

# Package ‘MSCquartets’

October 10, 2019

**Title** Analyzing gene trees through quartets under the multispecies coalescent model

**Version** 0.4.8

**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

**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:

AllTaxonNames . . . . .	2
CollapseQuartetTable . . . . .	3
DominantQuartetTable . . . . .	3
HolmBonferroni . . . . .	4
log10Hist . . . . .	5
MSCquartets . . . . .	6
MultiIndepQuartetStarTest . . . . .	7
MultiIndepQuartetTreeTest . . . . .	8
NANUQ . . . . .	9

NANUQdist . . . . .	11
NexusDist . . . . .	12
PlotQuartetTest . . . . .	12
power.div.stat . . . . .	13
QDC . . . . .	14
Qdist . . . . .	15
QDS . . . . .	15
QNdist . . . . .	16
quartet.table.print . . . . .	17
QuartetStarTest . . . . .	17
QuartetTable . . . . .	18
QuartetTreeErrorProb . . . . .	19
QuartetTreeTest . . . . .	20
ResolvedQuartetTable . . . . .	21
SLabels . . . . .	22
SPoint . . . . .	23
SPrepare . . . . .	23
SSegment . . . . .	24
SText . . . . .	25
StoP . . . . .	25
T1density . . . . .	26
T3density . . . . .	26
WQDC . . . . .	27
WQDCrec . . . . .	28
WQdist . . . . .	29
WQDS . . . . .	29
WQDSAdjustLengths . . . . .	30
<b>Index</b>	<b>32</b>

---

AllTaxonNames	<i>Get all taxon names from a collection of trees</i>
---------------	---

---

## Description

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

## Usage

```
AllTaxonNames(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("genetreeSample",package="MSCquartets"))
taxonnames=AllTaxonNames(gtrees)
```

---

CollapseQuartetTable	<i>Form a smaller resolved quartet table by lumping some taxa into a composite taxon</i>
----------------------	--

---

**Description**

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

**Usage**

```
CollapseQuartetTable(rqt, taxaA, taxaB)
```

**Arguments**

rqt	a resolved quartet table, as from ResolvedQuartetTable
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  $\text{length}(\text{taxaA})+1$  taxa. The composite taxon is named as the concatenation of the sorted names in taxaB

**See Also**

[WQDCrec](#)

---

DominantQuartetTable	<i>Produce table of dominant quartets, with estimates of internal edge lengths</i>
----------------------	--

---

**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**

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

Arguments

rqt	array as produced by ResolvedQuartetTable of size $(n \text{ choose } 4) \times (n+3)$ ;
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  $(n \text{ choose } 4) \times (n+1)$  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](#), [ResolvedQuartetTable](#)

Examples

```
gtrees=read.tree(file=system.file("genetreeSample",package="MSCquartets"))
taxonnames=AllTaxonNames(gtrees)
QT=QuartetTable(gtrees, taxonnames[1:5])
RQT=ResolvedQuartetTable(QT)
DQT=DominantQuartetTable(RQT)
```

---

HolmBonferroni	<i>Apply Holm-Bonferroni method to adjust for multiple tests</i>
----------------	--

---

Description

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

Usage

```
HolmBonferroni(pTable, columnname, alpha)
```

Arguments

pTable	a table of quartets with p-values, as computed by MultiIndepQuartetTest or MultiIndepStarTest
columnname	a name of a column of pTable containing p-values, such as "p_T3", "p_T1", "p_star"
alpha	a critical value, for rejection of adjusted p-values below 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**

[MultiIndepQuartetTreeTest](#), [MultiIndepQuartetStarTest](#)

**Examples**

```
gtrees=read.tree(file=system.file("genetreeSample",package="MSCquartets"))
taxanames=AllTaxonNames(gtrees)
QT=QuartetTable(gtrees,taxanames)
RQT=ResolvedQuartetTable(QT)
pTable=MultiIndepQuartetTreeTest(RQT,"T3")
HBpTable=HolmBonferroni(pTable,"p_T3",.05)
HBpTable
```

---

log10Hist

---

*Plot histogram of log base 10 of 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

**Usage**

```
log10Hist(pTable, pcol, pmin = 0)
```

**Arguments**

pTable	a quartet table with p-values such as from <a href="#">MultiIndepQuartetTreeTest</a> or from <a href="#">MultiIndepQuartetStarTest</a>
pcol	name of column of pTable containing p-values, "p_T1", "p_T3", or "p_star"
pmin	include only p-values above pmin

**Examples**

```
pTable=NANUQ(system.file("yeastRokas",package="MSCquartets"), alpha=0, beta=.95)
log10Hist(pTable,"p_T3")
NANUQdist(pTable, alpha=10^-5, beta=.95)
NANUQdist(pTable, alpha=10^-3, beta=.95)
```

---

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, as in [MAR19]
3. Infer a species tree using the QCCFs via the methods of [R19] and [YR19]
4. Infer a level-1 species network via the NANUQ method of [ABR19]

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

Two data sets are included as text files. The file "genetreeSample" contains a sample of 10000 gene trees sampled under the MSC from the species tree  $((a,b),(c,(d,e)))$ ; . The file "yeastRokas" contains 106 gene trees for a subset of 7 yeast and one outgroup taxa, subtrees of the [Rokas et al., 2003] data set.

## References

- [R19] *Topological metrizations of trees and new quartet methods of tree inference*, J.A. Rhodes, IEEE/ACM Trans. Comput. Biol. Bioinf., early access, 2019
- [MAR19] *Hypothesis testing near singularities and boundaries*, J.D. Mitchell, E.S. Allman, J.A. Rhodes, Electron. J. Statist., 13 (1) 2019
- [YR19] *Metrizing trees with weighted quartets, and species tree inference*, S. Yourdkhani, J.A. Rhodes, ??draft??, 2019
- [ABR19] *NANUQ: A method for inferring species networks from gene trees under the coalescent model*, E.S. Allman, H. Banos, J.A. Rhodes, submitted, 2019

---

**MultiIndepQuartetStarTest***Multiple 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

```
MultiIndepQuartetStarTest(rqt)
```

## Arguments

rqt	Table of resolved quartet counts, as produced by ResolvedQuartetTable, or MultiIndepQuartetTreeTest
-----	---

## 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](#), [MultiIndepQuartetTreeTest](#), [ResolvedQuartetTable](#), [PlotQuartetTest](#)

## Examples

```
gtrees=read.tree(file=system.file("genetreeSample",package="MSCquartets"))
taxanames=AllTaxonNames(gtrees)
QT=QuartetTable(gtrees,taxanames)
RQT=ResolvedQuartetTable(QT)
pTable=MultiIndepQuartetStarTest(RQT)
quartet.table.print(pTable)
```

---

MultiIndepQuartetTreeTest

*Multiple 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

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

## Arguments

rqt	table of resolved quartet counts, as produced by ResolvedQuartetTable, or MultiIndepQuartetStarTest
model	"T1" for a specific species tree topology, or "T3" for any species tree topology, with these models explained more fully in[MAR19]
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 QuartetTree Test.

## 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 abcd on species tree, 2 if acldb, 3 if adlbc

## See Also

[QuartetTreeTest](#), [PlotQuartetTest](#), [MultiIndepQuartetStarTest](#), [ResolvedQuartetTable](#)



## Examples

```
gtrees=read.tree(file=system.file("genetreeSample",package="MSCquartets"))
taxanames=AllTaxonNames(gtrees)
QT=QuartetTable(gtrees,taxanames)
RQT=ResolvedQuartetTable(QT)
pTable3=MultiIndepQuartetTreeTest(RQT,"T3")
quartet.table.print(pTable3)
stree="((a,b),(c,(d,e)));";
pTable1=MultiIndepQuartetTreeTest(RQT,"T1",speciestree=stree)
quartet.table.print(pTable1)
```

---

NANUQ

---

*Apply NANUQ network inference algorithm to gene tree data*


---

## Description

Apply the NANUQ algorithm of [ABR19] 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: <ol style="list-style-type: none"> <li>1. as a character string giving the name of a file containing Newick gene trees,</li> <li>2. as a multiPhylo object containing the gene trees, or</li> <li>3. as a table of quartets on the gene trees, as produced by a previous call to NANUQ or ResolvedQuartetTable, which has columns only for taxa, quartet counts, and possibly p_T3 and p_star.</li> </ol>
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 interpreted via the theory in [ABR19] 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.

## See Also

[QuartetTable](#), [DominantQuartetTable](#), [MultiIndepQuartetTreeTest](#), [MultiIndepQuartetStarTest](#), [NANUQdist](#), [PlotQuartetTest](#)

## Examples

```
pTable=NANUQ(system.file("yeastRokas",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 [ABR19], 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 MultiIndepQuartetTreeTest and MultiIndepQuartetStarTest, 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)

## See Also

[NANUQ](#), [MultiIndepQuartetTreeTest](#), [MultiIndepQuartetStarTest](#)

## Examples

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

---

NexusDist	<i>Write a distance table to a file in nexus format</i>
-----------	---

---

**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

---

PlotQuartetTest	<i>Produce simplex plot of results of quartet hypothesis test results</i>
-----------------	---

---

**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**

```
PlotQuartetTest(pTable, test, alpha = 0, beta = 1)
```

**Arguments**

pTable	table of quartets and p-values, as produced by MultiIndepQuartetTreeTest, MultiIndepQuartetStarTest, 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 MultiIndepQuartetTreeTest (for T1 or T3 p-values) and MultiIndepQuartetStarTest (for star tree). The NANUQ and NANUQdist functions include calls for the T3 and star tests

**Value**

None

**See Also**[MultiIndepQuartetTreeTest](#), [MultiIndepQuartetStarTest](#), [NANUQ](#), [NANUQdist](#)**Examples**

```
gtrees=read.tree(file=system.file("genetreeSample",package="MSCquartets"))
taxanames=AllTaxonNames(gtrees)
QT=QuartetTable(gtrees,taxanames)
RQT=ResolvedQuartetTable(QT)
stree="((a,b),(c,(d,e))); "
pTable=MultiIndepQuartetTreeTest(RQT,"T1",speciestree=stree)
pTable=MultiIndepQuartetStarTest(pTable)
PlotQuartetTest(pTable, "T1", alpha=.05, beta=.95)
```

power.div.stat

*Power divergence statistic of Cressie & Read***Description**

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

**Usage**

```
power.div.stat(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

**Examples**

```
obs=c(10,20,30)
expd=c(20,20,20)
power.div.stat(obs,expd,0)
```

QDC

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

Compute the Quartet Distance Consensus [R19] estimate of an unrooted topological species tree from gene tree data

**Usage**

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

**Arguments**

<code>genetreedata</code>	gene tree data either in one of 3 forms: <ol style="list-style-type: none"> <li>1. a character string giving the name of a file containing gene trees in Newick,</li> <li>2. a <code>multiphylo</code> object containing gene trees, or</li> <li>3. a resolved quartet table, such as produced by <code>ResolvedQuartetTable</code></li> </ol>
<code>taxanames</code>	list of taxa on which to construct tree; can be subset of those on trees; if <code>NULL</code> , uses taxa on first gene tree; this argument is ignored if <code>genetreedata</code> is a resolved quartet table
<code>omit</code>	<code>TRUE</code> ignores unresolved quartets, <code>FALSE</code> 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 distance 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`

**See Also**

[QuartetTable](#), [ResolvedQuartetTable](#), [DominantQuartetTable](#), [Qdist](#), [QDS](#), [WQDC](#), [WQDCrec](#)

**Examples**

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

---

Qdist	<i>Compute quartet distance between taxa</i>
-------	--

---

**Description**

Compute the Quartet Distance of [R19] from a table specifying a collection of quartets on n taxa.

**Usage**

```
Qdist(dqt)
```

**Arguments**

dqt                    an  $(n \text{ choose } 4) \times n$  (or  $n+1$ ) matrix of form output by `DominantQuartetTable`  
(Note:  $n+1$ th column of dqt is ignored)

**Value**

a pairwise distance matrix on n taxa

**See Also**

[DominantQuartetTable](#), [QDS](#), [QDC](#), [WQdist](#)

**Examples**

```
gtrees=read.tree(file=system.file("genetreeSample",package="MSCquartets"))
taxanames=AllTaxonNames(gtrees)
QT=QuartetTable(gtrees,taxanames[1:5])
RQT=ResolvedQuartetTable(QT)
DQT=DominantQuartetTable(RQT)
Dist=Qdist(DQT)
tree=NJ(Dist)
```

---

QDS	<i>Compute Quartet Distance Supertree</i>
-----	---

---

**Description**

Apply the Quartet Distance Supertree method of [R19] to a table specifying a collection of quartets on n taxa.

**Usage**

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

**Arguments**

dqt                    an  $(n \text{ choose } 4) \times n$  (or  $n+1$ ) matrix of form output by `DominantQuartetTable`  
(Note:  $n+1$ th column of dqt is ignored)

method                tree building method function (`fastme.bal`, `nj`, etc.)

Details

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

Value

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

See Also

[DominantQuartetTable](#), [Qdist](#), [QDC](#), [WQDS](#), [WQDC](#), [WQDCrec](#)

Examples

```
gtrees=read.tree(file=system.file("genetreeSample",package="MSCquartets"))
taxanames=AllTaxonNames(gtrees)
QT=QuartetTable(gtrees,taxanames[1:5])
RQT=ResolvedQuartetTable(QT)
DQT=DominantQuartetTable(RQT)
tree=QDS(DQT)
```

---

QNdist	<i>Compute network quartet distance between taxa</i>
--------	--

---

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 in [ABR19] generalizes the quartet distance of [R19].

Usage

```
QNdist(pTable, alpha0, beta0)
```

Arguments

- pTable            a table of quartets and p-values, as computed by NANUQ, or MultiIndepQuartetTreeTest and MultiIndepQuartetStarTest
- 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

See Also

[NANUQ](#), [NANUQdist](#)



---

quartet.table.print	<i>Print a quartet table with nice formatting</i>
---------------------	---

---

**Description**

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

**Usage**

```
quartet.table.print(qt)
```

**Arguments**

qt	a table such as returned by QuartetTable, ResolvedQuartetTable, or DominantQuartetTable, possibly with extra columns added by other functions
----	---

**Examples**

```
gtrees=read.tree(file=system.file("genetreeSample",package="MSCquartets"))
taxonnames=AllTaxonNames(gtrees)
QT=QuartetTable(gtrees,taxonnames)
quartet.table.print(QT)
RQT=ResolvedQuartetTable(QT)
quartet.table.print(RQT)
pTable=MultiIndepQuartetTreeTest(RQT,"T3")
quartet.table.print(pTable)
DQT=DominantQuartetTable(RQT)
quartet.table.print(DQT)
```

---

QuartetStarTest	<i>Hypothesis test for quartet counts fitting a star tree under the MSC</i>
-----------------	---

---

**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)
```

---

QuartetTable	<i>Produce table of counts of quartets displayed on trees</i>
--------------	---

---

**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	multi phylo 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 than 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**

[ResolvedQuartetTable](#), [DominantQuartetTable](#)

## Examples

```
gtrees=read.tree(file=system.file("genetreeSample",package="MSCquartets"))
taxonnames=AllTaxonNames(gtrees)
QT=QuartetTable(gtrees,taxonnames[1:5])
RQT=ResolvedQuartetTable(QT)
DQT=DominantQuartetTable(RQT)
```

---

QuartetTreeErrorProb	<i>Bayesian posterior probability of error in 4-taxon unrooted species tree topology estimate</i>
----------------------	---

---

## 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 [MAR19] 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

## 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 [MAR19], vs.  $H_1 =$  everything else.  $T1$  is for a specific 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 frequencies,
model	"T1" or "T3", for the models of [MAR19],
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 in [MAR19] as well as parametric bootstrapping, with other procedures for when some counts are small. Due to the singularity and boundary of the models, when the topology and/or the internal quartet branch length is unknown 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 [MAR19]. For model  $T1$  this test is slightly anticonservative if the true internal edge of the quartet species tree is very short. Although the test generally performs well in practice, it lacks a uniform asymptotic guarantee over the full parameter space.

If `method="conservative"`, a more conservative test described in [MAR19] 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 test 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 bootstrapping 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 [MAR19] 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.

### See Also

[MultiIndepQuartetTreeTest](#)

### Examples

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

---

ResolvedQuartetTable    *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

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

### Arguments

qt	table as produced by QuartetTable of size $(n \text{ choose } 4) \times (n+4)$
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  $(n \text{ choose } 4) \times (n+3)$ , similar to QuartetTable

### See Also

[QuartetTable](#), [DominantQuartetTable](#)

Examples

```
gtrees=read.tree(file=system.file("genetreeSample",package="MSCquartets"))
taxonnames=AllTaxonNames(gtrees)
QT=QuartetTable(gtrees,taxonnames[1:5])
RQT=ResolvedQuartetTable(QT)
DQT=DominantQuartetTable(RQT)
```

---

SLabels	<i>Label vertices of 2-d simplex</i>
---------	--------------------------------------

---

Description

Label vertices of 2-d simplex

Usage

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

Arguments

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

Value

None

See Also

[SPoint](#), [SPrepare](#), [SSegment](#), [SText](#), [StoP](#)

Examples

```
SPrepare("T3", "Example Plot")
SLabels("ab|cd", "ac|bd", "ad|bc")
```

---

SPoint	<i>Plot point in 2-d simplex</i>
--------	----------------------------------

---

**Description**

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

**Usage**

```
SPoint(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**

[SLabels](#), [SPrepare](#), [SSegment](#), [SText](#), [StoP](#)

**Examples**

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

---

SPrepare	<i>Draw 2-d probability simplex, with model lines for T3 or T1 model</i>
----------	--

---

**Description**

Outline the 2-d simplex, and draw the T1 or T3 model points for quartet frequencies. The models are described more fully in [MAR19].

**Usage**

```
SPrepare(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

See Also

[SLabels](#), [SPoint](#), [SSegment](#), [SText](#), [StoP](#)

Examples

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

---

SSegment	<i>Plot line segment in 2-d simplex</i>
----------	---

---

Description

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

Usage

```
SSegment(v, w, ...)
```

Arguments

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

Value

None

See Also

[SLabels](#), [SPoint](#), [SPrepare](#), [SText](#), [StoP](#)

Examples

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



SText

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

Add text at a point in 2-d simplex

**Usage**

```
SText(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**

[SLabels](#), [SPoint](#), [SPrepare](#), [SSegment](#), [StoP](#)

**Examples**

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

StoP

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

Convert 3-d coordinates to 2-d simplex coordinates

**Usage**

```
StoP(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

[SLabels](#), [SPoint](#), [SPrepare](#), [SSegment](#), [SText](#)

Examples

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

---

T1density	<i>Probability density function for Model T1 of [MAR19], Proposition 5.2</i>
-----------	--

---

Description

Probability density function for Model T1 of [MAR19], 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

See Also

[T3density](#)

---

T3density	<i>Probability density function for Model T3 of [MAR19], Proposition 4.2</i>
-----------	--

---

Description

Probability density function for Model T3 of [MAR19], 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

See Also

[T1density](#)

---

WQDC	<i>Compute Weighted Quartet Distance Consensus tree from gene tree data</i>
------	---

---

Description

Compute the Weighted Quartet Distance Consensus [YR19] 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 <ol style="list-style-type: none"><li>1. a character string giving the name of a file containing gene trees in Newick</li><li>2. a multiphylo object containing gene trees,</li><li>3. a resolved quartet table, as produced by ResolveQuartetTable</li></ol>
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 distance 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"

See Also

[QuartetTable](#), [ResolvedQuartetTable](#), [DominantQuartetTable](#), [WQdist](#), [WQDCrec](#) [WQDS](#), [QDC](#)

Examples

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

---

WQDCrec	<i>Compute the Recursive Weighted Quartet Distance Consensus tree from gene tree data</i>
---------	---

---

Description

Infer a metric species tree from counts of quartets displayed on a collection of gene trees, as described in [YR2019].

Usage

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

Arguments

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

Details

The algorithm counts quarets displayed on the gene trees, builds a tree using WQDS, determines the split corresponding 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  $(n \text{ choose } 4) \times (n+3)$ . 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"

See Also

[ResolvedQuartetTable](#), [QuartetTable](#), [QDC](#), [QDS](#), [CollapseQuartetTable](#)

Examples

```
gtrees=read.tree(file=system.file("genetreeSample",package="MSCquartets"))
taxanames=AllTaxonNames(gtrees)
QT=QuartetTable(gtrees,taxanames[1:5])
RQT=ResolvedQuartetTable(QT)
stree=WQDCrec(RQT)
plot(stree)
stree
```

---

WQdist

---

*Compute the Weighted Quartet Distance between taxa*


---

**Description**

Compute the Weighted Quartet Distance between taxa of [YR19] from a table specifying a collection of quartets on  $n$  taxa and the quartets' internal branch lengths.

**Usage**

```
WQdist(dqt)
```

**Arguments**

dqt                      an  $(n \text{ choose } 4) \times n+1$  matrix of form output by DominantQuartetTable

**Value**

A pairwise distance matrix on  $n$  taxa

**See Also**

[DominantQuartetTable](#), [WQDSAdjustLengths](#), [WQDS](#), [WQDC](#), [WQDCrec](#), [WQdist](#)

**Examples**

```
gtrees=read.tree(file=system.file("genetreeSample",package="MSCquartets"))
taxanames=AllTaxonNames(gtrees)
QT=QuartetTable(gtrees,taxanames[1:5])
RQT=ResolvedQuartetTable(QT)
DQT=DominantQuartetTable(RQT,bigweights="finite")
D=WQdist(DQT)
tree=NJ(D)
stree=WQDSAdjustLengths(tree)
```

---

WQDS

---

*Compute the Weighted Quartet Distance Supertree*


---

**Description**

Apply the Weighted Quartet Distance Supertree method of [YR19] 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 DominantQuartetTable
method	tree building method function (fastme.bal, NJ, etc.)

Details

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

Value

an unrooted metric tree, of type "phylo"

See Also

[DominantQuartetTable](#), [WQdist](#), [WQDSAdjustLengths](#), [WQDC](#), [WQDCrec](#), [QDS](#)

Examples

```
gtrees=read.tree(file=system.file("genetreeSample",package="MSCquartets"))
taxanames=AllTaxonNames(gtrees)
QT=QuartetTable(gtrees,taxanames[1:5])
RQT=ResolvedQuartetTable(QT)
DQT=DominantQuartetTable(RQT,bigweights= "finite")
tree=WQDS(DQT)
```

---

WQDSAdjustLengths	<i>Adjust edge lengths on tree built from Weighted Quartet distance to estimate metric tree</i>
-------------------	---

---

Description

Modify edge lengths of tree built from distance table produced by WQdist, 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 in [YR19], 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 WQDCrec, as their edges are already adjusted. It can be used on trees built from the distance computed by WQdist.

**Value**

an unrooted metric tree, of type "phylo"

**See Also**

[WQDS](#), [WQDC](#)

**Examples**

```
gtrees=read.tree(file=system.file("genetreeSample",package="MSCquartets"))
taxanames=AllTaxonNames(gtrees)
QT=QuartetTable(gtrees,taxanames[1:5])
RQT=ResolvedQuartetTable(QT)
DQT=DominantQuartetTable(RQT,bigweights="finite")
D=WQdist(DQT)
tree=NJ(D)
stree=WQDSAdjustLengths(tree)
```

# Index

AllTaxonNames, [2](#)

CollapseQuartetTable, [3](#), [28](#)

DominantQuartetTable, [3](#), [10](#), [14–16](#), [18](#), [21](#),  
[27](#), [29](#), [30](#)

HolmBonferroni, [4](#)

log10Hist, [5](#)

MSCQuartets, [6](#)

MSCQuartets-package (MSCQuartets), [6](#)

MultiIndepQuartetStarTest, [5](#), [7](#), [8](#), [10](#), [11](#),  
[13](#)

MultiIndepQuartetTreeTest, [5](#), [7](#), [8](#), [10](#), [11](#),  
[13](#), [21](#)

NANUQ, [9](#), [11](#), [13](#), [16](#)

NANUQdist, [10](#), [11](#), [13](#), [16](#)

NexusDist, [12](#)

PlotQuartetTest, [7](#), [8](#), [10](#), [12](#)

power.div.stat, [13](#)

QDC, [14](#), [15](#), [16](#), [27](#), [28](#)

Qdist, [14](#), [15](#), [16](#)

QDS, [14](#), [15](#), [15](#), [28](#), [30](#)

QNdist, [16](#)

quartet.table.print, [17](#)

QuartetStarTest, [7](#), [17](#)

QuartetTable, [4](#), [10](#), [14](#), [18](#), [21](#), [27](#), [28](#)

QuartetTreeErrorProb, [19](#)

QuartetTreeTest, [7](#), [8](#), [20](#)

ResolvedQuartetTable, [4](#), [7](#), [8](#), [14](#), [18](#), [21](#),  
[27](#), [28](#)

SLabels, [22](#), [23–26](#)

SPoint, [22](#), [23](#), [24–26](#)

SPrepare, [22](#), [23](#), [23](#), [24–26](#)

SSegment, [22–24](#), [24](#), [25](#), [26](#)

SText, [22–24](#), [25](#), [26](#)

StoP, [22–25](#), [25](#)

T1density, [26](#), [27](#)

T3density, [26](#), [26](#)

WQDC, [14](#), [16](#), [27](#), [29–31](#)

WQDCrec, [3](#), [14](#), [16](#), [27](#), [28](#), [29](#), [30](#)

WQdist, [15](#), [27](#), [29](#), [29](#), [30](#)

WQDS, [16](#), [27](#), [29](#), [29](#), [31](#)

WQDSAdjustLengths, [29](#), [30](#), [30](#)