> should match some of the <code>mastertree\$tip.label</code> , though a perfect match is not required |
| <code>treesObj</code> | A <code>treesObj</code> created by readTrees |
| <code>metric</code> | The metric used to translate node values into branch values. "Diff" takes the difference and makes the result phylogenetically independent. Other possible values are "mean" (the mean of the two values) and "last" the value of the most recent species on the branch. The last two options are not phylogenetically independent and downstream computations for those are not yet implemented |
| <code>se.filter</code> | Will remove branch values that are not at least <code>se.filter*edge.se</code> away from 0 (where <code>edge.se</code> is the standard error in the estimate for the edge value). Only implemented for <code>metric="diff"</code> . Set <code>se.filter</code> to a positive value to filter. By default no filtering is done. |

Value

A vector of length equal to the number of paths in `treesObj`

```
click_select_foreground_branches
```

Interactive click-based function to select foreground branches showing convergent binary trait

Description

Interactive click-based function to select foreground branches showing convergent binary trait

Usage

```
click_select_foreground_branches(mastertree)
```

Arguments

`mastertree.` A phylo tree object defining the topology of all species

Value

A binary trait tree with branch lengths of 1 for selected foreground species and lengths of 0 for the rest

| | |
|-----------------|---|
| combinePermData | <i>Combines batches of permutations</i> |
|-----------------|---|

Description

Combines batches of permutations

Usage

```
combinePermData(permdat1, permdat2, enrich = T)
```

Arguments

| | |
|----------|--|
| permdat1 | Batch of permutations output from getPermsContinuous |
| permdat2 | Batch of permutations output from getPermsContinuous |
| enrich | Default T. Specifies if both 'permdat1' and 'permdat2' contain enrichment permutations |

Value

Combined permutations

| | |
|------------------------------|--|
| correlateWithBinaryPhenotype | <i>Computes the association statistics between RER from getAllResiduals and a phenotype paths vector for a binary phenotype made with tree2Paths</i> |
|------------------------------|--|

Description

Computes the association statistics between RER from [getAllResiduals](#) and a phenotype paths vector for a binary phenotype made with [tree2Paths](#)

Usage

```
correlateWithBinaryPhenotype(
  RERmat,
  charP,
  min.sp = 10,
  min.pos = 2,
  weighted = "auto"
)
```

Arguments

| | |
|----------|--|
| RERmat | RER matrix returned by getAllResiduals |
| charP | phenotype vector returned by tree2Paths or char2Paths |
| min.sp | Minimum number of species that must be present for a gene |
| min.pos | Minimum number of species that must be present in the foreground (non-zero phenotype values) |
| weighted | perform weighted correlation. This option turns on weighted correlation that uses the weights computed by foreground2Tree (wholeClade=T). This setting will treat each clade a single observation for the purpose of p-value estimation. The function will guess automatically if the charP vector is of "weighted" type and there should be not need to set this parameter. |
| method | Method used to compute correlations. Accepts the same arguments as cor . Set to "auto" to select automatically based on the number of unique values in charP. This will also auto set the winsorization for Pearson correlation. Set winsorizetrait=some number and winsorizeRER=some number to override |

correlateWithContinuousPhenotype

Computes the association statistics between RER from [getAllResiduals](#) and a phenotype paths vector for a continuous phenotype made with [char2Paths](#)

Description

Computes the association statistics between RER from [getAllResiduals](#) and a phenotype paths vector for a continuous phenotype made with [char2Paths](#)

Usage

```
correlateWithContinuousPhenotype (
  RERmat,
  charP,
  min.sp = 10,
  winsorizeRER = 3,
  winsorizetrait = 3
)
```

Arguments

| | |
|--------------|---|
| RERmat | RER matrix returned by getAllResiduals |
| charP | phenotype vector returned by tree2Paths or char2Paths |
| min.sp | Minimum number of species that must be present for a gene |
| winsorizeRER | Winsorize RER values before computing Pearson correlation. winsorizeRER=3 will set the 3 most extreme values at each end of each RER row to the the value closest to 0. |

| | |
|----------------|---|
| winsorizetrait | Winsorize trait values before computing Pearson correlation. winsorizetrait=3 will set the 3 most extreme values of the trait values to the value closest to 0. |
| method | Method used to compute correlations. Accepts the same arguments as <code>cor</code> . Set to "auto" to select automatically based on the number of unique values in charP. This will also auto set the winsorization for Pearson correlation. Set winsorizetrait=some number and winsorizeRER=some number to override |

| | |
|----------|--|
| edgeVars | <i>maps a vector of traits onto a reference tree</i> |
|----------|--|

Description

maps a vector of traits onto a reference tree

Usage

```
edgeVars(mastertree, tip.vals, metric = "diff", se.filter = -1, return.var = F)
```

Arguments

| | |
|------------|--|
| mastertree | the tree species the topology of the output tree and the branch lengths are used to infer ancestral states |
| tip.vals | the trait/phenotype/character value at the tip, <code>names(tip.vals)</code> should match some of the <code>mastertree\$tip.label</code> , though a perfect match is not required |
| metric | The metric used to translate node values into branch values. "Diff" takes the difference and makes the result phylogenetically independent. Other possible values are "mean" (the mean of the two values) and "last" the value of the most recent species on the branch. The last two options are not phylogenetically independent and downstream computations for those are not yet implemented |
| se.filter | Will remove branch values that are not at least <code>se.filter*edge.se</code> away from 0 (where <code>edge.se</code> is the standard error in the estimate for the edge value). Only implemented for <code>metric="diff"</code> . Set <code>se.filter</code> to a positive value to filter. By default no filtering is done. |
| return.var | Returns the variance instead of the mean. Useful for seeing which estimates have high confidence. |

Value

A phylo tree with branch values computed from the input tip.values

`estimatePhangornTree`

Estimate a ML tree from a given topology Uses 'pml' and 'optim.pml' from the 'phangorn' package to estimate the tree.

Description

Estimate a ML tree from a given topology Uses 'pml' and 'optim.pml' from the 'phangorn' package to estimate the tree.

Usage

```
estimatePhangornTree(  
  alnfile,  
  treefile,  
  submodel = "LG",  
  type = "AA",  
  format = "fasta",  
  k = 4,  
  ...  
)
```

Arguments

| | |
|-----------------------|---|
| <code>alnfile</code> | The path to the sequence alignment file |
| <code>treefile</code> | The path to the master tree file (whose topology will be used to generate the tree) |
| <code>submodel</code> | Substitution model to be used to estimate the tree |
| <code>type</code> | "AA" for amino acid or "DNA" for DNA |
| <code>format</code> | Format of the alignment file (commonly used formats include fasta and phylip) |
| <code>k</code> | Number of intervals in the discrete gamma distribution for 'pml' |
| <code>...</code> | Further arguments passed to 'pml' or 'optim.pml' |

Value

A list: 'tree.opt' is the tree from the optimized output of 'optim.pml'; 'results.opt' is the optimized output of 'optim.pml'; 'results.init' is the initial results estimated by 'pml'

See Also

[phyDat](#) for alignment formats, [pml](#) and [optim.pml](#) for tree estimation

```
estimatePhangornTreeAll
```

Estimates a ML trees from a given topology for a list of alignments. Wirtes a text file compatble with readTrees(). Uses 'pml' and 'optim.pml' from the 'phangorn' package to estimate the tree.

Description

Estimates a ML trees from a given topology for a list of alignments. Wirtes a text file compatble with readTrees(). Uses 'pml' and 'optim.pml' from the 'phangorn' package to estimate the tree.

Usage

```
estimatePhangornTreeAll(
  alnfiles = NULL,
  alndir = NULL,
  pattern = NULL,
  treefile,
  output.file = NULL,
  submodel = "LG",
  type = "AA",
  format = "fasta",
  k = 4,
  ...
)
```

Arguments

| | |
|--------------------------|---|
| <code>alnfiles</code> | A character vector of paths to alignment files. Such as one produced by <code>list.files</code> . Either <code>alnfiles</code> or <code>alndir</code> must be supplied. |
| <code>alndir</code> | The path to directories with alignment files. Trees will be named with the alignment file name after stripping the extension. Either <code>alnfiles</code> or <code>alndir</code> must be supplied. |
| <code>pattern</code> | An optional regular expression for files in the <code>alndir</code> director. As in <code>"*.fasta"</code> . |
| <code>treefile</code> | The path to the master tree file (whose topology will be used to generate the tree) |
| <code>output.file</code> | The file where the output trees will be written. This file can be read with <code>readTrees()</code> . |
| <code>submodel</code> | Substitution model to be used to estimate the tree |
| <code>type</code> | "AA" for amino acid or "DNA" for DNA |
| <code>format</code> | Format of the alignment file (commonly used formats include fasta and phylip) |
| <code>k</code> | Number of intervals in the discrete gamma distribution for 'pml' |
| <code>...</code> | Further arguments passed to 'pml' or 'optim.pml' |

Value

A list: ‘tree.opt’ is the tree from the optimized output of ‘optim.pml’; ‘results.opt’ is the optimized output of ‘optim.pml’; ‘results.init’ is the initial results estimated by ‘pml’

See Also

[phyDat](#) for alignment formats, [pml](#) and [optim.pml](#) for tree estimation

| | |
|------------------|---|
| fastwilcoxGMTall | <i>Performs Wilcoxon Rank-Sum pathway enrichment on pathway gene-sets</i> |
|------------------|---|

Description

Performs Wilcoxon Rank-Sum pathway enrichment on pathway genesets

Usage

```
fastwilcoxGMTall(vals, annotList, ...)
```

Arguments

| | |
|-----------|---|
| vals | The output from getStat run on RERconverge correlation functions (correlate-WithContinuousPhenotype, correlateWithBinaryPhenotype, getAllCor) |
| annotList | Pathway annotations formatted as a list of lists of annotation information |

Value

A list of pathways with enrichment statistics

| | |
|---------------|--|
| fixPseudoroot | <i>pseudoroot trait tree to match the psuedoroot of trees from readTrees if the trees are reconcilable</i> |
|---------------|--|

Description

pseudoroot trait tree to match the psuedoroot of trees from [readTrees](#) if the trees are reconcilable

Usage

```
fixPseudoroot(tree, treesObj)
```

Arguments

| | |
|----------|--|
| tree | A trait tree with branch lengths representing trait values |
| treesObj | A treesObject created by readTrees |

Value

A trait tree with the correct topology

foreground2Paths *Creates paths from a set of foreground species*

Description

Creates paths from a set of foreground species

Usage

```
foreground2Paths(
  foreground,
  treesObj,
  plotTree = F,
  clade = c("ancestral", "terminal", "all", "weighted"),
  useSpecies = NULL
)
```

Arguments

| | |
|-------------|--|
| treesObj | A treesObj created by readTrees |
| plotTree | Plot a tree representation of the result |
| clade | A character string indicating which branches within the clade containing the foreground species should be set to foreground. Must be one of the strings "ancestral", "terminal", "all", or "weighted". |
| useSpecies | Give only a subset of the species to use for ancestral state reconstruction (e.g., only those species for which the trait can be reliably determined). |
| foreground. | A character vector containing the foreground species |

Value

A vector of length equal to the number of paths in treesObj

| | |
|-----------------|--|
| foreground2Tree | <i>Creates a binary trait tree from a set of foreground species.</i> |
|-----------------|--|

Description

Creates a binary trait tree from a set of foreground species.

Usage

```
foreground2Tree(
  foreground,
  treesObj,
  plotTree = T,
  clade = c("ancestral", "terminal", "all"),
  weighted = F,
  transition = "unidirectional",
  useSpecies = NULL
)
```

Arguments

| | |
|--------------|---|
| treesObj | A treesObj created by readTrees |
| plotTree | Plot a tree representation of the result |
| clade | A character string indicating which branches within the clade containing the foreground species should be set to foreground. Must be one of the strings "ancestral", "terminal", "all". |
| weighted | if set to TRUE weights foreground edges belonging to the same clade such that their branch lengths sum up to 1 (only done for clade options "all" and "terminal"). |
| transition | A character string indicating whether transitions between background and foreground branches are "bidirectional" or "unidirectional" (no foreground to background transitions, the default) |
| useSpecies | Give only a subset of the species to use for ancestral state reconstruction (e.g., only those species for which the trait can be reliably determined). |
| foreground. | A character vector containing the foreground species |
| collapse2anc | Put all the weight on the ancestral branch when the trait appears on a whole clade (redundant to "clade", kept for backwards compatibility) |
| wholeClade | Whether to implement the weighted edge option across all members of a foreground clade (redundant to "clade", kept for backwards compatibility) |

Value

A tree with edge.lengths representing phenotypic states

```
foreground2TreeClades
```

Generates a binary phenotype tree using the list of tip foreground animals, foreground common ancestors, and their phylogenetic relationships

Description

Generates a binary phenotype tree using the list of tip foreground animals, foreground common ancestors, and their phylogenetic relationships

Usage

```
foreground2TreeClades (
  fg_vec,
  sisters_list = NULL,
  trees,
  plotTree = T,
  useSpecies = NULL
)
```

Arguments

| | |
|---------------------------|---|
| <code>fg_vec</code> | A vector containing the tip foreground species |
| <code>sisters_list</code> | A list containing pairs of "sister species" in the foreground set (put NULL if empty) |
| <code>trees</code> | <code>treesObj</code> from readTrees |
| <code>plotTree</code> | A boolean indicator for plotting the output tree (default=FALSE) |
| <code>useSpecies</code> | An array containing the tip labels in the output tree |

Value

`fg.tree` A binary phenotype tree corresponding to the input information

```
generatePermulatedBinPhen
```

Produces binary permutations for a gene

Description

Produces binary permutations for a gene

Usage

```
generatePermulatedBinPhen(
  tree,
  numperms,
  trees,
  root_sp,
  fg_vec,
  sisters_list,
  pathvec,
  permmode = "cc"
)
```

Arguments

| | |
|--------------|---|
| tree | Tree of the gene of interest (if permmode="cc", set this as the masterTree in trees (i.e., the output from readTrees)) |
| numperms | An integer number of permutations |
| trees | treesObj from readTrees |
| root_sp | The species to root the tree on |
| fg_vec | A vector containing the foreground species |
| sisters_list | A list containing pairs of "sister species" in the foreground set (put NULL if empty) |
| pathvec | A path vector generated from the real set of foreground animals |

Value

output.list a list containing the set of binary permulated trees

```
generatePermulatedBinPhenSSMBatched
```

Produces binary SSM permutations for a list of genes

Description

Produces binary SSM permutations for a list of genes

Usage

```
generatePermulatedBinPhenSSMBatched(
  trees_list,
  numperms,
  trees,
  root_sp,
  fg_vec,
  sisters_list,
  pathvec
)
```

Arguments

| | |
|--------------|---|
| trees_list | A list containing the trees of all genes of interest (formatted like trees in treesObj from readTrees) |
| numperms | An integer number of permutations |
| trees | treesObj from readTrees |
| root_sp | The species to root the tree on |
| fg_vec | A vector containing the foreground species |
| sisters_list | A list containing pairs of "sister species" in the foreground set (put NULL if empty) |
| pathvec | A path vector generated from the real set of foreground animals |

Value

simPhenoList A list containing binary permulated trees for each gene

| | |
|-----------|---|
| getAllCor | <i>Computes the association statistics between RER from getAllResiduals and a phenotype paths vector made with tree2Paths or char2Paths</i> |
|-----------|---|

Description

Computes the association statistics between RER from [getAllResiduals](#) and a phenotype paths vector made with [tree2Paths](#) or [char2Paths](#)

Usage

```
getAllCor(
  RERmat,
  charP,
  method = "auto",
  min.sp = 10,
  min.pos = 2,
  winsorizeRER = NULL,
  winsorizetrait = NULL,
  weighted = F
)
```

Arguments

| | |
|--------|--|
| RERmat | RER matrix returned by getAllResiduals |
| charP | phenotype vector returned by tree2Paths or char2Paths |
| method | Method used to compute correlations. Accepts the same arguments as cor . Set to "auto" to select automatically based on the number of unique values in charP. This will also auto set the winsorization for Pearson correlation. Set winsorizetrait=some number and winsorizeRER=some number to override |

| | |
|----------------|---|
| min.sp | Minimum number of species that must be present for a gene |
| min.pos | Minimum number of species that must be present in the foreground (non-zero phenotype values) |
| winsorizeRER | Winsorize RER values before computing Pearson correlation. winsorizeRER=3 will set the 3 most extreme RER values at each end of each row to the value closest to 0. |
| winsorizetrait | Winsorize trait values before computing Pearson correlation. winsorizetrait=3 will set the 3 most extreme trait values at each end to the value closest to 0. |
| weighted | perform weighted correlation. This option needs to be set if the clade weights computed in foreground2Tree (wholeClade=T) are to be used. This setting will treat the clade a single observation for the purpose of p-value estimation. |

Value

A list object with correlation values, p-values, and the number of data points used for each tree

Note

winsorize is in terms of number of observations at each end, NOT quantiles

| | |
|-----------------|--------------------------------------|
| getAllResiduals | <i>main RER computation function</i> |
|-----------------|--------------------------------------|

Description

main RER computation function

Usage

```
getAllResiduals(
  treesObj,
  cutoff = NULL,
  transform = "sqrt",
  weighted = T,
  useSpecies = NULL,
  min.sp = 10,
  scale = T,
  doOnly = NULL,
  maxT = NULL,
  scaleForPproj = F,
  mean.trim = 0.05,
  plot = T
)
```

Arguments

| | |
|------------|---|
| treesObj | A treesObj created by readTrees |
| transform | The transformation to apply to the trees branch values before computing relative rates. Available options are sqrt and log, sqrt is recommended. |
| weighted | Use weighted regression to compute relative rates, meant to correct for the non-constant mean-variance relationship in evolutionary rate data. |
| useSpecies | Give only a subset of the species to use for RER calculation. Some times excluding unusually long branches can provide more stable results |
| min.sp | The minimum number of species needed to compute RER |
| scale | Scale relative rates internally for each species subset. Increases computation time with little apparent benefit. Better to scale the final matrix. |
| doOnly | The index of a specific tree in the treesObj to calculate RER for. Useful if a single result is needed quickly. |
| maxT | The maximum number of trees to compute results for. Since this function takes some time this is useful for debugging. |
| plot | Whether to plot the output of the correction for mean-variance relationship. |
| a | cutoff value for branch lengths bellow which the branch lengths will be discarded, very data dependent but should roughly correspond to 0 or 1 sequence change on that branch. If left NULL this will be set to the bottom 0.05 quantile. Set to 0 for no cutoff. |

Value

A numer of trees by number of paths matrix of relative evolutionary rates. Only an independent set of paths has non-NA values for each tree.

| | |
|----------------|--|
| getEnrichPerms | <i>Performs enrichment statistic permutations using existing gene correlation permutations</i> |
|----------------|--|

Description

Performs enrichment statistic permutations using existing gene correlation permutations

Usage

```
getEnrichPerms(corperms, realenrich, annotlist)
```

Arguments

| | |
|------------|---|
| corperms | Gene correlation permutations from getPermsContinuous |
| realenrich | Pathway enrichment results using observed phenotype obtained from ‘correlate-WithContinuousPhenotype’ or ‘correlateWithBinaryPhenotype’ |
| annotlist | Pathway annotations |

Value

Full null permutation statistics and p-values for gene correlations and pathway enrichment

```
getForegroundInfoClades
```

Generates a binary phenotype tree and foreground clades information using the list of tip foreground animals, the presence of foreground common ancestors, and their phylogenetic relationships

Description

Generates a binary phenotype tree and foreground clades information using the list of tip foreground animals, the presence of foreground common ancestors, and their phylogenetic relationships

Usage

```
getForegroundInfoClades (
  fg_vec,
  sisters_list = NULL,
  trees,
  plotTree = T,
  useSpecies = NULL
)
```

Arguments

| | |
|---------------------------|---|
| <code>fg_vec</code> | A vector containing the tip foreground species |
| <code>sisters_list</code> | A list containing pairs of "sister species" in the foreground set (put NULL if empty) |
| <code>trees</code> | treesObj from readTrees |
| <code>plotTree</code> | A boolean indicator for plotting the output tree (default=FALSE) |
| <code>useSpecies</code> | An array containing the tip labels in the output tree |

Value

output.list A list containing 1) "tree" = a binary phenotype tree corresponding to the input information, 2) "fg.sisters.table" = a table containing all sister species in the foreground set

| | |
|----------------|---|
| getPermsBinary | <i>Calculates permuted correlation and enrichment statistics for binary phenotype</i> |
|----------------|---|

Description

Calculates permuted correlation and enrichment statistics for binary phenotype

Usage

```
getPermsBinary(
  numperms,
  fg_vec,
  sisters_list,
  root_sp,
  RERmat,
  trees,
  mastertree,
  permmode = "cc",
  method = "k",
  min.pos = 2,
  trees_list = NULL,
  calculateenrich = F,
  annotlist = NULL
)
```

Arguments

| | |
|--------------|--|
| numperms | An integer number of permutations |
| fg_vec | A vector containing the tip foreground species |
| sisters_list | A list containing pairs of "sister species" in the foreground set (put NULL if empty) |
| root_sp | The species to root the tree on |
| RERmat | An RER matrix calculated using getAllResiduals . |
| trees | treesObj from readTrees |
| mastertree | A rooted, fully dichotomous tree derived from the treesObj master tree from readTrees . Must not contain species not in traitvec |
| permmode | Mode of binary permutation ("cc" for Complete Cases (default), "ssm" for Species Subset Match) |
| method | statistical method to use for correlations (set to "k" (default) for Kendall Tau test) |
| min.pos | minimum number of foreground species (default 2) |
| trees_list | A list containing the trees of all genes of interest (formatted like trees in treesObj from readTrees) |

| | |
|-----------------|--|
| calculateenrich | A boolean variable indicating if null permutation p-values for enrichment statistics |
| annotlist | Pathway annotations |

Value

A list object with enrichment statistics, correlation p-val, rho, and correlation effect size

getPermsContinuous *Calculates permuted correlation and enrichment statistics*

Description

Calculates permuted correlation and enrichment statistics

Usage

```
getPermsContinuous (
  numperms,
  traitvec,
  RERmat,
  annotlist,
  trees,
  mastertree,
  calculateenrich = T,
  type = "simperm",
  winR = 3,
  winT = 3,
  method = "p",
  min.pos = 0
)
```

Arguments

| | |
|-----------------|--|
| numperms | An integer number of permutations |
| traitvec | A named phenotype vector |
| RERmat | An RER matrix calculated using getAllResiduals |
| annotlist | Pathway annotations |
| trees | treesObj from readTrees |
| mastertree | A rooted, fully dichotomous tree derived from the treesObj master tree from readTrees . Must not contain species not in traitvec |
| calculateenrich | A boolean variable indicating if null permutation p-values for enrichment statistics |

| | |
|---------|---|
| type | One of "simperm", "sim", or "perm" for permutations, simulations, or permutations, respectively |
| winR | Integer winzORIZATION value for RERmat |
| winT | Integer winzORIZATION value for trait |
| method | statistical method to use for correlations |
| min.pos | minimum foreground species - should be set to 0 |

Value

A list object with enrichment statistics, correlation p-val, rho, and correlation effect size

Note

winsorize is in terms of number of observations at each end, NOT quantiles

| | |
|---------|--|
| getStat | <i>Calculates Rho-signed negative log-base-ten p-value for use in enrichment functions</i> |
|---------|--|

Description

Calculates Rho-signed negative log-base-ten p-value for use in enrichment functions

Usage

```
getStat(res)
```

Arguments

| | |
|-----|---|
| res | The output from RERconverge correlation functions (correlateWithContinuousPhenotype, correlateWithBinaryPhenotype, getAllCor) |
|-----|---|

Value

A dataframe of Rho-signed negative log-base-ten p-values for all genes, NAs removed

| | |
|-----------------|--|
| makeBinaryPaths | <i>Makes a binary path vector from either a tree of class "phylo" or a foreground species set supplied as a character vector</i> |
|-----------------|--|

Description

Makes a binary path vector from either a tree of class "phylo" or a foreground species set supplied as a character vector

Usage

```
makeBinaryPaths(input, treesObj)
```

Arguments

| | |
|----------|---|
| input | Either a phenotype tree of class "phylo" (with branch length encoding a phenotype) or a character vector of foreground branches |
| treesObj | A treesObj created by readTrees |

Value

A vector of length equal to the number of paths in treesObj

| | |
|---------------------|---|
| matchAllNodesClades | <i>Calculates the clade mappings between the gene tree and the master tree (with the complete topology)</i> |
|---------------------|---|

Description

Calculates the clade mappings between the gene tree and the master tree (with the complete topology)

Usage

```
matchAllNodesClades(gene_tree, treesObj)
```

Arguments

| | |
|-----------|---|
| gene_tree | A binary phenotype tree of a gene |
| treesObj | treesObj from readTrees |

Value

output.map A list containing a dataframe of clades mapping

| | |
|--------------------------------|--|
| <code>namePathsWSpecies</code> | <i>Provides names for paths/RERs representing terminal branches for plotting Originally an internal function but necessary for the vignette/walk-through</i> |
|--------------------------------|--|

Description

Provides names for paths/RERs representing terminal branches for plotting Originally an internal function but necessary for the vignette/walk-through

Usage

```
namePathsWSpecies(masterTree)
```

Arguments

| | |
|-------------------------|-----------------------------------|
| <code>masterTree</code> | The master tree used for analysis |
|-------------------------|-----------------------------------|

Value

Names corresponding to the paths/RERs for terminal branches

| | |
|--------------------------|---|
| <code>permpvalcor</code> | <i>Calculates correlation permutation pvals from output of getPermsContinuous</i> |
|--------------------------|---|

Description

Calculates correlation permutation pvals from output of [getPermsContinuous](#)

Usage

```
permpvalcor(realcor, permvals)
```

Arguments

| | |
|-----------------------|--|
| <code>realcor</code> | Real enrichment statistics from fastwilcoxGMTall |
| <code>permvals</code> | output from getPermsContinuous |

Value

A vector with permutation p-values

| | |
|----------------|--|
| permpvalenrich | <i>Calculates enrichment permutation pvals from output of getPermsContinuous</i> |
|----------------|--|

Description

Calculates enrichment permutation pvals from output of [getPermsContinuous](#)

Usage

```
permpvalenrich(realenrich, permvals)
```

Arguments

| | |
|------------|--|
| realenrich | Real enrichment statistics from fastwilcoxGMTall |
| permvals | output from getPermsContinuous |

Value

A list object with vectors of permutation p-values

| | |
|------------|---|
| permutevec | <i>Generates a permuted continuous phenotype given an observed continuous phenotype</i> |
|------------|---|

Description

Generates a permuted continuous phenotype given an observed continuous phenotype

Usage

```
permutevec(namedvec)
```

Arguments

| | |
|----------|---|
| namedvec | A named numeric vector with phenotype values for each speices |
|----------|---|

Value

A vector with permuted phenotype values

```
plotPositivesFromPermutations
```

Plots changes in number of statistically significant rate acceleration or deceleration versus the number of permutations

Description

Plots changes in number of statistically significant rate acceleration or deceleration versus the number of permutations

Usage

```
plotPositivesFromPermutations (
  res,
  perm.out,
  interval,
  pvalthres,
  output.res = FALSE
)
```

Arguments

| | |
|-------------------------|--|
| <code>res</code> | correlation statistic output from <code>correlateWithBinaryPhenotype</code> or <code>correlateWithContinuous</code> |
| <code>perm.out</code> | output from <code>getPermsBinary</code> or <code>getPermsContinuous</code> |
| <code>interval</code> | interval of number of permutations (e.g., <code>interval = 10</code> means that number of positives will be calculated for number of permutations = 10, 20, 30, ...) |
| <code>pvalthres</code> | p-value threshold for identifying statistically significant rate acceleration or deceleration |
| <code>output.res</code> | Boolean defining whether to output a list object containing changes in the number of identified elements with changing number of permutations (default=FALSE) |

Value

A list containing changes in the number of identified elements with changing number of permutations (default=NULL)

```
plotRers
```

Plot the residuals reflecting the relative evolutionary rates (RERs) of a gene across species present in the gene tree

Description

Plot the residuals reflecting the relative evolutionary rates (RERs) of a gene across species present in the gene tree

Usage

```
plotRers (
  rermat = NULL,
  index = NULL,
  phenv = NULL,
  rers = NULL,
  method = "k",
  xlims = NULL,
  plot = 1,
  xextend = 0.2,
  sortrers = F
)
```

Arguments

| | |
|---------|---|
| rermat. | A residual matrix, output of the <code>getAllResiduals()</code> function |
| index. | A character denoting the name of gene, or a numeric value corresponding to the gene's row index in the residuals matrix |
| phenv. | A phenotype vector returned by tree2Paths or foreground2Paths |

Value

A plot of the RERs with foreground species labelled in red, and the rest in blue

```
plotTreeHighlightBranches
```

Plot the provided tree, (optionally) rerooted, with specified branches highlighted

Description

Plot the provided tree, (optionally) rerooted, with specified branches highlighted

Usage

```
plotTreeHighlightBranches (
  tree,
  outgroup = NULL,
  hlspecies,
  hlcols = NULL,
  main = "",
  useGG = FALSE
)
```

Arguments

| | |
|-------------------------|--|
| <code>useGG</code> | option to creat plot with ggtree. Improves plot readability for most data sets. |
| <code>tree.</code> | A tree object. |
| <code>outgroup.</code> | A vector of species to use to root the tree. If not provided, the tree remains unrooted. |
| <code>hlspecies.</code> | A vector of species whose terminal branches to highlight, or a vector of branch numbers within the tree. |
| <code>hlcols.</code> | Colors to use in highlighting the branches. If not specified, will use default R colors. |
| <code>main.</code> | Main text for plot. |

Value

A plot of the the (optionally rerooted) tree, with branches highlighted.

| | |
|------------------------|--|
| <code>pruneTree</code> | <i>wrapper around</i> drop.tip |
|------------------------|--|

Description

wrapper around [drop.tip](#)

Usage

```
pruneTree(tree, tip.names)
```

Arguments

| | |
|------------------------|-----------------------------------|
| <code>tree</code> | a "phylo" tree |
| <code>tip.names</code> | The tip names to keep in the tree |

Value

A new pruned tree

| | |
|----------|--|
| read.gmt | <i>Reads pathway data from a gmt-formatted file from GSEA-MSigDB</i> |
|----------|--|

Description

Reads pathway data from a gmt-formatted file from GSEA-MSigDB

Usage

```
read.gmt(filename)
```

Arguments

| | |
|----------|--------------------------------|
| filename | The gmt-formatted pathway file |
|----------|--------------------------------|

Value

A list of pathway information with pathway names and genesets

| | |
|-----------|---|
| readTrees | <i>reads trees from a 2 column , tab seperated, file The first columns is the gene name and the second column is the corresponding tree in parenthetic format known as the Newick or New Hampshire format</i> |
|-----------|---|

Description

reads trees from a 2 column , tab seperated, file The first columns is the gene name and the second column is the corresponding tree in parenthetic format known as the Newick or New Hampshire format

Usage

```
readTrees (
  file,
  max.read = NA,
  masterTree = NULL,
  minTreesAll = 20,
  reestimateBranches = F,
  minSpecs = NULL
)
```

Arguments

| | |
|---------------------------------|--|
| <code>file</code> | The path to the tree file |
| <code>max.read</code> | This function takes a while for a whole genome, so <code>max.read</code> is useful for testing |
| <code>masterTree</code> | (optional) User can specify a master tree. Recommended only when the number of available gene trees with all species is small. |
| <code>minTreesAll</code> | The minimum number of trees with all species present in order to estimate master tree edge lengths (default 20). |
| <code>reestimateBranches</code> | Boolean indicating whether to re-estimate branch lengths if master tree topology is included (default FALSE) |
| <code>minSpecs</code> | the minimum number of species that needs to be present in a gene tree to be included in calculating master tree |

Value

A trees object of class "treeObj"
A trees object of class "treeObj"

| | |
|--------------------------|--------------------|
| <code>RERconverge</code> | <i>RERconverge</i> |
|--------------------------|--------------------|

Description

`RERconverge`
`RERconverge`

| | |
|--|---|
| <code>returnRersAsNewickStrings</code> | <i>Produce a vector of newick strings representing gene trees where the edge lengths correspond to RERs</i> |
|--|---|

Description

Produce a vector of newick strings representing gene trees where the edge lengths correspond to RERs

Usage

```
returnRersAsNewickStrings(treesObj, rermat)
```

Arguments

| | |
|------------------------|--|
| <code>treesObj.</code> | A treesObj created by <code>readTrees</code> |
| <code>rermat.</code> | A residual matrix, output of the <code>getAllResiduals()</code> function |

Value

A named character vector of newick strings, one per gene, representing RERs as edge lengths'

| | |
|------------------|---|
| returnRersAsTree | <i>Produce a gene tree with branch lengths representing RERs and (optionally) display these RERs as branch labels</i> |
|------------------|---|

Description

Produce a gene tree with branch lengths representing RERs and (optionally) display these RERs as branch labels

Usage

```
returnRersAsTree (
  treesObj,
  rermat,
  index,
  phenv = NULL,
  rer.cex = 0.7,
  tip.cex = 0.7,
  nalab = "NA",
  plot = T
)
```

Arguments

| | |
|-----------|---|
| treesObj. | A treesObj created by readTrees |
| rermat. | A residual matrix, output of the <code>getAllResiduals()</code> function |
| index. | A character denoting the name of gene, or a numeric value corresponding to the gene's row index in the residuals matrix |
| phenv. | A phenotype vector returned by tree2Paths or foreground2Paths |
| rer.cex. | Numeric expansion for RER labels |
| tip.cex. | Numeric expansion for tip labels |
| nalab. | Label given to any NA RERs |
| plot. | Whether to produce a plot displaying the RERs on the gene tree |

Value

An object of class "phylo" with edge lengths representing RERs for the given gene
 If plot = TRUE, also displays a plot of the gene tree with edges labeled with RERs

```
returnRersAsTreesAll
```

Produce a multiPhylo object of all gene trees with branch lengths representing RERs for each gene

Description

Produce a multiPhylo object of all gene trees with branch lengths representing RERs for each gene

Usage

```
returnRersAsTreesAll(treesObj, rermat)
```

Arguments

| | |
|------------------------|--|
| <code>treesObj.</code> | A treesObj created by readTrees |
| <code>rermat.</code> | A residual matrix, output of the <code>getAllResiduals()</code> function |

Value

An object of class "multiPhylo" of named gene trees with edge lengths representing RERs for the given gene

```
simBinPheno
```

Generates a permulated phylogenetic tree with specified number of foreground branches. User may specify the number of foreground branches that are internal branches.

Description

Generates a permulated phylogenetic tree with specified number of foreground branches. User may specify the number of foreground branches that are internal branches.

Usage

```
simBinPheno(trees, root, phenvec, fgnum = NULL, internal = 0, drop = NULL)
```

Arguments

| | |
|----------------------|---|
| <code>trees</code> | treesObj output from readTrees |
| <code>root</code> | Species on which to root the master tree |
| <code>phenvec</code> | Named vector of 1's and 0's representing phenotype values for each species |
| <code>fgnum</code> | Total number of foreground species - only required if internal foreground branches are required |

| | |
|----------|--|
| internal | Number of foreground species that should be internal branches - only required if internal foreground branches are required |
| drop | Character vector (or single character variable) of species names to be removed from the master tree (such as species in trees but not in phenotype vector) |

Value

A tree with permulated phenotype values

| | |
|---------------|--|
| simBinPhenoCC | <i>Produces one CC binary permutation for a gene</i> |
|---------------|--|

Description

Produces one CC binary permutation for a gene

Usage

```
simBinPhenoCC (
  trees,
  mastertree,
  root_sp,
  fg_vec,
  sisters_list = NULL,
  pathvec,
  plotTreeBool = F
)
```

Arguments

| | |
|--------------|---|
| trees | treesObj from readTrees |
| mastertree | A rooted, fully dichotomous tree derived from the treesObj master tree from readTrees . |
| root_sp | The species to root the tree on |
| fg_vec | a vector containing the foreground species |
| sisters_list | A list containing pairs of "sister species" in the foreground set (put NULL if empty) |
| pathvec | A path vector generated from the real set of foreground animals |
| plotTreeBool | Boolean indicator for plotting the output tree (default=FALSE) |

Value

A CC binary permulated tree

| | |
|----------------|---|
| simBinPhenoSSM | <i>Produces one SSM binary permutation for a gene</i> |
|----------------|---|

Description

Produces one SSM binary permutation for a gene

Usage

```
simBinPhenoSSM(
  tree,
  trees,
  root_sp,
  fg_vec,
  sisters_list = NULL,
  pathvec,
  plotTreeBool = F
)
```

Arguments

| | |
|--------------|---|
| tree | Tree of the gene of interest |
| trees | treesObj from readTrees |
| root_sp | The species to root the tree on |
| fg_vec | A vector containing the foreground species |
| sisters_list | A list containing pairs of "sister species" in the foreground set (put NULL if empty) |
| pathvec | A path vector generated from the real set of foreground animals |
| plotTreeBool | Boolean indicator for plotting the output tree (default=FALSE) |

Value

A SSM binary permulated tree

| | |
|----------------|--|
| simBinPhenoVec | <i>Generates a permulated phenotype vector whose phylogeny matches a desired structure. User may specify the number of foreground branches that are internal branches.</i> |
|----------------|--|

Description

Generates a permulated phenotype vector whose phylogeny matches a desired structure. User may specify the number of foreground branches that are internal branches.

Usage

```
simBinPhenoVec(trees, root, phenvec, fgnum = NULL, internal = 0, drop = NULL)
```

Arguments

| | |
|----------|--|
| trees | treesObj output from readTrees |
| root | Species on which to root the master tree |
| phenvec | Named vector of 1's and 0's representing phenotype values for each species |
| fgnum | Total number of foreground species - only required if internal foreground branches are required |
| internal | Number of foreground species that should be internal branches - only required if internal foreground branches are required |
| drop | Character vector (or single character variable) of species names to be removed from the master tree (such as species in trees but not in phenotype vector) |

Value

A vector of permulated foreground species

| | |
|------------|---|
| simpermvec | <i>Generates a permulated continuous phenotype given an observed continuous phenotype and a phylogeny</i> |
|------------|---|

Description

Generates a permulated continuous phenotype given an observed continuous phenotype and a phylogeny

Usage

```
simpermvec(namedvec, treewithbranchlengths)
```

Arguments

| | |
|-----------------------|---|
| namedvec | A named numeric vector with phenotype values for each speices |
| treewithbranchlengths | A rooted phylogenetic tree with the same species as namedvec and branch lengths representing average evolutionary rate. The master tree from readTrees may be rooted and used for this parameter. |

Value

A vector with permulated phenotype values

| | |
|--------------------------|--|
| <code>simulatevec</code> | <i>Generates a simulated continuous phenotype given an observed continuous phenotype and a phylogeny</i> |
|--------------------------|--|

Description

Generates a simulated continuous phenotype given an observed continuous phenotype and a phylogeny

Usage

```
simulatevec(namedvec, treewithbranchlengths)
```

Arguments

| | |
|------------------------------------|--|
| <code>namedvec</code> | A named numeric vector with phenotype values for each species |
| <code>treewithbranchlengths</code> | A rooted phylogenetic tree with the same species as <code>namedvec</code> and branch lengths representing average evolutionary rate. The master tree from readTrees may be rooted and used for this parameter. |

Value

A vector with simulated phenotype values

| | |
|-------------------------|--|
| <code>tree2Paths</code> | <i>Generate a phenotype paths vector from a phenotype tree</i> |
|-------------------------|--|

Description

`tree2Paths` generates a phenotype paths vector matching the `treesObject` from a tree where branches specify phenotypes.

Usage

```
tree2Paths(tree, treesObj, binarize = NULL, useSpecies = NULL)
```

Arguments

| | |
|-----------------------|--|
| <code>tree</code> | A phenotype tree, with branch length encoding a phenotype. |
| <code>treesObj</code> | A <code>treesObject</code> created by readTrees |
| <code>binarize</code> | Force binary path representation. Default action depends upon the type of data within the phenotype tree (binary or continuous). |

- If binary (all branch lengths == 0 or 1): Sets all positive path values to 1. Useful if the tree has non-zero branch lengths for an internal branch or branches; otherwise, values are simply added along branches when calculating paths. Default behavior: `binarize = TRUE`.
- If continuous (not all branch lengths == 0 or 1): Sets all path values > the mean to 1 and all those <= the mean to 0. Converts a continuous phenotype to a binary phenotype, with state determined by comparison to the mean across all paths. Default behavior: `binarize = FALSE`.

`useSpecies` Give only a subset of the species to use for ancestral state reconstruction (e.g., only those species for which the trait can be reliably determined).

Details

The tree topology of the phenotype tree must match that of the master tree within the `treesObj`.

Value

A vector of length equal to the number of paths in `treesObj`

| | |
|-------------------------------|---|
| <code>tree2PathsClades</code> | <i>A modification of the <code>tree2Paths</code> function that takes in pre-calculated mappings</i> |
|-------------------------------|---|

Description

A modification of the `tree2Paths` function that takes in pre-calculated mappings

Usage

```
tree2PathsClades(tree, trees)
```

Arguments

| | |
|--------------------|---|
| <code>tree</code> | the input tree to be converted into paths |
| <code>trees</code> | <code>treesObj</code> from <code>readTrees</code> |

| | |
|-------------|--|
| treePlotNew | <i>Plot 'tree' with branch labels colored in a heatmap based on values in 'vals'</i> |
|-------------|--|

Description

Plot 'tree' with branch labels colored in a heatmap based on values in 'vals'

Usage

```
treePlotNew(
  tree,
  vals = NULL,
  rank = F,
  nlevels = 9,
  type = "c",
  col = NULL,
  maintitle = NULL,
  useedge = F,
  doreroot = F,
  rerootby = NULL,
  useSpecies = NULL,
  species.names = NULL,
  speclist1 = NULL,
  speclist2 = NULL,
  aligntip = F,
  colpan1 = rgb(0, 119, 187, maxColorValue = 255),
  colpan2 = rgb(204, 51, 17, maxColorValue = 255),
  colpanmid = rgb(187, 187, 187, maxColorValue = 255),
  plotspecies = NULL,
  edgetype = NULL,
  textsize = 0.6,
  colbarlab = "",
  splist2sym = "psi",
  dolegend = T,
  nacol = rgb(0, 0, 0),
  figwid = 10,
  ...
)
```

Arguments

| | |
|----------|---|
| tree. | A phylo object, used for topology |
| vals. | Values to use for heatmap to color branches, in same order as edges of tree |
| rank. | Whether to plot ranks instead of values |
| nlevels. | How many colors to use in the heatmap |

| | |
|-----------------------------|--|
| <code>type.</code> | Type for <code>plot.phylo</code> |
| <code>col.</code> | Vector of user-defined colors |
| <code>maintitle.</code> | Main title label |
| <code>useedge.</code> | Whether to use edge lengths in ‘tree’ for plotting |
| <code>doreroor.</code> | Whether to re-root the tree before plotting |
| <code>rerootby.</code> | If re-rooting, what to use to root the tree (verify by checking against unrooted plot) |
| <code>useSpecies.</code> | A vector of species to include in the plot (verify by checking against full set) |
| <code>species.names.</code> | Data fram for converting names in ‘tree’ to names to be plotted (row names should be tip labels of tree, and first column should contain the corresponding desired tip labels) |
| <code>speclist1.</code> | A vector of tip labels to highlight in bold, blue text |
| <code>speclist2.</code> | A vector of tip labels to which to add an asterisk |
| <code>aligntip.</code> | Whether to align tip labels (default FALSE) |
| <code>colpan1.</code> | Color for lowest value in heatmap (default blue) |
| <code>colpan2.</code> | Color for highest value in heatmap (default rose) |
| <code>colpanmid.</code> | Color for middle value in heatmap (default gray) |
| <code>plotspecies.</code> | A vector of tip labels to display on the tree (the remainder will be masked, but the corresponding tips will be plotted on the tree) |
| <code>edgetype.</code> | Vector of line type for edges. |
| <code>textsize.</code> | cex value for tip labels (default 0.6) |
| <code>colbarlab.</code> | Label for the color bar legend |
| <code>splist2sym.</code> | A value within species names to display as a symbol |
| <code>dolegend.</code> | Whether to display the heatmap legend. |
| <code>nacol.</code> | Color to display for any edges with length NA |
| <code>figwid.</code> | Adjust x limits of <code>plot.phylo</code> by 1/ <code>figwid</code> . May be related to figure width (requires some optimization) |
| <code>....</code> | further arguments to be passed to ‘plot’ or to ‘plot.phylo’ |

Value

Plots a cladogram of ‘tree’ with branch colors determined by ‘vals’

| | |
|--------------|---|
| treePlotRers | <i>Plot a cladogram with RERs shown as either labels (type="label") or a color heatmap along the branches (type="color") Wraps around returnRersAsTree or treePlotNew, respectively</i> |
|--------------|---|

Description

Plot a cladogram with RERs shown as either labels (type="label") or a color heatmap along the branches (type="color") Wraps around [returnRersAsTree](#) or [treePlotNew](#), respectively

Usage

```
treePlotRers (
  treesObj,
  rermat = NULL,
  index = NULL,
  type = c("label", "color"),
  phenv = NULL,
  figwid = 10,
  ...
)
```

Arguments

| | |
|-----------|---|
| treesObj. | A treesObj created by readTrees |
| rermat. | A residual matrix, output of the <code>getAllResiduals()</code> function |
| index. | A character denoting the name of gene, or a numeric value corresponding to the gene's row index in the residuals matrix |
| type. | Whether to display RERs as branch labels ('label') or a heatmap ('color') |
| phenv. | A phenotype vector returned by tree2Paths or foreground2Paths |
| | Additional parameters to be passed to returnRersAsTree or treePlotNew #' @param figwid. Adjust x limits of plot.phylo by 1/figwid. May be related to figure width (requires some optimization) |

Value

Plots a cladogram of the master tree with RERs displayed as branch labels or colors

Index

calculateCorPermuted, 3
calculatePermulatedPaths_apply,
 3, 4
char2Paths, 4, 7, 16
click_select_foreground_branches,
 5
combinePermData, 6
cor, 3, 7, 8, 16
correlateWithBinaryPhenotype, 6,
 26
correlateWithContinuousPhenotype,
 7, 26

drop.tip, 28

edgeVars, 8
estimatePhangornTree, 9
estimatePhangornTreeAll, 10

fastwilcoxGMTall, 11, 24, 25
fixPseudoroot, 11
foreground2Paths, 12, 27, 31, 40
foreground2Tree, 7, 13, 17
foreground2TreeClades, 14

generatePermulatedBinPhen, 14
generatePermulatedBinPhenSSMBatched,
 15
getAllCor, 4, 16
getAllResiduals, 3, 6, 7, 16, 17, 20, 21
getEnrichPerms, 18
getForegroundInfoClades, 19
getPermsBinary, 20, 26
getPermsContinuous, 6, 18, 21, 24–26
getStat, 22

list.files, 10

makeBinaryPaths, 23
matchAllNodesClades, 23

namePathsWSpecies, 24

optim.pml, 9, 11

permpvalcor, 24
permpvalenrich, 25
permutevec, 25
phyDat, 9, 11
plotPositivesFromPermutations, 26
plotRers, 26
plotTreeHighlightBranches, 27
pml, 9, 11
pruneTree, 28

read.gmt, 29
readTrees, 4, 5, 11–16, 18–21, 23, 29,
 30–37, 40
RERconverge, 30
returnRersAsNewickStrings, 30
returnRersAsTree, 31, 40
returnRersAsTreesAll, 32

simBinPheno, 32
simBinPhenoCC, 33
simBinPhenoSSM, 34
simBinPhenoVec, 34
simpermvec, 35
simulatevec, 36

tree2Paths, 6, 7, 16, 27, 31, 36, 40
tree2PathsClades, 37
treePlotNew, 38, 40
treePlotRers, 40