

Package ‘Quartet’

March 6, 2019

Version 1.0.2

Date 2019-03-06

Title Comparison of Phylogenetic Trees Using Quartet and Bipartition Measures

Description Calculates the number of four-taxon subtrees consistent with a pair of cladograms, calculating the symmetric quartet distance of Bandelt & Dress (1986), Reconstructing the shape of a tree from observed dissimilarity data, *Advances in Applied Mathematics*, 7, 309-343 <doi:10.1016/0196-8858(86)90038-2>, and using the tqDist algorithm of Sand et al. (2014), tqDist: a library for computing the quartet and triplet distances between binary or general trees, *Bioinformatics*, 30, 2079–2080 <doi:10.1093/bioinformatics/btu157> for pairs of bifurcating trees.

URL <https://github.com/ms609/Quartet>

BugReports <https://github.com/ms609/Quartet/issues>

Copyright Incorporates code modified from tqDist <doi:10.1093/bioinformatics/btu157>.

License GPL (>= 2)

Encoding UTF-8

Language en-GB

Depends R (>= 3.4.0)

Imports ape, memoise, Rdpack, Ternary (>= 1.0), TreeSearch (>= 0.2.1)

Suggests bookdown, knitr, phangorn, Rcpp, rmarkdown, testthat, usethis

RdMacros Rdpack

LinkingTo Rcpp

LazyData true

ByteCompile true

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation yes

Author Martin R. Smith [aut, cre, cph]
 (<<https://orcid.org/0000-0001-5660-1727>>),
 Andreas Sand [ant],
 Gerth Stølting Brodal [ant],
 Rolf Fagerberg [ant],
 Thomas Mailund [ant],
 Christian N. S. Pedersen [ant],
 Jens Johansen [ant],
 Morten K. Holt [ant]

Maintainer Martin R. Smith <martin.smith@durham.ac.uk>

Repository CRAN

Date/Publication 2019-03-06 13:00:10 UTC

R topics documented:

AllQuartets	2
CompareQuartets	3
CompareSplits	5
Distances	6
PlotQuartet	8
QuartetPoints	9
QuartetState	10
ResolvedQuartets	12
SharedQuartetStatus	13
SimilarityMetrics	15
SplitStatus	18
sq_trees	19
SymmetricDifferenceLineEnds	20
TQDist	21

Index 23

AllQuartets *List all quartets*

Description

Lists all choices of four taxa from a tree.

A more computationally efficient alternative to [combn](#), AllQuartets uses [memoise](#) to make repeated calls faster.

Usage

AllQuartets(n_tips)

Arguments

`n_tips` Integer, specifying the number of tips in a tree.

Value

Returns a list of length `choose(n_tips, 4)`, with each entry corresponding to a unique selection of four different integers less than or equal to `n_tips`

Author(s)

Martin R. Smith

See Also

[combn](#)

Other quartet counting functions: [ResolvedQuartets](#)

Examples

```
n_tips <- 6
AllQuartets(n_tips)

combn(n_tips, 4) # Provides the same information, but for large
                 # values of n_tips is significantly slower.
```

CompareQuartets	<i>Compare quartet states by explicit enumeration</i>
-----------------	---

Description

Uses explicit enumeration to compare two lists of quartet states, detailing how many are identical and how many are unresolved. For most purposes, the faster function [QuartetStatus](#) will be preferable.

Usage

```
CompareQuartets(x, cf)
```

Arguments

`x, cf` List of quartet states, perhaps generated by [QuartetStates](#).

Value

Returns an array of seven numeric elements, corresponding to the quantities of Estabrook *et al.* (1985):

N The total number of quartet *statements* for two trees of n tips, i.e. $2Q$.

Q The total number of quartets for n tips.

s The number of quartets that are resolved identically in both trees.

d The number of quartets that are resolved differently in each tree.

r1 The number of quartets that are resolved in tree 1, but not in tree 2.

r2 The number of quartets that are resolved in tree 2, but not in tree 1.

u The number of quartets that are unresolved in both trees.

Author(s)

Martin R. Smith

References

Estabrook GF, McMorris FR, Meacham CA (1985). “Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units.” *Systematic Zoology*, **34**(2), 193–200. doi: [10.2307/2413326](https://doi.org/10.2307/2413326).

See Also

[QuartetStatus](#), generates this output from a list of trees.

Other element-by-element comparisons: [CompareSplits](#), [PairSharedQuartetStatus](#), [QuartetState](#), [SharedQuartetStatus](#), [SplitStatus](#)

Examples

```
n_tip <- 6
trees <- list(ape::rtree(n_tip, tip.label=seq_len(n_tip), br=NULL),
             ape::rtree(n_tip, tip.label=seq_len(n_tip), br=NULL))
splits <- lapply(trees, TreeSearch::Tree2Splits)
quartets <- QuartetStates(splits)
CompareQuartets(quartets[[1]], quartets[[2]])
```

CompareSplits *Compare status of bipartitions*

Description

Reports whether bipartition splits are present or contradicted in a set of reference splits.

Usage

```
CompareSplits(splits, splits2)
```

```
CompareBipartitions(splits, splits2)
```

Arguments

splits	A two-dimensional matrix of logicals, with rows corresponding to terminal taxa, and each column defining a split that divides taxa marked TRUE from taxa marked FALSE. Possibly generated using [Tree2Splits]. The parity of each column is irrelevant.
splits2	A matrix of bipartitions against which to compare splits. If row names are present, then all rows present in splits must be present in splits2. If they are absent, then both matrices must have the same number of rows, and tips will be assumed to be in the same sequence.

Value

A named vector of six integers, listing the number of unique splits that:

- **N** exist in total; i.e. the number of splits in splits1 plus the number in splits2, equivalent to $2s + d1 + d2 + r1 + r2$;
- **s** occur in both splits1 and splits2;
- **d1** occur in splits1 but are contradicted by splits2;
- **d2** occur in splits2 but are contradicted by splits1;
- **r1** occur in splits1 only, being neither present in nor contradicted by splits2;
- **r2** occur in splits2 only, being neither present in nor contradicted by splits1;
- **RF** occur in one tree only; i.e. $d1 + d2 + r1 + r2$, the Robinson-Foulds distance.

Author(s)

Martin R. Smith

References

Estabrook GF, McMorris FR, Meacham CA (1985). “Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units.” *Systematic Zoology*, **34**(2), 193–200. doi: [10.2307/2413326](https://doi.org/10.2307/2413326).

Robinson DF, Foulds LR (1981). “Comparison of phylogenetic trees.” *Mathematical Biosciences*, **53**(1-2), 131–147. doi: [10.1016/00255564\(81\)900432](https://doi.org/10.1016/00255564(81)900432).

See Also

[CompareQuartets](#): equivalent function for quartets.

Other element-by-element comparisons: [CompareQuartets](#), [PairSharedQuartetStatus](#), [QuartetState](#), [SharedQuartetStatus](#), [SplitStatus](#)

Distances

Triplet and quartet distances with tqDist

Description

Functions to calculate triplet and quartet distances between pairs of trees.

Usage

```
QuartetDistance(file1, file2)
```

```
QuartetAgreement(file1, file2)
```

```
PairsQuartetDistance(file1, file2)
```

```
OneToManyQuartetAgreement(file1, file2)
```

```
AllPairsQuartetDistance(file)
```

```
AllPairsQuartetAgreement(file)
```

```
TripletDistance(file1, file2)
```

```
PairsTripletDistance(file1, file2)
```

```
AllPairsTripletDistance(file)
```

Arguments

file, file1, file2

Paths to files containing a tree or trees in Newick format.

Value

Distance functions return the distance between the requested trees.

Agreement functions return the number of triplets or quartets that are:

- A, resolved in the same fashion in both trees;
- E, unresolved in both trees.

Comparing a tree against itself yields the totals (A+B+C) and (D+E) referred to by Brodal *et al.* (2013) and Holt *et al.* (2014).

Functions

- `QuartetDistance`: Returns the quartet distance between the tree in `file1` and the tree in `file2`.
- `QuartetAgreement`: Returns a vector of length two, listing [1] the number of resolved quartets that agree (A); [2] the number of quartets that are unresolved in both trees (E). See Brodal *et al.* (2013).
- `PairsQuartetDistance`: Quartet distance between the tree on each line of `file1` and the tree on the corresponding line of `file2`.
- `OneToManyQuartetAgreement`: Quartet distance between the tree in `file1` and the tree on each line of `file2`.
- `AllPairsQuartetDistance`: Quartet distance between each tree listed in `file` and each other tree therein.
- `AllPairsQuartetAgreement`: Quartet status for each pair of trees in `file`.
- `TripletDistance`: Triplet distance between the single tree given in each file.
- `PairsTripletDistance`: Triplet distance between the tree on each line of `file1` and the tree on the corresponding line of `file2`.
- `AllPairsTripletDistance`: Triplet distance between each tree listed in `file` and each other tree therein.

Author(s)

- Algorithms: Brodal *et al.* (2013); Holt *et al.* (2014).
- C implementation: Sand *et al.* (2014); modified for portability by Martin R. Smith.
- R interface: Martin R. Smith.

References

Brodal GS, Fagerberg R, Mailund T, Pedersen CNS, Sand A (2013). “Efficient algorithms for computing the triplet and quartet distance between trees of arbitrary degree.” *SODA '13 Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1814–1832. doi: [10.1137/1.9781611973105.130](https://doi.org/10.1137/1.9781611973105.130).

Holt MK, Johansen J, Brodal GS (2014). “On the scalability of computing triplet and quartet distances.” In *Proceedings of 16th Workshop on Algorithm Engineering and Experiments (ALENEX) Portland, Oregon, USA*.

Sand A, Holt MK, Johansen J, Brodal GS, Mailund T, Pedersen CNS (2014). “tqDist: a library for computing the quartet and triplet distances between binary or general trees.” *Bioinformatics*, **30**(14), 2079–2080. ISSN 1460-2059, doi: [10.1093/bioinformatics/btu157](https://doi.org/10.1093/bioinformatics/btu157).

 PlotQuartet

Plot quartet on a tree topology

Description

Draws a tree, highlighting the members of a specified quartet in colour.

Usage

```
PlotQuartet(tree, quartet, overwritePar = TRUE, caption = TRUE, ...)
```

Arguments

tree	A tree of class phylo , or a list of such trees.
quartet	A vector of four integers, corresponding to numbered tips on the tree; or a character vector specifying the labels of four tips.
overwritePar	Logical specifying whether to use existing par <code>mfrow</code> and <code>mar</code> parameters (FALSE), or to plot trees side-by-side in a new graphical device (TRUE).
caption	Logical specifying whether to annotate each plot to specify whether the quartet selected is in the same or a different state to the reference tree.
...	Additional parameters to send to plot .

Value

Returns `invisible()`, having plotted a tree in which the first two members of `quartet` are highlighted in orange, and the second two highlighted in blue.

Author(s)

Martin R. Smith

Examples

```
data('sq_trees')

par(mfrow=c(3, 5), mar=rep(0.5, 4))
PlotQuartet(sq_trees, c(2, 5, 3, 8), overwritePar = FALSE)
```

`QuartetPoints`*Plot tree differences on ternary plots*

Description

Generate points to depict tree difference (in terms of resolution and accuracy) on a ternary plot.

Usage

```
QuartetPoints(trees, cf = trees[[1]])
```

```
SplitPoints(trees, cf = trees[[1]])
```

```
BipartitionPoints(trees, cf = trees[[1]])
```

Arguments

<code>trees</code>	A list of trees of class <code>phylo</code> , with identically labelled tips.
<code>cf</code>	Comparison tree of class <code>phylo</code> . If unspecified, each tree is compared to the first tree in <code>trees</code> .

Details

The ternary plot will depict the number of quartets or bipartitions that are:

- resolved in the reference tree (`cf`), but neither present nor contradicted in each comparison tree (`trees`);
- resolved differently in the reference and the comparison tree;
- resolved in the same manner in the reference and comparison trees.

If the reference tree (`cf`) is taken to represent the best possible knowledge of the 'true' topology, then polytomies in the reference tree represent uncertainty. If a tree in `trees` resolves relationships within this polytomy, it is not possible to establish (based only on the reference tree) whether this resolution is correct or erroneous. As such, extra resolution in `trees` that is neither corroborated nor contradicted by `cf` is ignored.

Value

A data frame listing the ternary coordinates of trees, based on the amount of information that they have in common with the comparison tree (which defaults to the first member of the list, if unspecified).

Functions

- `SplitPoints`: Uses partitions instead of quartets to calculate tree distances.

Author(s)

Martin R. Smith

References

Smith MR (2019). “Bayesian and parsimony approaches reconstruct informative trees from simulated morphological datasets.” *Biology Letters*, **15**, 20180632. doi: [10.1098/rsbl.2018.0632](https://doi.org/10.1098/rsbl.2018.0632).

Examples

```
{
  library('Ternary')
  data('sq_trees')

  TernaryPlot(alab='Unresolved', blab='Contradicted', clab='Consistent', point='right')
  TernaryLines(list(c(0, 2/3, 1/3), c(1, 0, 0)), col='red', lty='dotted')
  TernaryText(QuartetPoints(sq_trees, cf=sq_trees$collapse_one), 1:15,
    col=Ternary::cbPalette8[2], cex=0.8)
  TernaryText(SplitPoints(sq_trees, cf=sq_trees$collapse_one), 1:15,
    col=Ternary::cbPalette8[3], cex=0.8)
  legend('bottomright', c("Quartets", "Bipartitions"), bty='n', pch=1, cex=0.8,
    col=Ternary::cbPalette8[2:3])
}
```

 QuartetState

Quartet State(s)

Description

Report the status of the specified quartet(s).

Usage

```
QuartetState(tips, bips)
```

```
QuartetStates(splits)
```

Arguments

tips	A four-element array listing a quartet of tips, either by their number (if class numeric) or their name (if class character).
bips	Bipartitions to evaluate.
splits	A list of bipartition splits, perhaps generated using Tree2Splits , with row names corresponding to taxon labels. If a tree or list of trees (of class phylo) is sent instead, it will be silently converted to its constituent splits.

Details

One of the three possible four-taxon trees will be consistent with any set of bipartitions generated from a fully resolved tree. If the taxa are numbered 1 to 4, this tree can be identified by naming the tip most closely related to taxon 1. If a set of bipartitions is generated from a tree that contains polytomies, it is possible that all three four-taxon trees are consistent with the set of bipartitions.

Value

Returns \emptyset if the relationships of the four taxa are not constrained by the provided bipartitions, or the index of the closest relative to `tips[1]`, otherwise.

Functions

- `QuartetStates`: A convenience wrapper that need only be provided with a tree or a list of splits.

Author(s)

Martin R. Smith

References

Estabrook GF, McMorris FR, Meacham CA (1985). "Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units." *Systematic Zoology*, **34**(2), 193–200. doi: [10.2307/2413326](https://doi.org/10.2307/2413326).

See Also

[CompareQuartets](#), used to compare quartet states between trees.

Other element-by-element comparisons: [CompareQuartets](#), [CompareSplits](#), [PairSharedQuartetStatus](#), [SharedQuartetStatus](#), [SplitStatus](#)

Examples

```
{
  n_tip <- 6
  trees <- list(ape::rtree(n_tip, tip.label=seq_len(n_tip), br=NULL),
              ape::rtree(n_tip, tip.label=seq_len(n_tip), br=NULL))
  splits <- lapply(trees, TreeSearch::Tree2Splits)
  QuartetState(c(1, 3, 4, 6), splits[[2]])
  QuartetState(1:4, splits[[1]]) == QuartetState(1:4, splits[[2]])
  vapply(AllQuartets(n_tip), QuartetState, bips=splits[[1]], double(1))
}
```

ResolvedQuartets	<i>Count resolved quartets</i>
------------------	--------------------------------

Description

Counts how many quartets are resolved or unresolved in a given tree, following Brodal *et al.* (2013).

Usage

```
ResolvedQuartets(tree, countTriplets = FALSE)
```

```
ResolvedTriplets(tree)
```

Arguments

tree	A tree of class phylo .
countTriplets	Logical; if TRUE, the function will return the number of triplets instead of the number of quartets.

Value

A vector of length two, listing the number of quartets (or triplets) that are [1] resolved; [2] unresolved in the specified tree.

Functions

- ResolvedTriplets: Convenience function to calculate the number of resolved/unresolved triplets.

Author(s)

Martin R. Smith

References

Brodal GS, Fagerberg R, Mailund T, Pedersen CNS, Sand A (2013). “Efficient algorithms for computing the triplet and quartet distance between trees of arbitrary degree.” *SODA '13 Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1814–1832. doi: [10.1137/1.9781611973105.130](https://doi.org/10.1137/1.9781611973105.130).

See Also

Other quartet counting functions: [AllQuartets](#)

Examples

```

{
  data(sq_trees)

  ResolvedTriplets(sq_trees$collapse_some)
  # Equivalent to:
  ResolvedQuartets(sq_trees$collapse_some, countTriplets=TRUE)

  vapply(sq_trees, ResolvedQuartets, integer(2))
}

```

SharedQuartetStatus *Status of quartets*

Description

Determines the number of quartets that are consistent within pairs of cladograms.

Usage

```
SharedQuartetStatus(trees, cf = trees[[1]])
```

```
QuartetStatus(trees, cf = trees[[1]])
```

Arguments

trees	A list of trees of class <code>phylo</code> , with identically labelled tips.
cf	Comparison tree of class <code>phylo</code> . If unspecified, each tree is compared to the first tree in trees.

Details

Given a list of trees, returns the number of quartet statements present in the reference tree (the first entry in `trees`, if `cf` is not specified) that are also present in each other tree. A random pair of fully resolved trees is expected to share $\text{choose}(n_{\text{tip}}, 4) / 3$ quartets.

If trees do not bear the same number of tips, `SharedQuartetStatus` will consider only the quartets that include taxa common to both trees.

From this information it is possible to calculate how many of all possible quartets occur in one tree or the other, though there is not yet a function calculating this; **let us know** if you would appreciate this functionality.

The status of each quartet is calculated using the algorithms of Brodal *et al.* (2013) and Holt *et al.* (2014), implemented in the `tqdist` C library (Sand *et al.* 2014).

Value

Returns a two dimensional array. Rows correspond to the input trees; the first row will report a perfect match if the first tree is specified as the comparison tree (or if *cf* is not specified). Columns list the status of each quartet:

N The total number of quartet *statements* for two trees of n tips, i.e. $2Q$.

Q The total number of quartets for n tips.

s The number of quartets that are resolved identically in both trees.

d The number of quartets that are resolved differently in each tree.

r1 The number of quartets that are resolved in tree 1, but not in tree 2.

r2 The number of quartets that are resolved in tree 2, but not in tree 1.

u The number of quartets that are unresolved in both trees.

Functions

- SharedQuartetStatus: Reports split statistics obtained after removing all tips that do not occur in both trees being compared.

Author(s)

Martin R. Smith

References

Brodal GS, Fagerberg R, Mailund T, Pedersen CNS, Sand A (2013). “Efficient algorithms for computing the triplet and quartet distance between trees of arbitrary degree.” *SODA '13 Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1814–1832. doi: [10.1137/1.9781611973105.130](https://doi.org/10.1137/1.9781611973105.130).

Estabrook GF, McMorris FR, Meacham CA (1985). “Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units.” *Systematic Zoology*, **34**(2), 193–200. doi: [10.2307/2413326](https://doi.org/10.2307/2413326).

Holt MK, Johansen J, Brodal GS (2014). “On the scalability of computing triplet and quartet distances.” In *Proceedings of 16th Workshop on Algorithm Engineering and Experiments (ALENEX) Portland, Oregon, USA*.

Sand A, Holt MK, Johansen J, Brodal GS, Mailund T, Pedersen CNS (2014). “tqDist: a library for computing the quartet and triplet distances between binary or general trees.” *Bioinformatics*, **30**(14), 2079–2080. ISSN 1460-2059, doi: [10.1093/bioinformatics/btu157](https://doi.org/10.1093/bioinformatics/btu157).

See Also

[SplitStatus](#): Uses bipartition splits (groups/clades defined by nodes or edges of the tree) instead of quartets as the unit of comparison.

Other element-by-element comparisons: [CompareQuartets](#), [CompareSplits](#), [PairSharedQuartetStatus](#), [QuartetState](#), [SplitStatus](#)

Examples

```
data('sq_trees')
# Calculate the status of each quartet
sq_status <- QuartetStatus(sq_trees)

# Calculate Estabrook et al's similarity measures:
SimilarityMetrics(sq_status)
```

SimilarityMetrics *Tree similarity metrics*

Description

Functions to calculate tree similarity / difference metrics.

Usage

```
SimilarityMetrics(elementStatus, similarity = TRUE)
DoNotConflict(elementStatus, similarity = TRUE)
ExplicitlyAgree(elementStatus, similarity = TRUE)
StrictJointAssertions(elementStatus, similarity = TRUE)
SemiStrictJointAssertions(elementStatus, similarity = TRUE)
SymmetricDifference(elementStatus, similarity = TRUE)
RobinsonFoulds(elementStatus, similarity = FALSE)
MarczewskiSteinhaus(elementStatus, similarity = TRUE)
SteelPenny(elementStatus, similarity = TRUE)
QuartetDivergence(elementStatus, similarity = TRUE)
```

Arguments

elementStatus Two-dimensional integer array, with rows corresponding to counts of matching quartets or partitions for each tree, and columns named according to the output of [QuartetStatus] or [SplitStatus].

similarity Logical specifying whether to calculate the similarity or dissimilarity.

Details

Estabrook *et al.* (1985, table 2) define four similarity metrics in terms of the total number of quartets (Q), the number of quartets resolved in the same manner in two trees (s), the number resolved differently in both trees (d), the number resolved in tree 1 or 2 but unresolved in the other tree ($r1$, $r2$), and the number that are unresolved in both trees (u).

The similarity metrics are then given as below. The dissimilarity metrics are their complement (i.e. $1 - \text{similarity}$), and can be calculated algebraically using the identity $Q = s + d + r1 + r2 + u$.

- Do Not Conflict (DC): $1 - (d / Q)$
- Explicitly Agree (EA): s / Q
- Strict Joint Assertions (SJA): $s / (s + d)$
- SemiStrict Joint Assertions (SSJA): $s / (s + d + u)$

(The numerator of the SemiStrict Joint Assertions similarity metric is given in Estabrook *et al.* (1985)'s table 2 as $s + d$, but this is interpreted, with reference to their text, as a typographic error.)

Steel & Penny (1993) propose a further metric, which they denote dQ , which this package calculates using the function `SteelPenny`:

- Steel & Penny's Quartet Metric (dQ): $(s + u) / Q$

Although defined using quartets, analogous values can be calculated using partitions by replacing s and d with $2_s_$ and $(d1 + d2)$, and Q with N . Note that, for reasons listed elsewhere (see Smith 2019, supplementary text), quartets offer a more meaningful measure of the amount of information shared by two trees.

Another take on tree similarity is to consider the symmetric difference: that is, the number of quartets or partitions present in one tree that do not appear in the other, originally used to measure tree similarity by Robinson & Foulds (1981).

- Robinson Foulds (RF): $d1 + d2 + r1 + r2$

With quartets, $d1 + d2 = 2 d$.

(Note that, given the familiarity of the Robinson Foulds distance metric, this quantity is by default expressed as a difference rather than a similarity.)

To contextualize the symmetric difference, it may be normalized against:

- The total number of resolved quartets or partitions present in both trees (Day 1986):
 - Day's Symmetric Difference (SD): $(d1 + d2 + r1 + r2) / (d1 + d2 + 2 s + r1 + r2)$
- The total distinctly resolved quartets or partitions (Day 1986):
 - Marczewski-Steinhaus (MS): $(d1 + d2 + r1 + r2) / (d1 + d2 + s + r1 + r2)$
- The maximum number of quartets or partitions that could have been resolved, given the number of tips (Smith 2019; analogous to the Kullback-Leibler divergence):
 - Quartet Divergence: $(d1 + d2 + r1 + r2) / 2 Q$

The partition equivalent to the latter will depend on the question being asked, as Q should denote the maximum difference that *could* have been obtained.

Value

SimilarityMetrics returns a named two-dimensional array in which each row corresponds to an input tree, and each column corresponds to one of the listed measures.

DoNotConflict and equivalents return a named vector describing the requested similarity (or difference) between the trees.

Author(s)

Martin R. Smith

References

Day WH (1986). “Analysis of quartet dissimilarity measures between undirected phylogenetic trees.” *Systematic Biology*, **35**(3), 325–333. doi: [10.1093/sysbio/35.3.325](https://doi.org/10.1093/sysbio/35.3.325).

Estabrook GF, McMorris FR, Meacham CA (1985). “Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units.” *Systematic Zoology*, **34**(2), 193–200. doi: [10.2307/2413326](https://doi.org/10.2307/2413326).

Kullback S, Leibler RA (1951). “On information and sufficiency.” *The Annals of Mathematical Statistics*, **22**(1), 79–86. doi: [10.1214/aoms/1177729694](https://doi.org/10.1214/aoms/1177729694).

Marczewski E, Steinhaus H (1958). “On a certain distance of sets and the corresponding distance of functions.” *Colloquium Mathematicae*, **6**(1), 319–327. <https://eudml.org/doc/210378>.

Robinson DF, Foulds LR (1981). “Comparison of phylogenetic trees.” *Mathematical Biosciences*, **53**(1-2), 131–147. doi: [10.1016/00255564\(81\)900432](https://doi.org/10.1016/00255564(81)900432).

Smith MR (2019). “Bayesian and parsimony approaches reconstruct informative trees from simulated morphological datasets.” *Biology Letters*, **15**, 20180632. doi: [10.1098/rsbl.2018.0632](https://doi.org/10.1098/rsbl.2018.0632).

Steel MA, Penny D (1993). “Distributions of tree comparison metrics—some new results.” *Systematic Biology*, **42**(2), 126–141. doi: [10.1093/sysbio/42.2.126](https://doi.org/10.1093/sysbio/42.2.126), http://www.math.canterbury.ac.nz/~m.steel/Non_UC/files/research/distributions.pdf.

Smith MR (2019). “Bayesian and parsimony approaches reconstruct informative trees from simulated morphological datasets.” *Biology Letters*, **15**, 20180632. doi: [10.1098/rsbl.2018.0632](https://doi.org/10.1098/rsbl.2018.0632).

See Also

QuartetStatus: Calculate status of each quartet: the raw material from which the metrics are calculated.

SplitStatus, **CompareSplits**: equivalent metrics for bipartition splits.

Examples

```
data('sq_trees')

sq_status <- QuartetStatus(sq_trees)
SimilarityMetrics(sq_status)
QuartetDivergence(sq_status, similarity=FALSE)
```

SplitStatus

Matching partitions

Description

Calculates how many of the partitions present in tree A are also present in tree B, how many of the partitions in tree A are absent in tree B, and how many of the partitions in tree B are absent in tree A. The Robinson-Foulds (symmetric partition) distance is the sum of the latter two quantities.

Usage

```
SplitStatus(trees, cf = trees[[1]])
```

```
SharedSplitStatus(trees, cf = trees[[1]])
```

Arguments

trees	A list of trees of class <code>phylo</code> , with identically labelled tips.
cf	Comparison tree of class <code>phylo</code> . If unspecified, each tree is compared to the first tree in trees.

Value

Returns a two dimensional array. Rows correspond to the input trees, and are named if names were present. Columns report:

N: The total number of partitions present in the two trees, i.e. $P1 + P2$.

P1: The number of partitions present in tree 1.

P2: The number of partitions present in tree 2.

s: The number of partitions present in both trees.

d1: The number of partitions present in tree 1, but contradicted by tree 2.

d2: The number of partitions present in tree 2, but contradicted by tree 1.

r1: The number of partitions present in tree 1, and neither present nor contradicted in tree 2.

r2: The number of partitions present in tree 2, and neither present nor contradicted in tree 1.

Functions

- `SharedSplitStatus`: Reports split statistics obtained after removing all tips that do not occur in both trees being compared.

Author(s)

Martin R. Smith

References

- Robinson DF, Foulds LR (1981). “Comparison of phylogenetic trees.” *Mathematical Biosciences*, **53**(1-2), 131–147. doi: [10.1016/00255564\(81\)900432](https://doi.org/10.1016/00255564(81)900432).
- Penny D, Hendy MD (1985). “The use of tree comparison metrics.” *Systematic Zoology*, **34**(1), 75–82. doi: [10.2307/2413347](https://doi.org/10.2307/2413347).

See Also

Other element-by-element comparisons: [CompareQuartets](#), [CompareSplits](#), [PairSharedQuartetStatus](#), [QuartetState](#), [SharedQuartetStatus](#)

Examples

```
{
  data('sq_trees')

  # Calculate the status of each quartet
  splitStatuses <- SplitStatus(sq_trees)

  # Calculate the Robinson Foulds distances
  RobinsonFoulds(splitStatuses)

  # Normalize the Robinson Foulds distance by dividing by the number of
  # splits (bipartitions) present in the two trees:
  RobinsonFoulds(splitStatuses) / splitStatuses[, 'N']

  # Normalize the Robinson Foulds distance by dividing by the total number of
  # splits (bipartitions) that it is possible to resolve for `n` tips:
  nTip <- length(sq_trees[[1]]$tip.label)
  nPartitions <- 2 * (nTip - 3L) # Does not include the nTip partitions that
                                # comprise but a single tip
  RobinsonFoulds(splitStatuses) / nPartitions
}
```

sq_trees

Eighteen trees

Description

A list of class `multiPhylo` containing phylogenetic trees:

`ref_tree` A reference tree, bearing tips labelled 1 to 11.

`move_one_near` Tip 1 has been moved a short distance.

`move_one_mid` Tip 1 has been moved further.

`move_one_far` Tip 1 has been moved further still.

move_two_near Tips 10 & 11 have been moved a short distance.
 move_two_mid Tips 10 & 11 have been moved further.
 move_two_far Tips 10 & 11 have been moved further still.
 collapse_one One node has been collapsed into a polytomy.
 collapse_some Several nodes have been collapsed.
 m1mid_col1 Tree move_one_mid with one node collapsed.
 m1mid_colsome Tree move_one_mid with several nodes collapsed.
 m2mid_col1 Tree move_two_mid with one node collapsed.
 m2mid_colsome Tree move_two_mid with several nodes collapsed.
 opposite_tree A tree that shares fewer quartets with ref_tree than expected by chance.
 caterpillar A pectinate 'caterpillar' tree.
 top_and_tail Tree caterpillar, with its outermost taxa swapped such that it shares no partitions with caterpillar.
 anti_pectinate A random tree that shares no partitions with caterpillar.
 random_tree A random tree.

Usage

sq_trees

Format

An object of class multiPhylo of length 18.

SymmetricDifferenceLineEnds

Plot contours of equal symmetric difference on a ternary plot

Description

Assumes that tree 1 is perfectly resolved, but that the resolution of tree 2 can vary.

Usage

SymmetricDifferenceLineEnds(nsd)

SymmetricDifferenceLines(nsd, ...)

Arguments

nsd Vector specifying normalized symmetric differences to plot.

... Further parameters to pass to [TernaryLines](#).

Value

Returns a matrix of dim (length(nsd), 6), with columns named r2a, da, sa, r2b, db and sb. Lines from a to b in each row connect points of equal symmetric difference.

Functions

- `SymmetricDifferenceLines`: Plot the lines onto the active ternary plot.

Author(s)

Martin R. Smith

TQDist	<i>tqDist wrapper</i>
--------	-----------------------

Description

Convenience function that takes a list of trees, writes them to the text file expected by the C implementation of `tqDist` (Sand *et al.* 2014). `tqDist` is then called, and the temporary file is deleted when analysis is complete.

Usage

```
TQDist(treeList)
```

```
TQAE(treeList)
```

```
ManyToManyQuartetAgreement(treeList)
```

```
SingleTreeQuartetAgreement(treeList, comparison)
```

Arguments

`treeList` List of phylogenetic trees, of class `list` or `phylo`. All trees must be bifurcating.
`comparison` A tree of class `phylo` against which to compare the trees in `treeList`.

Details

Quartets can be resolved in one of five ways, which Brodal *et al.* (2013) and Holt *et al.* (2014) distinguish using the letters A–E, and Estabrook (1985) refers to as:

- A: *s* = resolved the **s**ame in both trees;
- B: *d* = resolved **d**ifferently in both trees;
- C: *r1* = resolved only in tree **1**;
- D: *r2* = resolved only in tree **2** (the comparison tree);
- E: *u* = **u**nrresolved in both trees.

Value

TQDist returns the quartet distance between each pair of trees.

TQAE returns the number of resolved quartets in agreement between each pair of trees (A in Brodal *et al.* 2013) and the number of quartets that are unresolved in both trees (E in Brodal *et al.* 2013).

ManyToManyQuartetAgreement returns a three-dimensional array listing, for each pair of trees in turn, the number of quartets in each category.

SingleTreeQuartetAgreement returns a two-dimensional array listing, for tree in treeList, the total number of quartets and the number of quartets in each category. The comparison tree is treated as tree2.

Functions

- TQAE: Number of agreeing quartets that are resolved / unresolved.
- ManyToManyQuartetAgreement: Agreement of each quartet, comparing each pair of trees in a list.
- SingleTreeQuartetAgreement: Agreement of each quartet in trees in a list with the quartets in a comparison tree.

Author(s)

Martin R. Smith

References

Brodal GS, Fagerberg R, Mailund T, Pedersen CNS, Sand A (2013). “Efficient algorithms for computing the triplet and quartet distance between trees of arbitrary degree.” *SODA '13 Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1814–1832. doi: [10.1137/1.9781611973105.130](https://doi.org/10.1137/1.9781611973105.130).

Estabrook GF, McMorris FR, Meacham CA (1985). “Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units.” *Systematic Zoology*, **34**(2), 193–200. doi: [10.2307/2413326](https://doi.org/10.2307/2413326).

Holt MK, Johansen J, Brodal GS (2014). “On the scalability of computing triplet and quartet distances.” In *Proceedings of 16th Workshop on Algorithm Engineering and Experiments (ALENEX) Portland, Oregon, USA*.

Sand A, Holt MK, Johansen J, Brodal GS, Mailund T, Pedersen CNS (2014). “tqDist: a library for computing the quartet and triplet distances between binary or general trees.” *Bioinformatics*, **30**(14), 2079–2080. ISSN 1460-2059, doi: [10.1093/bioinformatics/btu157](https://doi.org/10.1093/bioinformatics/btu157).

Index

*Topic **datasets**

- sq_trees, 19
- AllPairsQuartetAgreement (Distances), 6
- AllPairsQuartetDistance (Distances), 6
- AllPairsTripletDistance (Distances), 6
- AllQuartets, 2, 12
- BipartitionPoints (QuartetPoints), 9
- BipartitionStatus (SplitStatus), 18
- combn, 2, 3
- CompareBipartitions (CompareSplits), 5
- CompareQuartets, 3, 6, 11, 14, 19
- CompareSplits, 4, 5, 11, 14, 17, 19
- Distances, 6
- DoNotConflict (SimilarityMetrics), 15
- ExplicitlyAgree (SimilarityMetrics), 15
- ManyToManyQuartetAgreement (TQDist), 21
- MarczewskiSteinhaus (SimilarityMetrics), 15
- memoise, 2
- multiPhylo, 19
- OneToManyQuartetAgreement (Distances), 6
- PairSharedQuartetStatus, 4, 6, 11, 14, 19
- PairsQuartetDistance (Distances), 6
- PairsTripletDistance (Distances), 6
- par, 8
- phylo, 8–10, 12, 13, 18, 21
- plot, 8
- PlotQuartet, 8
- QuartetAgreement (Distances), 6
- QuartetDistance (Distances), 6
- QuartetDivergence (SimilarityMetrics), 15
- QuartetPoints, 9
- QuartetState, 4, 6, 10, 14, 19
- QuartetStates, 3
- QuartetStates (QuartetState), 10
- QuartetStatus, 3, 4, 17
- QuartetStatus (SharedQuartetStatus), 13
- ResolvedQuartets, 3, 12
- ResolvedTriplets (ResolvedQuartets), 12
- RobinsonFoulds (SimilarityMetrics), 15
- SemiStrictJointAssertions (SimilarityMetrics), 15
- SharedBipartitionStatus (SplitStatus), 18
- SharedQuartetStatus, 4, 6, 11, 13, 19
- SharedSplitStatus (SplitStatus), 18
- SimilarityMetrics, 15
- SingleTreeQuartetAgreement (TQDist), 21
- SplitPoints (QuartetPoints), 9
- SplitStatus, 4, 6, 11, 14, 17, 18
- sq_trees, 19
- SteelPenny (SimilarityMetrics), 15
- StrictJointAssertions (SimilarityMetrics), 15
- SymmetricDifference (SimilarityMetrics), 15
- SymmetricDifferenceLineEnds, 20
- SymmetricDifferenceLines (SymmetricDifferenceLineEnds), 20
- TernaryLines, 20
- TQAE (TQDist), 21
- TQDist, 21
- Tree2Splits, 10
- TripletDistance (Distances), 6