Package ‘Quartet’

January 28, 2020

Version 1.1.0

Title Comparison of Phylogenetic Trees Using Quartet and Split 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 binary 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), TreeTools

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

SystemRequirements C++11

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 2020-01-28 17:00:03 UTC

R topics documented:

`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 (martin.smith@durham.ac.uk)

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` *Compare quartet states by explicit enumeration*

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. 2 Q.

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 (martin.smith@durham.ac.uk)

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.

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))
quartets <- QuartetStates(trees)
CompareQuartets(quartets[[1]], quartets[[2]])
```

`CompareSplits` *Compare status of splits*

Description

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

Usage

```
CompareSplits(splits, splits2)

CompareBipartitions(splits, splits2)
```

Arguments

`splits`	An object that can be coerced into class `Splits` using `as.Splits`.
`splits2`	Splits against which to compare `splits`.

Value

A named vector of eight 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 2 $s + d1 + d2 + r1 + r2$;
- **P1** occur in `splits1`
- **P2** occur in `splits2`
- **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 (martin.smith@durham.ac.uk)

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.

Robinson DF, Foulds LR (1981). "Comparison of phylogenetic trees." *Mathematical Biosciences*, **53**(1-2), 131–147. doi: 10.1016/00255564(81)900432.

See Also

`CompareQuartets`: equivalent function for quartets.

Other element-by-element comparisons: `CompareQuartets`, `PairSharedQuartetStatus`, `QuartetState`, `SharedQuartetStatus`, `SplitStatus`

Examples

```
splits1 <- TreeTools::BalancedTree(8)
splits2 <- TreeTools::PectinateTree(8)

CompareSplits(splits1, splits2)
```

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, possible created using `TQFile`.

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.

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.

See Also

- `QuartetStatus()` takes trees, rather than files, as input.
- `TQFile()` creates a temporary file containing specified trees.

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` `mfrow` and `mar` 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 (martin.smith@durham.ac.uk)

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

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

Details

The ternary plot will depict the number of quartets or splits 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 (martin.smith@durham.ac.uk)

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.

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", "Splits"), 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, splits = 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`	Depreciated; included for compatibility with v1.0.2 and below.
`splits`	An object that can be induced to a `Splits` object using `as.Splits`.

Details

One of the three possible four-taxon trees will be consistent with any set of splits 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 splits is generated from a tree that contains polytomies, it is possible that all three four-taxon trees are consistent with the set of splits

Value

`QuartetState` returns `0` if the relationships of the four taxa are not constrained by the provided splits, or the index of the closest relative to `tips[1]`, otherwise.

Functions

- `QuartetStates`: A convenience wrapper that lists the status of all possible quartets for a given `Splits` object.

Author(s)

Martin R. Smith (martin.smith@durham.ac.uk)

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.

See Also

`CompareQuartets`, used to compare quartet states between trees.

Other element-by-element comparisons: `CompareQuartets`, `CompareSplits`, `PairSharedQuartetStatus`, `SharedQuartetStatus`, `SplitStatus`

Examples

```
{
  nTip <- 6
  trees <- list(ape::rtree(nTip, tip.label=seq_len(nTip), br=NULL),
                ape::rtree(nTip, tip.label=seq_len(nTip), br=NULL))

  trees[[3]] <- TreeTools::CollapseNode(trees[[2]], 9:10)

  QuartetState(c(1, 3, 4, 6), trees[[2]])
  QuartetState(1:4, trees[[1]]) == QuartetState(1:4, trees[[2]])
  QuartetState(c(1, 3, 4, 6), trees[[3]])

  QuartetStates(trees[[2]])
  QuartetStates(trees[[3]])

}
```

ResolvedQuartets *Count resolved quartets*

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 (martin.smith@durham.ac.uk)

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.

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]])

ManyToManyQuartetAgreement(trees)

TwoListQuartetAgreement(trees1, trees2)

SingleTreeQuartetAgreement(trees, comparison)
```

Arguments

`trees`	A list of trees of class `phylo`, with identically labelled tips.
`cf`	Comparison tree of class `phylo`. If unspecified, each tree is compared to the first tree in `trees`.
`trees1, trees2`	List or `multiPhylo` objects containing trees of class `phylo`.
`comparison`	A tree of class `phylo` against which to compare `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 `choose(n_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. 2 Q.

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.

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

`TwoListQuartetAgreement` 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 `trees`, the total number of quartets and the number of quartets in each category. The `comparison` tree is treated as `tree2`.

Functions

- `SharedQuartetStatus`: Reports split statistics obtained after removing all tips that do not occur in both trees being compared.
- `ManyToManyQuartetAgreement`: Agreement of each quartet, comparing each pair of trees in a list.
- `TwoListQuartetAgreement`: Agreement of each quartet in trees in one list with each quartet in trees in a second list.
- `SingleTreeQuartetAgreement`: Agreement of each quartet in trees in a list with the quartets in a comparison tree.

Author(s)

Martin R. Smith (martin.smith@durham.ac.uk)

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.

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.

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.

See Also

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

`SimilarityMetrics`: Generates distance metrics from quartet statuses.

Other element-by-element comparisons: `CompareQuartets`, `CompareSplits`, `PairSharedQuartetStatus`, `QuartetState`, `SplitStatus`

Examples

```
data('sq_trees')
# Calculate the status of each quartet relative to the first entry in
# sq_trees
sq_status <- QuartetStatus(sq_trees)

# Calculate the status of each quartet relative to a given tree
two_moved <- sq_trees[5:7]
sq_status <- QuartetStatus(two_moved, sq_trees$ref_tree)

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

# Calculate Quartet Divergence between each tree and each other tree in a
# list
QuartetDivergence(ManyToManyQuartetAgreement(two_moved))
 # Calculate Quartet Divergence between each tree in one list and each
 # tree in another
 QuartetDivergence(TwoListQuartetAgreement(sq_trees[1:3], sq_trees[10:13]))
```

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 (N, their 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 - *similarity*), and can be calculated algebraically using the identity $N = s + d + r1 + r2 + u$.

Although defined using quartets, analogous values can be calculated using partitions – though for a number of reasons, quartets may offer a more meaningful measure of the amount of information shared by two trees.

- Do Not Conflict (DC): $(s + r1 + r2 + u) / N$
- Explicitly Agree (EA): s / N
- 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 understood, with reference to the text to be a typographic error.)

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

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

Another take on tree similarity is to consider the symmetric difference: that is, the number of partitions or quartets present in one tree that do not appear in the other, originally used to measure tree similarity by Robinson & Foulds (1981). (Note that, given the familiarity of the Robinson Foulds distance metric, this quantity is be default expressed as a difference rather than a similarity.)

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

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

The total number of resolved quartets or partitions present in both trees (Day 1986):

- Symmetric Difference (SD): $(2\ d + r1 + r2) / (2\ d + 2\ s + r1 + r2)$

The total distinctly resolved quartets or partitions (Day 1986):

- Marczewski-Steinhaus (MS): $(2\ d + r1 + r2) / (2\ d + s + r1 + r2)$

The maximum number of quartets or partitions that could have been resolved, given the number of tips (Smith 2019):

- Symmetric Divergence: $(d + d + r1 + r2) / 2\ Q$

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 others return a named vector describing the requested similarity (or difference) between the trees.

Author(s)

Martin R. Smith (martin.smith@durham.ac.uk)

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.

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.

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.

Smith MR (2019). "Bayesian and parsimony approaches reconstruct informative trees from simulated morphological datasets." *Biology Letters*, **15**, 20180632. doi: 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, `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.

See Also

- `QuartetStatus()`: Calculate status of each quartet: the raw material from which the Estabrook *et al.* 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 1 are also present in tree 2 (`s`), how many of the partitions in tree 1 are absent in tree 2 (`d1`), and how many of the partitions in tree 2 are absent in tree 1 (`d2`). The Robinson-Foulds (symmetric partition) distance is the sum of the latter two quantities, i.e. `d1` + `d2`.

Usage

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

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

Arguments

trees	A list of trees of class phylo, with identically labelled tips.
cf	Comparison tree of class phylo. 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 (martin.smith@durham.ac.uk)

References

Robinson DF, Foulds LR (1981). "Comparison of phylogenetic trees." *Mathematical Biosciences*, **53**(1-2), 131–147. doi: 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.

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 present in the two trees:
  RobinsonFoulds(splitStatuses) / splitStatuses[, 'N']

  # Normalize the Robinson Foulds distance by dividing by the total number of
  # splits 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 (martin.smith@durham.ac.uk)

TQDist *tqDist wrapper*

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(trees)

TQAE(trees)
```

Arguments

`trees`	List of phylogenetic trees, of class `list` or `multiPhylo`.

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* = **r**esolved only in tree **1**;
- D: *r2* = **r**esolved only in tree **2** (the comparison tree);
- E: *u* = **u**nresolved 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).

Functions

- `TQAE`: Number of agreeing quartets that are resolved / unresolved.

Author(s)

Martin R. Smith (martin.smith@durham.ac.uk)

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.

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.

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.

See Also

`CompareQuartets()`, `QuartetStatus()`

Index