Package 'MSCquartets'

October 17, 2019

Title Analyzing gene trees through quartets under the multispecies coalescent model

Version 0.5.1

Description A package for analyzing and using quartets displayed on a collection of gene trees, primarily to make inferences about the species tree or network under the multispecies coalescent (MSC) model.

License MIT + file LICENSE

Imports RandomFieldsUtils,
zipfR,
MCMCpack,
graphics,
stats,
utils,
Rdpack

RdMacros Rdpack

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Elizabeth Allman [aut],
Hector Banos [aut],
Jonathan Mitchell [aut],
John Rhodes [aut, cre]

Maintainer John Rhodes <j.rhodes@alaska.edu>

Depends R (>= 3.2.0),
ape (>= 5.0),
phangorn

R topics documented:

`dataGeneTreeSample` *Simulated gene tree dataset*

Description

A text file dataset containing ??? gene trees on ??? taxa simulated under the MSC on the species tree ?????

Format

A text file with ??? metric Newick gene trees on the taxa ????

Details

File is accessed as `system.file("extdata","dataGeneTreeSample",package="MSCquartets")`

Source

Simulated using SimPhy (ref)

---

dataHeliconiusMartin *Heliconius gene tree dataset*

Description

A text file dataset for Heliconius butterflies containing 2909 gene trees on 7 taxa, 3 with 4 individuals sampled each, for a total of 16 leaves per gene tree. This is a subset of the data of Martin et al. (2013)

Format

A text file with 2909 metric Newick gene trees each with 16 leaves labelled: chioneus.553, chioneus.560, chioneus.564, chioneus.565, ethilla.67, hecale.273, melpomeneFG.13435, melpomeneFG.9315, melpomeneFG.9316, melpomeneFG.9317, pardalinus.371, rosina.2071, rosina.531, rosina.533, rosina.546, sergestus.202

Details

File is accessed as `system.file("extdata","dataHeliconiusMartin",package="MSCquartets")`

Source

http://datadryad.org/resource/doi:10.5061/dryad.dk712

References

Martin S, K.K. D, Nadeau N, Salazar C, Walters J, Simpson F, Blaxter M, Manica A, Mallet J, Jiggins C (2013). "Genome-wide evidence for speciation with gene flow in Heliconius butterflies." *Genome Res*, **23**, 1817-1828.

---

dataYeastRokas *Yeast gene tree dataset*

Description

A text file dataset for Yeast containing 106 gene trees on 8 taxa (7 yeasts and 1 outgroup). This is a subset of the data of Rokas et al. (2003).

Format

A text file with 106 topological Newick gene trees on the taxa: Sbay, Scas, Scer, Sklu, Skud, Smik, Spar, and Calb (outgroup).

Details

File is accessed as `system.file("extdata","dataYeastRokas",package="MSCquartets")`

Source

https://wiki.rice.edu/confluence/download/attachments/8898533/yeast.trees?version=1&modificationDate=1360603275797&api=v2

References

Rokas A, Williams B, Carrol S (2003). "Genome-scale approaches to resolving incongruence in molecular phylogenies." *Nature*, **425**, 798–804.

`HolmBonferroni` *Apply Holm-Bonferroni method to adjust for multiple tests*

Description

Apply Holm-Bonferroni method to adjust for multiple tests performed on quartets from a data set of gene trees.

Usage

```
HolmBonferroni(pTable, model, alpha)
```

Arguments

`pTable`	a table of quartets with p-values, as computed by `MultiIndepQuartetTest` or `MultiIndepStarTest`
`model`	one of "T1", "T3", or "star", where pTable contains a column `p_model` of p-values
`alpha`	a critcal value, for rejection of adjusted p-values less than or equal to alpha

Details

When p-values are computed for each quartet using `MultiIndepQuartetTest` or `MultiIndepStarTest`, multiple comparisons are being done for one dataset. The Holm-Bonferroni method adjusts these p-values upward, controlling the familywise error rate. The probability of at least one false discovery (rejection of null hypothesis) is no more than the significance level.

Value

the same table, with rows reordered, and 2 new columns of 1) adjusted p-values, and 2) "Y" or "N" for indicating "reject" or "fail to reject"

See Also

`quartetTreeTestInd`, `quartetStarTestInd`

Examples

```
gtrees=read.tree(file=system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
taxanames=taxonNames(gtrees)
QT=quartetTable(gtrees,taxanames)
RQT=quartetTableResolved(QT)
pTable=quartetTreeTestInd(RQT,"T3")
HBpTable=HolmBonferroni(pTable,"T3",.05)
HBpTable
```

MSCquartets *Multispecies Coalescent Model Quartet Package*

Description

A package for analyzing quartets displayed on gene trees, under the multispecies coalescent (MSC) model.

Details

This package contains routines to analyze a collection of gene trees through the displayed quartets on them.

Recall that a quartet count concordance factor (QCCF) for a set of 4 taxa is the triple of counts of the three possible resolved quartet trees on those taxa across some set of gene trees. The major routines in this package can:

1. Tabulate all QCCFs for a collection of gene trees.
2. Perform hypothesis tests of whether one or more QCCFs are consistent with the MSC model on a species tree (Mitchell et al. 2019).
3. Infer a species tree using the QCCFs via the QDC and WQDC methods (Rhodes 2019; Yourdkhani and Rhodes 2020).
4. Infer a level-1 species network via the NANUQ method (Allman et al. 2019).

As discussed in the cited works, the inference methods for species trees and networks are statistically consistent under the MSC and Network MSC respectively.

Several gene tree data sets, simulated and empirical, are included.

References

Rhodes J (2019). "Topological metrizations of trees, and new quartet methods of tree inference." *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **vol**(num), pp-qq.

Mitchell J, Allman E, Rhodes J (2019). "Hypothesis testing near singularities and boundaries." *Electron. J. Statist.*, **13**(1), 2150-2193.

Allman E, Baños H, Rhodes J (2019). "NANUQ: A method for inferring species networks from gene trees under the coalescent model." *Algorithms for Molecular Biology*, **vol**(num), pp-qq.

Yourdkhani S, Rhodes J (2020). "Inferring species treees from weighted quartets." *unknown journal*, **vol**(num), pp-qq.

NANUQ *Apply NANUQ network inference algorithm to gene tree data*

Description

Apply the NANUQ algorithm of Allman et al. (2019) to infer a hybridization network from a collection of gene trees, under the level-1 network multispecies coalescent (NMSC) model.

Usage

```
NANUQ(genedata, outfile = "NANUQdist", alpha = 0.05, beta = 0.95,
  taxanames = NULL, plot = TRUE)
```

Arguments

genedata	gene tree data that may be supplied in one of 3 ways: 1. as a character string giving the name of a file containing Newick gene trees, 2. as a multiPhylo object containing the gene trees, or 3. as a table of quartets on the gene trees, as produced by a previous call to NANUQ or quartetTableResolved, which has columns only for taxa, quartet counts, and possibly p_T3 and p_star.
outfile	a character string giving an output file name stub, to which will be appended an alpha and beta value and ".nex", for saving the NANUQ distance matrix; if NULL then distance matrix not computed
alpha	a value or vector of significance levels for judging p-values testing a null hypothesis of no hybridization for each quartet; smaller values mean fewer calls of hybridization
beta	a value or vector of significance levels for judging p-values testing a null hypothesis of a star tree for each quartet; smaller values result in fewer calls of a resolved tree; if vectors, alpha and beta must have the same length
taxanames	if genedata is a file or multiphylo object, a subset of taxa on the gene trees, which will be the only ones analyzed, if NULL all taxa on the first gene tree are used; if genedata is a quartet table, this argument is ignored
plot	TRUE produces simplex plots of hypothesis test results, FALSE omits plots

Details

This function

1. counts displayed quartets across gene trees,
2. applies appropriate hypothesis tests to judge quartet CFs as representing putative hybridization, resolved trees, or unresolved (star) trees using alpha and beta as significance levels, and
3. computes the appropriate distance table under the level-1 network quartet distance, writing it to a file.

The distance table file can then be opened in SplitsTree to obtain a circular split system under the Neighbor-Net algorithm, which is then depicted as a splits graph. The splits graph should be interpretted via the theory of Allman et al. (2019) to infer the level-1 species network, or to conclude the data does not arise from the NMSC on such a network.

If alpha and beta are vectors, they must be of same length, k. Then the i-th entries are paired to produce k plots and k output files. This is equivalent to k calls to NANUQ with scalar values of alpha and beta.

A call of NANUQ with genedata given as a table previously output from NANUQ is equivalent to a call of NANUQdist.

If plots are produced, each point represents an empirical quartet concordance factor, color-coded to represent test results

In most instances of NANUQ, an initial call to NANUQ will not give a good analysis, as values of alpha and beta are likely to need some adjustment based on inspecting the data. Saving the returned table from NANUQ will allow for the results of the time-consuming computation of quartet counts and p-values to be saved, for input to further calls of NANUQ with new choices of alpha and beta.

Value

a table of quartets and p-values for judging fit to the MSC on quartet trees (returned invisibly); this table can be used as input to NANUQ or NANUQdist with new choices of alpha and beta, without re-tallying quartets on the gene trees. A distance table to be used as input for SplitsTree is written to a nexus file.

References

Allman E, Baños H, Rhodes J (2019). "NANUQ: A method for inferring species networks from gene trees under the coalescent model." *Algorithms for Molecular Biology*, **vol**(num), pp-qq.

See Also

quartetTable, quartetTableDominant, quartetTreeTestInd, quartetStarTestInd, NANUQdist, quartetTestPlot

Examples

```
pTable=NANUQ(system.file("extdata", "dataYeastRokas",package="MSCquartets"), alpha=.0001, beta=.95)
NANUQ(pTable, alpha=.05, beta=.95)
```

NANUQdist *Compute NANUQ distance and write to file*

Description

Computes the quartet distance tables for NANUQ of Allman et al. (2019), using precomputed p-values for quartets, for each of several levels specified. Distance tables are written to files, in nexus format.

Usage

```
NANUQdist(pTable, outfile = "NANUQdist", alpha, beta, plot = TRUE)
```

Arguments

pTable	a table of quartets and p-values, as computed by NANUQ, or by quartetTreeTestInd and quartetStarTestInd, with columns "p_T3" and "p_star"
outfile	a character string giving an output file name stub, to which will be appended an alpha and beta value and ".nex", for saving distance matrix; if NULL then distance matrix not written to file
alpha	a value or vector of significance levels for judging p-values indicating hybridization on quartet, one for each distance table/output file; smaller values mean fewer calls of hybridization
beta	a value or vector of significance levels for judging p-values indicating star quartet tree, one for each distance/table output file; smaller values result in fewer calls of resolved tree; alpha and beta should be vectors of the same length
plot	TRUE produces simplex plots of hypothesis tests, FALSE omits plots

Details

If plots are produced, each point represents an empirical quartet concordance factor, color-coded to represent test results giving interpretation as network, resolved tree, or star tree.

If `alpha` and `beta` are vectors, they must be of same length, k. Then the i-th entries are paired to produce k plots and k distance tables/output files.

Value

a NANUQ distance table, or a list of such tables if alpha and beta are vectors (returned invisibly)

References

Allman E, Baños H, Rhodes J (2019). "NANUQ: A method for inferring species networks from gene trees under the coalescent model." *Algorithms for Molecular Biology*, **vol**(num), pp-qq.

See Also

NANUQ, quartetTreeTestInd, quartetStarTestInd

Examples

```
pTable=NANUQ(system.file("extdata","dataYeastRokas",package="MSCquartets"), alpha=.0001, beta=.95)
NANUQdist(pTable, alpha=.05, beta=.95)
```

---

nexusDist *Write a distance table to a file in nexus format*

Description

Write a distance table to a file in nexus format

Usage

```
nexusDist(distMatrix, outfilename)
```

Arguments

distMatrix	a square matrix giving a distance table, with rows and columns labeled by taxon names
outfilename	the name of an output file

Value

None

---

powerDivStat *Power divergence statistic of Cressie & Read*

Description

Computes any of the family of power-divergence statistics for multinomial data of Cressie and Read (1984), to compare observed and expected counts. Includes Likelihood Ratio and Chi-squared statistics as special cases.

Usage

```
powerDivStat(obs, expd, lambda)
```

Arguments

obs	observation vector
expd	expected vector
lambda	statistic parameter (e.g., 0=Likelihood Ratio, 1=Chi-squared)

Value

value of statistic

References

Cressie N, Read T (1984). "Multinomial Goodness-Of-Fit Tests." *J. Royal Stat. Soc. B*, **46**(3), 440-464.

Examples

```
obs=c(10,20,30)
expd=c(20,20,20)
powerDivStat(obs,expd,0)
```

pvalHist *Plot histogram of log p-values in table*

Description

Graphical exploration of extreme p-values from quartet hypothesis tests, to aid in choosing critical values for use in NANUQ algorithm. Log base 10 of p-values exceeding some minimum are plotted, to explore gaps in the tail of the distribution.

Usage

```
pvalHist(pTable, model, pmin = 0)
```

Arguments

pTable	a quartet table with p-values such as from quartetTreeTestInd or from quartetStarTestInd
model	one of "T1", "T3", or "star", where pTable contains a column p_model of p-values
pmin	include only p-values above pmin

Examples

```
pTable=NANUQ(system.file("extdata","dataYeastRokas",package="MSCquartets"), alpha=0, beta=.95)
pvalHist(pTable,"T3")
NANUQdist(pTable, alpha=10^-5, beta=.95)
NANUQdist(pTable, alpha=10^-3, beta=.95)
```

QDC *Compute Quartet Distance Consensus tree from gene tree data*

Description

Compute the Quartet Distance Consensus (Rhodes 2019) estimate of an unrooted topological species tree from gene tree data

Usage

```
QDC(genetreedata, taxanames = NULL, omit = FALSE)
```

Arguments

genetreedata	gene tree data either in one of 3 forms: 1. a character string giving the name of a file containing gene trees in Newick, 2. a multiphylo object containing gene trees, or 3. a resolved quartet table, such as produced by quartetTableResolved
taxanames	list of taxa on which to construct tree; can be subset of those on trees; if NULL, uses taxa on first gene tree; this argument is ignored if genetreedata is a resolved quartet table
omit	TRUE ignores unresolved quartets, FALSE treats them as 1/3 of each resolution.

Details

This function is a wrapper which performs the the steps of reading in a collection of gene trees, tallying quartets, computing the quartet distanace between taxa, and building a tree which consistently estimates the unrooted species tree topology under the MSC.

Value

an unrooted topological tree, of type phylo

References

Rhodes J (2019). "Topological metrizations of trees, and new quartet methods of tree inference." *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **vol**(num), pp-qq.

See Also

quartetTable, quartetTableResolved, quartetTableDominant, quartetDist, QDS, WQDC, WQDCrecursive

Examples

```
stree=QDC(system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
```

QDS *Compute Quartet Distance Supertree*

Description

Apply the Quartet Distance Supertree method of Rhodes (2019) to a table specifying a collection of quartets on n taxa.

Usage

```
QDS(dqt, method = fastme.bal)
```

Arguments

dqt	an (n choose 4) x n (or n+1) matrix of form output by quartetTableDominant (Note: n+1th column of dqt is ignored)
method	tree building method function (fastme.bal, nj, etc.)

Details

This function is a wrapper which runs quartetDist and then builds a tree.

Value

An unrooted metric tree of type "phylo". Edge lengths are not in interpretable units.

References

Rhodes J (2019). "Topological metrizations of trees, and new quartet methods of tree inference." *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **vol**(num), pp-qq.

See Also

quartetTableDominant, quartetDist, QDC, WQDS, WQDC, WQDCrecursive

Examples

```
gtrees=read.tree(file=system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
taxanames=taxonNames(gtrees)
QT=quartetTable(gtrees,taxanames[1:5])
RQT=quartetTableResolved(QT)
DQT=quartetTableDominant(RQT)
tree=QDS(DQT)
```

---

quartetDist *Compute quartet distance between taxa*

Description

Compute the Quartet Distance of Rhodes (2019) from a table specifying a collection of quartets on n taxa.

Usage

```
quartetDist(dqt)
```

Arguments

dqt	an (n choose 4) x n (or n+1) matrix of form output by quartetTableDominant (Note: n+1th column of dqt is ignored)

Value

a pairwise distance matrix on n taxa

References

Rhodes J (2019). "Topological metrizations of trees, and new quartet methods of tree inference." *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **vol**(num), pp-qq.

See Also

quartetTableDominant, QDS, QDC, quartetWeightedDist

Examples

```
gtrees=read.tree(file=system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
taxanames=taxonNames(gtrees)
QT=quartetTable(gtrees,taxanames[1:5])
RQT=quartetTableResolved(QT)
DQT=quartetTableDominant(RQT)
Dist=quartetDist(DQT)
tree=NJ(Dist)
```

quartetNetworkDist *Compute network quartet distance between taxa*

Description

Produce network quartet distance table for NANUQ, from a table of quartets and p-values, and specified levels of quartet hypothesis tests. The network quartet distance, which is described more fully by Allman et al. (2019), generalizes the quartet distance of Rhodes (2019).

Usage

```
quartetNetworkDist(pTable, alpha0, beta0)
```

Arguments

pTable	a table of quartets and p-values, as computed by NANUQ, or quartetTreeTestInd and quartetStarTestInd
alpha0	a scalar significance level for judging p_T3 indicating hybridization on quartet; smaller value gives fewer hybridization calls
beta0	a scalar significance level for judging p_star indicating star quartet tree; smaller value gives fewer resolved tree calls

Value

a distance table

References

Allman E, Baños H, Rhodes J (2019). "NANUQ: A method for inferring species networks from gene trees under the coalescent model." *Algorithms for Molecular Biology*, **vol**(num), pp-qq.

Rhodes J (2019). "Topological metrizations of trees, and new quartet methods of tree inference." *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **vol**(num), pp-qq.

See Also

NANUQ, NANUQdist

quartetStarTest *Hypothesis test for quartet counts fitting a star tree under the MSC*

Description

Perform hypothesis test for star tree for a vector of quartet counts to fit expected frequencies of (1/3,1/3,1/3). The test performed is a standard chi squared.

Usage

```
quartetStarTest(obs)
```

Arguments

obs	vector of 3 counts of resolved quartet frequencies

Value

p-value

Examples

```
obs=c(16,72,12)
quartetStarTest(obs)
```

---

quartetStarTestInd *Mutiple independent hypothesis tests for gene quartet counts fitting a star species tree under the MSC*

Description

Perform hypothesis test for star species tree for all quartet counts in an input table, as if the quartets are independent.

Usage

```
quartetStarTestInd(rqt)
```

Arguments

rqt	Table of resolved quartet counts, as produced by quartetTableResolved, or quartetTreeTestInd

Details

This function assumes all quartets are resolved. The test performed is described in quartetStarTest.

Value

The same table as the input rqt with column "p_star" appended, containing p-values for judging fit to MSC on a star tree

See Also

quartetStarTest, quartetTreeTest, quartetTreeTestInd, quartetTableResolved, quartetTestPlot

Examples

```
gtrees=read.tree(file=system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
taxanames=taxonNames(gtrees)
QT=quartetTable(gtrees,taxanames)
RQT=quartetTableResolved(QT)
pTable=quartetStarTestInd(RQT)
quartetTablePrint(pTable)
```

quartetTable *Produce table of counts of quartets displayed on trees*

Description

Compiles table of counts of topological quartets displayed on a collection of trees.

Usage

```
quartetTable(trees, taxonnames = NULL, epsilon = 0, random = 0)
```

Arguments

trees	multiphylo object containing un/rooted metric/topological trees
taxonnames	vector of names of taxa of interest, of length n; if NULL then taken from taxa on trees[[1]]
epsilon	minimum for branch lengths to be treated as non-zero
random	number of random subsets of 4 taxa to consider; if 0, use all n choose 4 subsets

Details

The taxa on the trees may be any set overlapping with taxonnames. Branch lengths of non-negative size less than or equal to epsilon are treated as zero, giving polytomies.

Error if any branch length <0; Warnings if some of taxonnames are missing on some trees, or if some 4-taxon set is on no trees.

If random>0, then for efficiency it should be much smaller then the number of possible 4 taxon subsets.

Value

A (n choose 4)x(n+4) matrix (or (random)x(n+4) matrix) encoding 4 taxon subsets of taxonnames and counts of each of the quartets 12|34, 13|24, 14|23, 1234 across the trees. Columns are labeled by taxa names and quartet names("12|34", etc.). 1s and 0s in taxon columns indicate the taxa in quartet. Quartet 12|34 means first and second indicated taxa form cherry, 13|24 means first and third form cherry, 14|23 means first and fourth form cherry, and 1234 means the quartet is unresolved

See Also

quartetTableResolved, quartetTableDominant

Examples

```
gtrees=read.tree(file=system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
taxonnames=taxonNames(gtrees)
QT=quartetTable(gtrees,taxonnames[1:5])
RQT=quartetTableResolved(QT)
DQT=quartetTableDominant(RQT)
```

quartetTableCollapse *Form a smaller resolved quartet table by lumping some taxa into a composite taxon*

Description

Form a smaller resolved quartet table by lumping some taxa into a composite taxon

Usage

```
quartetTableCollapse(rqt, taxaA, taxaB)
```

Arguments

rqt	a resolved quartet table, as from quartetTableResolved
taxaA	a vector of taxon names in rqt to be included in new table
taxaB	a vector of taxon names in rqt to form new composite taxon in new table

Details

This function is needed for the recursive calls in WQDSrec. It should only be applied to a resolved quartet table which includes counts for all possible quartets on the taxa (though counts can be zero).

Value

a resolved quartet table with length(taxaA)+1 taxa. The composite taxon is named as the concatenation of the sorted names in taxaB

See Also

WQDCrecursive

quartetTableDominant *Produce table of dominant quartets, with estimates of internal edge lengths*

Description

Converts table of counts of resolved quartets on n taxa to show only dominant one, with maximum likelihood estimate of internal edge weight under the multispecies coalescent model.

Usage

```
quartetTableDominant(rqt, bigweights = "infinite")
```

Arguments

rqt	array as produced by quartetTableResolved of size (n choose 4)x(n+3);
bigweights	"infinite" or "finite", determines if the weight (internal edge length) of a quartet for which only one topology appears is given as "Inf" or a finite, but large, numerical value

Value

An (n choose 4)x(n+1) array with dominant quartet topology encoded by 1,1,-1,-1 in taxon columns, with signs indicating cherries. Column "weight" contains ML estimate under MSC model of quartets central edge length, in coalescent units.

See Also

quartetTable, quartetTableResolved

Examples

```
gtrees=read.tree(file=system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
taxonnames=taxonNames(gtrees)
QT=quartetTable(gtrees, taxonnames[1:5])
RQT=quartetTableResolved(QT)
DQT=quartetTableDominant(RQT)
```

---

quartetTablePrint *Print a quartet table with nice formatting*

Description

Print a quartet table so that taxa in each quartet are shown by name.

Usage

```
quartetTablePrint(qt)
```

Arguments

qt	a table such as returned by quartetTable, quartetTableResolved, or quartetTableDominant, possibly with extra columns added by other functions

Examples

```
gtrees=read.tree(file=system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
taxonnames=taxonNames(gtrees)
QT=quartetTable(gtrees,taxonnames)
quartetTablePrint(QT)
RQT=quartetTableResolved(QT)
quartetTablePrint(RQT)
pTable=quartetTreeTestInd(RQT,"T3")
quartetTablePrint(pTable)
DQT=quartetTableDominant(RQT)
quartetTablePrint(DQT)
```

quartetTableResolved *Modify quartet table to only show resolved quartets*

Description

Converts table of all quartet counts, including unresolved ones, by either dropping unresolved ones, or distributing them uniformly among the three resolved counts.

Usage

```
quartetTableResolved(qt, omit = FALSE)
```

Arguments

qt	table as produced by quartetTable of size (n choose 4)x(n+4)
omit	TRUE deletes unresolved quartets column; FALSE redistributes unresolved counts as (1/3,1/3,1/3) to resolved counts

Value

A table of size (n choose 4)x(n+3), similar to quartetTable

See Also

quartetTable, quartetTableDominant

Examples

```
gtrees=read.tree(file=system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
taxonnames=taxonNames(gtrees)
QT=quartetTable(gtrees,taxonnames[1:5])
RQT=quartetTableResolved(QT)
DQT=quartetTableDominant(RQT)
```

quartetTestPlot *Produce simplex plot of results of quartet hypothesis test results*

Description

Plot a 2-d probability simplex, with points for all quartet count vectors. Color of point indicates rejection or failure to reject of tests, at specified level.

Usage

```
quartetTestPlot(pTable, test, alpha = 0.05, beta = 1)
```

Arguments

pTable	table of quartets and p-values, as produced by quartetTreeTestInd, quartetStarTestInd, or NANUQ
test	model to use, for tree null hypothesis; options are "T1", "T3"
alpha	significance level for tree test with null hypothesis given by test
beta	significance level for test with null hypothesis star tree; test results plotted only if beta<1

Details

This function must be supplied with a table of quartets and p-values. The plot may show results of either the T1 or T3 test, with or without a star tree test (dependning on whether a p_star column is in the table). The p-values must be supplied by previous calls to quartetTreeTestInd (for T1 or T3 p-values) and quartetStarTestInd (for star tree). The NANUQ and NANUQdist functions include calls for the T3 and star tests

Value

None

See Also

quartetTreeTestInd, quartetStarTestInd, NANUQ,NANUQdist

Examples

```
gtrees=read.tree(file=system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
taxanames=taxonNames(gtrees)
QT=quartetTable(gtrees,taxanames)
RQT=quartetTableResolved(QT)
stree="((a,b),(c,(d,e)));"
pTable=quartetTreeTestInd(RQT,"T1",speciestree=stree)
pTable=quartetStarTestInd(pTable)
quartetTestPlot(pTable, "T1", alpha=.05, beta=.95)
```

---

quartetTreeErrorProb *Bayesian posterior probability of error in 4-taxon unrooted species tree topology estimate*

Description

Computes Bayesian posterior probability that the ML estimate of 4-taxon species tree topology from gene quartet topology counts is incorrect, under the assumption that the counts arise from some species tree.

Usage

```
quartetTreeErrorProb(obs, model = "T3")
```

Arguments

obs	vector of counts for 3 topologies
model	"T3" or "T1", for the models of Mitchell et al. (2019) describing an unspecified species tree topology ("T3"), or the topology corresponding to the first entry of obs ("T1")

Details

The Jeffreys prior is used for internal branch length, along with the uniform prior on the resolved topology.

Value

posterior probability

References

Mitchell J, Allman E, Rhodes J (2019). "Hypothesis testing near singularities and boundaries." *Electron. J. Statist.*, **13**(1), 2150-2193.

Examples

```
obs <- c(28,32,30)
quartetTreeErrorProb(obs,model="T1")
quartetTreeErrorProb(obs,model="T3")
```

---

quartetTreeTest *Hypothesis test for quartet counts fitting a tree under the MSC*

Description

Test the hypothesis H_0= T1 or T3 model of Mitchell et al. (2019), vs. H_1 = everything else. T1 is for a specfic tree topology, and T3 for any tree topology.

Usage

```
quartetTreeTest(obs, model = "T3", lambda = 0,
  smallcounts = "approximate", bootstraps = 10^4, method = "MLest")
```

Arguments

obs	vector of 3 counts of resolved quartet frequncies,
model	"T1" or "T3", for the models of Mitchell et al. (2019),
lambda	parameter for power-divergence statistic (e.g., 0 for likelihood ratio statistic, 1 for Chi-squared statistic)
smallcounts	"bootstrap" or "approximate", method of obtaining p-value when some counts are small
bootstraps	number of samples for bootstrapping
method	"MLtest","conservative", or "bootstrap"

Details

This function implements two of the versions of the test given by Mitchell et al. (2019) as well as parametric boostrapping, with other procedures for when some counts are small. Due to the singularities and boundaries of the models, when the topology and/or the internal quartet branch length is not specified by the null hypothesis these are more accurate tests than, say, a chi-squared with one degree of freedom which assumes no model boundary or singularity near the data.

If `method="MLtest"`, this uses the test by that name described in Section 7 of Mitchell et al. (2019). For both models the test is slightly anticonservative over a small range of true internal edges of the quartet species tree. Although the test generally performs well in practice, it lacks a uniform asymptotic guarantee over the full parameter space for either T1 or T3.

If `method="conservative"`, a conservative test described by Mitchell et al. (2019) is used. For model T3 this uses the chi^2_1 distribution (the "least favorable" approach), while for model T1 it uses the Minimum Adjusted Bonferroni, based on precomputed values from simulations. These conservative tests are asymptotically guaranteed to reject the null hypothesis at most at a specified level, but at the expense of increased type II errors.

If `method="bootstrap"`, then parametric bootsrapping is done, based on parameter estimates of the quartet topology and internal edge length. The bootstrap sample size is given by the `bootstrap` argument.

When some expected topology counts are small, the methods "MLest" and "conservative" are not appropriate. The argument `smallcounts` determines whether bootstrapping or an approximate method that uses precomputed p-values is used. These both involve estimates of the quartet topology and internal edge length.

The returned p-value should be taken with caution when there is a small sample size, e.g. less than 30 gene trees.

For model T1, the first entry of `obs` is treated as the count of gene quartets concordant with the species tree.

The returned value of `t` is a consistent estimator, but not the MLE, of the internal edge length in coalescent units. Although consistent, the MLE for t is biased. Our consistent estimator is still biased, but with less bias than the MLE. See Mitchell et al. (2019) for more discussion on dealing with the bias of parameter estimates in the presence of boundaries and/or singularities of parameter spaces.

Value

`(p-value, t)` where `t` is a consistent estimator of the internal edge length in coalescent units, possibly `Inf`.

References

Mitchell J, Allman E, Rhodes J (2019). "Hypothesis testing near singularities and boundaries." *Electron. J. Statist.*, **13**(1), 2150-2193.

See Also

`quartetTreeTestInd`

Examples

```
obs=c(17,72,11)
quartetTreeTest(obs,"T3")
quartetTreeTest(obs,"T1")
```

`quartetTreeTestInd` *Mutiple independent hypothesis tests for quartet counts fitting a species tree under the MSC*

Description

Perform a tree hypothesis test for all quartet counts in an input table, as if the quartets are independent.

Usage

```
quartetTreeTestInd(rqt, model = "T3", lambda = 0,
  smallcounts = "approximate", bootstraps = 10^4, method = "MLest",
  speciestree = NULL)
```

Arguments

`rqt`	table of resolved quartet counts, as produced by `quartetTableResolved`, or `quartetStarTestInd`
`model`	"T1" for a specific species tree topology, or "T3" for any species tree topology, with these models explained more fully by Mitchell et al. (2019)
`lambda`	power divergence statistic parameter (e.g., 0 for likelihood ratio statistic, 1 for Chi-squared statistic)
`smallcounts`	"bootstrap" or "approximate", method of obtaining p-value when some counts are small
`bootstraps`	number of samples for bootstrapping
`method`	"MLest", "conservative", or "bootstrap"
`speciestree`	species tree, in Newick as text, to determine quartet for T1 test; required for `model="T1"`, ignored for `model="T3"`

Details

This function assumes all quartets are resolved. The test performed is described in `QuartetTreeTest`.

Value

If `model="T3"`, a copy of `rqt` with a new column "p_T3" appended with p-values for each quartet; If `model="T1"`, a copy of `rqt` with 2 columns appended: "p_T1" with p-values, and "qindex" giving index of quartet consistent with specified species tree, i.e., 1 if ab|cd on species tree, 2 if ac|bd, 3 if ad|bc

References

Mitchell J, Allman E, Rhodes J (2019). "Hypothesis testing near singularities and boundaries." *Electron. J. Statist.*, **13**(1), 2150-2193.

See Also

`quartetTreeTest`, `quartetTestPlot`, `quartetStarTestInd`, `quartetTableResolved`

Examples

```
gtrees=read.tree(file=system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
taxanames=taxonNames(gtrees)
QT=quartetTable(gtrees,taxanames)
RQT=quartetTableResolved(QT)
pTable3=quartetTreeTestInd(RQT,"T3")
quartetTablePrint(pTable3)
stree="((a,b),(c,(d,e)));"
pTable1=quartetTreeTestInd(RQT,"T1",speciestree=stree)
quartetTablePrint(pTable1)
```

---

quartetWeightedDist *Compute the Weighted Quartet Distance between taxa*

Description

Compute the Weighted Quartet Distance between taxa of Yourdkhani and Rhodes (2020) from a table specifying a collection of quartets on n taxa and the quartets' internal branch lengths.

Usage

```
quartetWeightedDist(dqt)
```

Arguments

dqt	an (n choose 4) x n+1) matrix of form output by quartetTableDominant

Value

A pairwise distance matrix on n taxa

References

Yourdkhani S, Rhodes J (2020). "Inferring species treees from weighted quartets." *unknown journal*, **vol**(num), pp-qq.

See Also

quartetTableDominant, WQDSAdjustLengths, WQDS, WQDC, WQDCrecursive, quartetWeightedDist

Examples

```
gtrees=read.tree(file=system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
taxanames=taxonNames(gtrees)
QT=quartetTable(gtrees,taxanames[1:5])
RQT=quartetTableResolved(QT)
DQT=quartetTableDominant(RQT,bigweights="finite")
D=quartetWeightedDist(DQT)
tree=NJ(D)
stree=WQDSAdjustLengths(tree)
```

simplexCoords *Convert 3-d coordinates to 2-d simplex coordinates*

Description

Convert 3-d coordinates to 2-d simplex coordinates

Usage

```
simplexCoords(v)
```

Arguments

v	vector of 3 non-negative numbers, not summing to 0

Value

2-d coordinates to plot normalized point in simplex

See Also

simplexLabels, simplexPoint, simplexPrepare, simplexSegment, simplexText

Examples

```
simplexCoords(c(15,65,20))
```

simplexLabels *Label vertices of 2-d simplex*

Description

Label vertices of 2-d simplex

Usage

```
simplexLabels(top = "", left = "", right = "")
```

Arguments

top	label for top
left	label for left bottom
right	label for right bottom

Value

None

See Also

simplexPoint, simplexPrepare, simplexSegment, simplexText, simplexCoords

Examples

```
simplexPrepare("T3","Example Plot")
simplexLabels("ab|cd","ac|bd","ad|bc")
```

---

simplexPoint *Plot point in 2-d simplex*

Description

Normalizes a point given in 3-d non-normalized coordinates, then plots it in the 2-d simplex

Usage

```
simplexPoint(v, ...)
```

Arguments

v	a 3-d point in non-negative orthant, coords not summing to 0
...	other options to pass to graphics::points function

Value

None

See Also

simplexLabels, simplexPrepare, simplexSegment, simplexText, simplexCoords

Examples

```
simplexPrepare("T3","Example Plot")
simplexPoint(c(15,65,20),pch=3,col="blue")
```

simplexPrepare *Draw 2-d probability simplex, with model lines for T3 or T1 model*

Description

Outline the 2-d simplex, and draw the T1 or T3 model points for quartet frequencies. The models are described more fully by Mitchell et al. (2019).

Usage

```
simplexPrepare(model = "T3", maintitle = NULL, titletext = NULL)
```

Arguments

model	"T1" or "T3", for 1-tree or 3-tree model
maintitle	main title for plot
titletext	additional text for title

Value

None

References

Mitchell J, Allman E, Rhodes J (2019). "Hypothesis testing near singularities and boundaries." *Electron. J. Statist.*, **13**(1), 2150-2193.

See Also

simplexLabels, simplexPoint, simplexSegment, simplexText, simplexCoords

Examples

```
simplexPrepare("T3",maintitle="Main title",titletext="further text")
```

simplexSegment *Plot line segment in 2-d simplex*

Description

Normalizes two points in 3-d, and draws segment between them in 2-d simplex

Usage

```
simplexSegment(v, w, ...)
```

Arguments

v, w	3-d endpoints of segment in non-negative orthant, coords not summing to 0
...	other options to pass to graphics::segments function

Value

None

See Also

simplexLabels, simplexPoint, simplexPrepare, simplexText, simplexCoords

Examples

```
simplexPrepare("T3","Example Plot")
simplexSegment(c(15,65,20),c(15,70, 15),col="green")
```

---

simplexText *Add text at a point in 2-d simplex*

Description

Add text at a point in 2-d simplex

Usage

```
simplexText(v, label = "", ...)
```

Arguments

v	a 3-d point in non-negative orthant, coords not summing to 0
label	text
...	other options to pass to graphics text function

Value

None

See Also

simplexLabels, simplexPoint, simplexPrepare, simplexSegment, simplexCoords

Examples

```
simplexPrepare("T3","Example Plot")
simplexText(c(15,65,20),"tree ac|bd")
```

T1density — *Probability density function for Model T1 of Mitchell et al. (2019), Proposition 5.2*

Description

Probability density function for Model T1 of Mitchell et al. (2019), Proposition 5.2

Usage

```
T1density(x, mu0)
```

Arguments

x	statistic value (e.g., likelihood ratio stat, or other power divergence stat)
mu0	parameter

Value

value of density function

References

Mitchell J, Allman E, Rhodes J (2019). "Hypothesis testing near singularities and boundaries." *Electron. J. Statist.*, **13**(1), 2150-2193.

See Also

T3density

T3density — *Probability density function for Model T3 of Mitchell et al. (2019), Proposition 4.2*

Description

Probability density function for Model T3 of Mitchell et al. (2019), Proposition 4.2

Usage

```
T3density(x, mu0, alpha0, beta0)
```

Arguments

x	statistic value (e.g., LR stat, or other power divergence stat)
mu0	parameter
alpha0	parameter
beta0	parameter

Value

value of density function

References

Mitchell J, Allman E, Rhodes J (2019). "Hypothesis testing near singularities and boundaries." *Electron. J. Statist.*, **13**(1), 2150-2193.

See Also

T1density

---

taxonNames *Get all taxon names from a collection of trees*

Description

Create a vector of all taxa appearing on a collection of trees, with no repeats.

Usage

```
taxonNames(trees)
```

Arguments

trees	a multiphylo object containing a collection of trees

Value

a vector of unique names of taxa appearing on the trees

Examples

```
gtrees=read.tree(file=system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
taxonnames=taxonNames(gtrees)
```

WQDC *Compute Weighted Quartet Distance Consensus tree from gene tree data*

Description

Compute the Weighted Quartet Distance Consensus (Yourdkhani and Rhodes 2020) estimate of a species tree from gene tree data. This is a consistent estimator of the unrooted species tree topology and all internal branch lengths.

Usage

```
WQDC(genetreedata, taxanames = NULL, omit = FALSE, terminal = 1)
```

Arguments

genetreedata	gene tree data in one of the 3 forms 1. a character string giving the name of a file containing gene trees in Newick 2. a multiphylo object containing gene trees, 3. a resolved quartet table, as produced by quartetTableResolved
taxanames	list of taxa to construct tree on; may be subset of those on gene trees; ignored if genetreedata given as resolved quartet table
omit	TRUE leaves out unresolved quartets, FALSE treats them as 1/3 of each resolution; ignored if genetreedata given as resolved quartet table
terminal	non-negative branch length to supply for terminal branches, whose length cannot be inferred by WQDC

Details

This function is a wrapper which performs the the steps of reading in a collection of gene trees, tallying quartets, estimating quartet internal branch lengths, computing the weighted quartet distanace between taxa, building a tree, and adjusting edge lengths, to give a consistent estimate of the metric species tree (in coalescent units) under the multispecies coalescent model.

Value

an unrooted metric tree of type "phylo"

References

Yourdkhani S, Rhodes J (2020). "Inferring species treees from weighted quartets." *unknown journal*, **vol**(num), pp-qq.

See Also

quartetTable, quartetTableResolved, quartetTableDominant, quartetWeightedDist, WQDCrecursive WQDS, QDC

Examples

```
stree=WQDC(system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
```

WQDCrecursive *Compute the Recursive Weighted Quartet Distance Consensus tree from gene tree data*

Description

Infer a metric species tree from counts of quartets displayed on a collection of gene trees, as described by Yourdkhani and Rhodes (2020).

Usage

```
WQDCrecursive(rqt, terminal = 1)
```

Arguments

rqt	a resolved quartet table as produced by quartetTableResolved
terminal	non-negative branch length to supply for terminal branches, whose length cannot be inferred by WQDCrecursive

Details

The algorithm counts quarets displayed on the gene trees, builds a tree using WQDS, determines the split corrsponding to the longest edge in that tree, and then recursively builds trees on the taxa in each split set together with a 'composite taxon' formed by all taxa in the other split set. This approach is slower the WQDS, but increases topological accuracy.

This function must be called with its argument a resolved quartet table (not a dominant quartet table) of size (n choose 4)x(n+3). Its recursive nature requires building smaller resolved quartet tables on split sets with an additional composite taxon.

Value

an unrooted metric tree, of type "phylo"

References

Yourdkhani S, Rhodes J (2020). "Inferring species treees from weighted quartets." *unknown journal*, **vol**(num), pp-qq.

See Also

quartetTableResolved,quartetTable, QDC, QDS, quartetTableCollapse

Examples

```
gtrees=read.tree(file=system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
taxanames=taxonNames(gtrees)
QT=quartetTable(gtrees,taxanames[1:5])
RQT=quartetTableResolved(QT)
stree=WQDCrecursive(RQT)
plot(stree)
stree
```

WQDS *Compute the Weighted Quartet Distance Supertree*

Description

Apply the Weighted Quartet Distance Supertree method of Yourdkhani and Rhodes (2020) to a collection of quartets together with internal quartet branch lengths, specified by a table.

Usage

```
WQDS(dqt, method = fastme.bal)
```

Arguments

dqt	an (n choose 4) x n+1) matrix of form output by quartetTableDominant
method	tree building method function (fastme.bal, NJ, etc.)

Details

This function is a wrapper which runs quartetWeightedDist, builds a tree, and then adjusts edge lengths with WQDSAdjustLengths.

Value

an unrooted metric tree, of type "phylo"

References

Yourdkhani S, Rhodes J (2020). "Inferring species treees from weighted quartets." *unknown journal*, **vol**(num), pp-qq.

See Also

quartetTableDominant, quartetWeightedDist, WQDSAdjustLengths, WQDC, WQDCrecursive, QDS

Examples

```
gtrees=read.tree(file=system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
taxanames=taxonNames(gtrees)
QT=quartetTable(gtrees,taxanames[1:5])
RQT=quartetTableResolved(QT)
DQT=quartetTableDominant(RQT,bigweights= "finite")
tree=WQDS(DQT)
```

WQDSAdjustLengths *Adjust edge lengths on tree built from Weighted Quartet distance to estimate metric tree*

Description

Modify edge lengths of tree built from distance table produced by quartetWeightedDist, to remove scaling factors related to the size of the split associated to the edge.

Usage

```
WQDSAdjustLengths(tree)
```

Arguments

tree	an unrooted metric tree, of type "phylo"

Details

As explained by Yourdkhani and Rhodes (2020), a metric tree produced from the weighted quartet distance has edge lengths inflated by a factor dependent on the associated split size. Removing these factors yields a consistent estimate of the metric species tree displaying the weighted quartets, if such a tree exists.

This function should not be used on trees output from `WQDS`, `WQDC`, or `WQDCrecursive`, as their edges are already adjusted. It can be used on trees built from the distance computed by `quartetWeightedDist`.

Value

an unrooted metric tree, of type "phylo"

References

Yourdkhani S, Rhodes J (2020). "Inferring species treees from weighted quartets." *unknown journal*, **vol**(num), pp-qq.

See Also

`WQDS`, `WQDC`

Examples

```
gtrees=read.tree(file=system.file("extdata","dataGeneTreeSample",package="MSCquartets"))
taxanames=taxonNames(gtrees)
QT=quartetTable(gtrees,taxanames[1:5])
RQT=quartetTableResolved(QT)
DQT=quartetTableDominant(RQT,bigweights="finite")
D=quartetWeightedDist(DQT)
tree=NJ(D)
stree=WQDSAdjustLengths(tree)
```

Index