

# Package ‘TreeSearch’

July 9, 2020

**Title** Phylogenetic Tree Search Using Custom Optimality Criteria

**Version** 0.4.3

**License** GPL (>= 3)

**Copyright** Incorporates C/C++ code from Morphy Phylogenetic Library by Martin Brazeau <<https://github.com/mbrazeau/MorphyLib>> (GPL3)

**Description** Searches for phylogenetic trees that are optimal using a user-defined criterion.  
Handles inapplicable data using the algorithm of Brazeau, Guillerme and Smith (2019) <[doi:10.1093/sysbio/syy083](https://doi.org/10.1093/sysbio/syy083)>. Implements Profile Parsimony (Faith and Trueman, 2001) <[doi:10.1080/10635150118627](https://doi.org/10.1080/10635150118627)>, and Successive Approximations (Farris, 1969) <[doi:10.2307/2412182](https://doi.org/10.2307/2412182)>.

**URL** <https://ms609.github.io/TreeSearch>,  
<https://github.com/ms609/TreeSearch>

**BugReports** <https://github.com/ms609/TreeSearch/issues>

**Depends** R (>= 3.5.0)

**Imports** ape (>= 5.0), phangorn (>= 2.2.1), R.cache, Rdpack, stats, TreeTools (>= 1.0.0)

**Suggests** knitr, rmarkdown, shiny, testthat

**RdMacros** Rdpack

**LazyData** true

**ByteCompile** true

**Encoding** UTF-8

**Language** en-GB

**VignetteBuilder** knitr

**RoxygenNote** 7.1.1

**NeedsCompilation** yes

**Author** Martin R. Smith [aut, cre, cph]  
(<<https://orcid.org/0000-0001-5660-1727>>),  
Martin Brazeau [cph] (<<https://orcid.org/0000-0002-0650-1282>>)

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

**Repository** CRAN

**Date/Publication** 2020-07-09 12:20:06 UTC

## R topics documented:

AllSPR	3
AsBinary	3
CharacterLength	4
congreveLamsdellMatrices	5
Evaluate	6
Fitch	6
ICSteps	7
inapplicable.citations	8
inapplicable.datasets	11
inapplicable.phyData	14
InfoAmounts	17
IWScore	18
IWScoreMorphy	19
Jackknife	21
JackLabels	26
LogisticPoints	28
MinimumLength	28
MorphyBootstrap	29
MorphyWeights	31
NNI	32
PhyDat2Morphy	34
PrepareDataProfile	34
RandomMorphyTree	35
RandomTreeScore	36
RearrangeEdges	36
referenceTree	38
SetMorphyWeights	39
SingleCharMorphy	39
SPR	40
SuccessiveApproximations	42
summary.morphyPtr	43
TBR	44
UnloadMorphy	46
WithOneExtraStep	47
<b>Index</b>	<b>48</b>

---

AllSPR	<i>All SPR trees</i>
--------	----------------------

---

**Description**

All SPR trees

**Usage**

AllSPR(parent, child, nEdge, notDuplicateRoot, edgeToBreak)

**Arguments**

parent	Integer vector corresponding to the first column of the edge matrix of a tree of class <code>phylo</code> , i.e. <code>tree\$edge[, 1]</code> .
child	Integer vector corresponding to the second column of the edge matrix of a tree of class <code>phylo</code> , i.e. <code>tree\$edge[, 2]</code> .
nEdge	integer specifying the number of edges of a tree of class <code>phylo</code> , i.e. <code>dim(tree\$edge)[1]</code>
notDuplicateRoot	logical vector of length nEdge, specifying for each edge whether it is the second edge leading to the root (in which case its breaking will be equivalent to breaking the other root edge... except insofar as it moves the position of the root.)
edgeToBreak	(optional) integer specifying the index of an edge to bisect/prune, generated randomly if not specified. Alternatively, set to -1 to return a complete list of all trees one step from the input tree.

**Value**

AllSPR() returns a list of edge matrices for all trees one SPR rearrangement from the starting tree

**Author(s)**

Martin R. Smith

---

AsBinary	<i>Convert a number to binary</i>
----------	-----------------------------------

---

**Description**

Provides a (reversed) binary representation of a decimal integer

**Usage**

AsBinary(x)

**Arguments**

x                    Decimal integer to be converted to binary bits

**Details**

Provides an array corresponding to binary digits 1, 2, 4, 8, 16, ...

Binary number 0100 (= decimal 4) will be represented as 0 0 1.

**Value**

An array corresponding to binary digits 1, 2, 4, 8, 16, ...

'Leading zeros' are not included.

**Author(s)**

Martin R. Smith, adapted from code posted to R mailing list by Spencer Graves

**Examples**

```
AsBinary(4) # 0 0 1
AsBinary(10) # 0 1 0 1
```

---

CharacterLength	<i>Character length</i>
-----------------	-------------------------

---

**Description**

Homoplasy length of each character in a dataset on a specified tree.

**Usage**

```
CharacterLength(tree, dataset)
```

```
FitchSteps(tree, dataset)
```

```
FastCharacterLength(tree, dataset)
```

**Arguments**

tree                A tree of class [phylo](#).

dataset            A phylogenetic data matrix of class [phyDat](#), whose names correspond to the labels of any accompanying tree.

**Value**

CharacterLength() returns a vector listing the contribution of each character to tree score, according to the algorithm of Brazeau, Guillerme and Smith (2019).

**Functions**

- `FastCharacterLength`: Do not perform checks. Use with care: may cause erroneous results or software crash if variables are in the incorrect format.

**References**

Brazeau MD, Guillaume T, Smith MR (2019). “An algorithm for morphological phylogenetic analysis with inapplicable data.” *Systematic Biology*, **68**, 619–631. doi: [10.1093/sysbio/syy083](https://doi.org/10.1093/sysbio/syy083).

**See Also**

Other tree scoring: [Fitch\(\)](#), [IWScore\(\)](#), [MorphyTreeLength\(\)](#)

**Examples**

```
data('inapplicable.datasets')
dataset <- inapplicable.phyData[[12]]
tree <- TreeTools::NJTree(dataset)
CharacterLength(tree, dataset)
```

---

congreveLamsdellMatrices

*100 simulated data matrices*

---

**Description**

Contains the 100 simulated matrices generated by Congreve & Lamsdell (2016) using a heterogeneous Markov-k model, generated from the [referenceTree](#) topology, with all branches sharing an equal length.

**Usage**

```
congreveLamsdellMatrices
```

**Format**

A list with 100 entries, each comprising a phyDat object of 55 characters for 22 taxa

**Source**

<https://datadryad.org/resource/doi:10.5061/dryad.7dq0j>

**References**

Congreve CR, Lamsdell JC (2016). “Implied weighting and its utility in palaeontological datasets: a study using modelled phylogenetic matrices.” *Palaeontology*, **59**, 447–465. doi: [10.1111/pala.12236](https://doi.org/10.1111/pala.12236).  
Congreve CR, Lamsdell JC (2016). “Data from: Implied weighting and its utility in palaeontological datasets: a study using modelled phylogenetic matrices.” *Dryad Digital Repository*, doi:10.5061/dryad.7dq0j. doi: [10.5061/dryad.7dq0j](https://doi.org/10.5061/dryad.7dq0j).

**Examples**

```

data('referenceTree')
data('congreveLamsdellMatrices')
## Not run: ProfileScore(referenceTree,
  PrepareDataProfile(congreveLamsdellMatrices[[17]]))
## End(Not run)

```

---

Evaluate

*Evaluate tree*

---

**Description**

Evaluate tree

**Usage**

```
Evaluate(tree, dataset, warn = TRUE)
```

**Arguments**

tree	A tree of class <a href="#">phylo</a> .
dataset	A phylogenetic data matrix of class <a href="#">phyDat</a> , whose names correspond to the labels of any accompanying tree.
warn	Boolean (TRUE/FALSE): display warnings when concavity functions are generated by approximation.

---

Fitch

*Calculate parsimony score with inapplicable data*

---

**Description**

Uses code modified from the Morphy library to calculate a parsimony score in datasets that contain inapplicable data.

**Usage**

```
Fitch(tree, dataset)
```

**Arguments**

tree	A tree of class <a href="#">phylo</a> .
dataset	A phylogenetic data matrix of class <a href="#">phyDat</a> , whose names correspond to the labels of any accompanying tree.

**Value**

Fitch() returns the elements from a list containing:

- The total parsimony score
- The parsimony score associated with each character
- A matrix comprising character reconstructions for each node after the final pass

The elements to return are specified by the parameter detail. If a single element is requested (default) then just that element will be returned. If multiple elements are requested then these will be returned in a list.

**Author(s)**

Martin R. Smith (using Morphy C library, by Martin Brazeau)

**See Also**

- [TreeSearch\(\)](#)

Other tree scoring: [CharacterLength\(\)](#), [IWScore\(\)](#), [MorphyTreeLength\(\)](#)

**Examples**

```
data("inapplicable.datasets")
tree <- TreeTools::BalancedTree(inapplicable.phyData[[1]])
Fitch(tree, inapplicable.phyData[[1]])
```

---

ICSteps

*Information Content Steps*

---

**Description**

This function estimates the information content of a character char when e extra steps are present, for all possible values of e.

**Usage**

```
ICSteps(  
  char,  
  ambiguousToken = 0,  
  expectedMinima = 25L,  
  maxIter = 10000L,  
  warn = TRUE  
)
```

**Arguments**

char	The character in question.
ambiguousToken	Which token, if any, corresponds to the ambiguous token (?) (not yet fully implemented).
expectedMinima	sample enough trees that the rarest step counts is expected to be seen at least this many times.
maxIter	Maximum iterations to conduct.
warn	Boolean (TRUE/FALSE): display warnings when concavity functions are generated by approximation.

**Details**

Calculates the number of trees consistent with the character having  $e$  extra steps, where  $e$  ranges from its minimum possible value (i.e. number of different tokens minus one) to its maximum. The number of trees with no extra steps can be calculated exactly; the number of trees with more additional steps must be approximated. The function samples `n.iter` trees, or enough trees that the trees with the minimum number of steps will be recovered at least `expected.minima` times, in order to obtain precise results.

**Author(s)**

Martin R. Smith ([martin.smith@durham.ac.uk](mailto:martin.smith@durham.ac.uk))

**References**

Faith DP, Trueman JWH (2001). “Towards an inclusive philosophy for phylogenetic inference.” *Systematic Biology*, **50**(3), 331–350. doi: [10.1080/10635150118627](https://doi.org/10.1080/10635150118627).

**Examples**

```
# A character that is present in ten taxa and absent in five
character <- c(rep(1, 10), rep(2, 5))
suppressWarnings(ICSteps(character))
```

---

inapplicable.citations

*Thirty Datasets with Inapplicable data*

---

**Description**

These are the datasets used to evaluate the behaviour of the inapplicable algorithm in Brazeau, Guillerme and Smith (2018).

**Usage**

```
inapplicable.citations
```



**Format**

An object of class character of length 30.

**Details**

The name of each item corresponds to the datasets listed below. The value gives its citation.

**Source**

- Agnarsson2004** AGNARSSON, I. 2004. Morphological phylogeny of cobweb spiders and their relatives (Araneae, Araneoidea, Theridiidae). *Zoological Journal of the Linnean Society*, 141, 447–626.
- Capa2011** CAPA, M., HUTCHINGS, P., AGUADO, M. T. and BOTT, N. J. 2011. Phylogeny of Sabellidae (Annelida) and relationships with other taxa inferred from morphology and multiple genes. *Cladistics*, 27, 449–469.
- DeAssis2011** DE ASSIS, J. E. and CHRISTOFFERSEN, M. L. 2011. Phylogenetic relationships within Maldanidae (Capitellida, Annelida), based on morphological characters. *Systematics and Biodiversity*, 9, 233–245.
- OLeary1999** O’LEARY, M. A. and GEISLER, J. H. 1999. The position of Cetacea within Mammalia: phylogenetic analysis of morphological data from extinct and extant taxa. *Systematic Biology*, 48, 455–490.
- Rousset2004** ROUSSET, V., ROUSE, G. W., SIDDALL, M. E., TILLIER, A. and PLEIJEL, F. 2004. The phylogenetic position of Siboglinidae (Annelida) inferred from 18S rRNA, 28S rRNA and morphological data. *Cladistics*, 20, 518–533.
- Sano2011** SANO, M. and AKIMOTO, S.-I. 2011. Morphological phylogeny of gall-forming aphids of the tribe Eriosomatini (Aphididae: Eriosomatinae). *Systematic Entomology*, 36, 607–627.
- Sansom2010** SANSOM, R. S., FREEDMAN, K., GABBOTT, S. E., ALDRIDGE, R. J. and PURNELL, M. A. 2010. Taphonomy and affinity of an enigmatic Silurian vertebrate, *Jamoytius kerwoodi* White. *Palaeontology*, 53, 1393–1409.
- Schulze2007** SCHULZE, A., CUTLER, E. B. and GIRIBET, G. 2007. Phylogeny of sipunculan worms: A combined analysis of four gene regions and morphology. *Molecular Phylogenetics and Evolution*, 42, 171–92.
- Shultz2007** SHULTZ, J. W. 2007. A phylogenetic analysis of the arachnid orders based on morphological characters. *Zoological Journal of the Linnean Society*, 150, 221–265.
- Wetterer2000** WETTERER, A. L., ROCKKMAN, M. V. and SIMMONS, N. B. 2000. Phylogeny of phyllostomid bats (Mammalia: Chiroptera): data from diverse morphological systems, sex chromosomes, and restriction sites. *Bulletin of the American Museum of Natural History*, 248, 1–200.
- Wills2012** WILLS, M. A., GERBER, S., RUTA, M. and HUGHES, M. 2012. The disparity of priapulid, archaeopriapulid and palaeoscolecid worms in the light of new data. *Journal of Evolutionary Biology*, 25, 2056–2076.
- Aguado2009** AGUADO, M. T. and SAN MARTIN, G. 2009. Phylogeny of Syllidae (Polychaeta) based on morphological data. *Zoologica Scripta*, 38, 379–402.
- Aria2015** ARIA, C., CARON, J. B. and GAINES, R. 2015. A large new leancoiliid from the Burgess Shale and the influence of inapplicable states on stem arthropod phylogeny. *Palaeontology*, 58, 629–660.

- Asher2005** ASHER, R. J. and HOFREITER, M. 2006. Tenrec phylogeny and the noninvasive extraction of nuclear DNA. *Systematic biology*, 55, 181–94.
- Baker2009** BAKER, W. J., SAVOLAINEN, V., ASMUSSEN-LANGE, C. B., CHASE, M. W., DRANSFIELD, J., FOREST, F., HARLEY, M. M., UHL, N. W. and WILKINSON, M. 2009. Complete generic-level phylogenetic analyses of palms (Arecaceae) with comparisons of supertree and supermatrix approaches. *Systematic Biology*, 58, 240–256.
- Bouchenak2010** BOUCHENAK-KHELLADI, Y., VERBOOM, G. A., SAVOLAINEN, V. and HODKINSON, T. R. 2010. Biogeography of the grasses (Poaceae): a phylogenetic approach to reveal evolutionary history in geographical space and geological time. *Botanical Journal of the Linnean Society*, 162, 543–557.
- Conrad2008** CONRAD, J. L. 2008. Phylogeny And Systematics Of Squamata (Reptilia) Based On Morphology. *Bulletin of the American Museum of Natural History*, 310, 1–182.
- Dikow2009** DIKOW, T. 2009. A phylogenetic hypothesis for Asilidae based on a total evidence analysis of morphological and DNA sequence data (Insecta: Diptera: Brachycera: Asiloidea). *Organisms Diversity and Evolution*, 9, 165–188.
- Eklund2004** EKLUND, H., DOYLE, J. A. and HERENDEEN, P. S. 2004. Morphological phylogenetic analysis of living and fossil Chloranthaceae. *International Journal of Plant Sciences*, 165, 107–151.
- Geisler2001** GEISLER, J. H. 2001. New morphological evidence for the phylogeny of Artiodactyla, Cetacea, and Mesonychidae. *American Museum Novitates*, 3344, 53.
- Giles2015** GILES, S., FRIEDMAN, M. and BRAZEAU, M. D. 2015. Osteichthyan-like cranial conditions in an Early Devonian stem gnathostome. *Nature*, 520, 82–85.
- Griswold1999** GRISWOLD, C. E., CODDINGTON, J. A., PLATNICK, N. I. and FORSTER, R. R. 1999. Towards a phylogeny of entelegyne spiders (Araneae, Araneomorphae, Entelegynae). *Journal of Arachnology*, 27, 53–63.
- Liljeblad2008** LILJEBLAD, J., RONQUIST, F., NIEVES-ALDREY, J. L., FONTAL-CAZALLA, F., ROS-FARRE, P., GAITROS, D. and PUJADE-VILLAR, J. 2008. A fully web-illustrated morphological phylogenetic study of relationships among oak gall wasps and their closest relatives (Hymenoptera: Cynipidae).
- Loconte1991** LOCONTE, H. and STEVENSON, D. W. 1991. Cladistics of the Magnoliidae. *Cladistics*, 7, 267–296.
- Longrich2010** LONGRICH, N. R., SANKEY, J. and TANKE, D. 2010. *Texacephale langstoni*, a new genus of pachycephalosaurid (Dinosauria: Ornithischia) from the upper Campanian Aguja Formation, southern Texas, USA. *Cretaceous Research*, 31, 274–284.
- OMeara2014** O’MEARA, R. N. and THOMPSON, R. S. 2014. Were There Miocene Meridiolestidans? Assessing the phylogenetic placement of *Necrolestes patagonensis* and the presence of a 40 million year Meridiolestidan ghost lineage. *Journal of Mammalian Evolution*, 21, 271–284.
- Rougier2012** ROUGIER, G. W., WIBLE, J. R., BECK, R. M. D. and APESTEGUIA, S. 2012. The Miocene mammal *Necrolestes* demonstrates the survival of a Mesozoic nontherian lineage into the late Cenozoic of South America. *Proceedings of the National Academy of Sciences*, 109, 20053–8.
- Sharkey2011** SHARKEY, M. J., CARPENTER, J. M., VILHELMSSEN, L., HERATY, J., LILJEBLAD, J., DOWLING, A. P. G., SCHULMEISTER, S., MURRAY, D., DEANS, A. R., RONQUIST, F., KROGMANN, L. and WHEELER, W. C. 2012. Phylogenetic relationships among superfamilies of Hymenoptera. *Cladistics*, 28, 80–112.

- Sundue2010** SUNDUE, M. A., ISLAM, M. B. and RANKER, T. A. 2010. Systematics of Grammitid Ferns (Polypodiaceae): Using Morphology and Plastid Sequence Data to Resolve the Circumscriptions of Melpomene and the Polyphyletic Genera *Lellingeria* and *Terpsichore*. *Systematic Botany*, 35, 701–715.
- Vinther2008** VINTHER, J., VAN ROY, P. and BRIGGS, D. E. G. 2008. Machaeridians are Palaeozoic armoured annelids. *Nature*, 451, 185–188.
- Wilson2003** WILSON, G. D. F. and EDGECOMBE, G. D. 2003. The Triassic isopod *Protamphisopus wianamattensis* (Chilton) and comparison by extant taxa (Crustacea, Phreatoicoidea). *Journal of Paleontology*, 77, 454–470.
- Wortley2006** WORTLEY, A. H. and SCOTLAND, R. W. 2006. The effect of combining molecular and morphological data in published phylogenetic analyses. *Systematic Biology*, 55, 677–685.
- Zanol2014** ZANOL, J., HALANYCH, K. M. and FAUCHALD, K. 2014. Reconciling taxonomy and phylogeny in the bristleworm family Eunicidae (Polychaete, Annelida). *Zoologica Scripta*, 43, 79–100.
- Zhu2013** ZHU, M., YU, X., AHLBERG, P. E., CHOO, B., LU, J., QIAO, T., QU, Q., ZHAO, W., JIA, L., BLOM, H. and ZHU, Y. 2013. A Silurian placoderm with osteichthyan-like marginal jaw bones. *Nature*, 502, 188–193.

## References

- Brazeau MD, Guillaume T, Smith MR (2019). “An algorithm for morphological phylogenetic analysis with inapplicable data.” *Systematic Biology*, 68, 619–631. doi: [10.1093/sysbio/syy083](https://doi.org/10.1093/sysbio/syy083).

---

inapplicable.datasets *Thirty Datasets with Inapplicable data*

---

## Description

These are the datasets used to evaluate the behaviour of the inapplicable algorithm in Brazeau, Guillaume and Smith (2017).

## Usage

inapplicable.datasets

## Format

An object of class list of length 30.

## Details

The name of each item corresponds to the datasets listed below. Datasets are sorted into two subsets, each sorted alphabetically; the first subset comprise simpler datasets with faster processing times. The value is the dataset in the format generated by [read.nexus.data](#).

**Source**

- Agnarsson2004** AGNARSSON, I. 2004. Morphological phylogeny of cobweb spiders and their relatives (Araneae, Araneoidea, Theridiidae). *Zoological Journal of the Linnean Society*, 141, 447–626.
- Capa2011** CAPA, M., HUTCHINGS, P., AGUADO, M. T. and BOTT, N. J. 2011. Phylogeny of Sabellidae (Annelida) and relationships with other taxa inferred from morphology and multiple genes. *Cladistics*, 27, 449–469.
- DeAssis2011** DE ASSIS, J. E. and CHRISTOFFERSEN, M. L. 2011. Phylogenetic relationships within Maldanidae (Capitellida, Annelida), based on morphological characters. *Systematics and Biodiversity*, 9, 233–245.
- OLeary1999** O’LEARY, M. A. and GEISLER, J. H. 1999. The position of Cetacea within Mammalia: phylogenetic analysis of morphological data from extinct and extant taxa. *Systematic Biology*, 48, 455–490.
- Rousset2004** ROUSSET, V., ROUSE, G. W., SIDDALL, M. E., TILLIER, A. and PLEIJEL, F. 2004. The phylogenetic position of Siboglinidae (Annelida) inferred from 18S rRNA, 28S rRNA and morphological data. *Cladistics*, 20, 518–533.
- Sano2011** SANO, M. and AKIMOTO, S.-I. 2011. Morphological phylogeny of gall-forming aphids of the tribe Eriosomatini (Aphididae: Eriosomatinae). *Systematic Entomology*, 36, 607–627.
- Sansom2010** SANSOM, R. S., FREEDMAN, K., GABBOTT, S. E., ALDRIDGE, R. J. and PURNELL, M. A. 2010. Taphonomy and affinity of an enigmatic Silurian vertebrate, *Jamoytius kerwoodi* White. *Palaeontology*, 53, 1393–1409.
- Schulze2007** SCHULZE, A., CUTLER, E. B. and GIRIBET, G. 2007. Phylogeny of sipunculan worms: A combined analysis of four gene regions and morphology. *Molecular Phylogenetics and Evolution*, 42, 171–92.
- Shultz2007** SHULTZ, J. W. 2007. A phylogenetic analysis of the arachnid orders based on morphological characters. *Zoological Journal of the Linnean Society*, 150, 221–265.
- Wetterer2000** WETTERER, A. L., ROCKKMAN, M. V. and SIMMONS, N. B. 2000. Phylogeny of phyllostomid bats (Mammalia: Chiroptera): data from diverse morphological systems, sex chromosomes, and restriction sites. *Bulletin of the American Museum of Natural History*, 248, 1–200.
- Wills2012** WILLS, M. A., GERBER, S., RUTA, M. and HUGHES, M. 2012. The disparity of priapulid, archaeopriapulid and palaeoscolecoid worms in the light of new data. *Journal of Evolutionary Biology*, 25, 2056–2076.
- Aguado2009** AGUADO, M. T. and SAN MARTIN, G. 2009. Phylogeny of Syllidae (Polychaeta) based on morphological data. *Zoologica Scripta*, 38, 379–402.
- Aria2015** ARIA, C., CARON, J. B. and GAINES, R. 2015. A large new leanchoiliid from the Burgess Shale and the influence of inapplicable states on stem arthropod phylogeny. *Palaeontology*, 58, 629–660.
- Asher2005** ASHER, R. J. and HOFREITER, M. 2006. Tenrec phylogeny and the noninvasive extraction of nuclear DNA. *Systematic biology*, 55, 181–94.
- Baker2009** BAKER, W. J., SAVOLAINEN, V., ASMUSSEN-LANGE, C. B., CHASE, M. W., DRANSFIELD, J., FOREST, F., HARLEY, M. M., UHL, N. W. and WILKINSON, M. 2009. Complete generic-level phylogenetic analyses of palms (Arecaceae) with comparisons of supertree and supermatrix approaches. *Systematic Biology*, 58, 240–256.

- Bouchenak2010** BOUCHENAK-KHELLADI, Y., VERBOOM, G. A., SAVOLAINEN, V. and HODKINSON, T. R. 2010. Biogeography of the grasses (Poaceae): a phylogenetic approach to reveal evolutionary history in geographical space and geological time. *Botanical Journal of the Linnean Society*, 162, 543–557.
- Conrad2008** CONRAD, J. L. 2008. Phylogeny And Systematics Of Squamata (Reptilia) Based On Morphology. *Bulletin of the American Museum of Natural History*, 310, 1–182.
- Dikow2009** DIKOW, T. 2009. A phylogenetic hypothesis for Asilidae based on a total evidence analysis of morphological and DNA sequence data (Insecta: Diptera: Brachycera: Asiloidea). *Organisms Diversity and Evolution*, 9, 165–188.
- Eklund2004** EKLUND, H., DOYLE, J. A. and HERENDEEN, P. S. 2004. Morphological phylogenetic analysis of living and fossil Chloranthaceae. *International Journal of Plant Sciences*, 165, 107–151.
- Geisler2001** GEISLER, J. H. 2001. New morphological evidence for the phylogeny of Artiodactyla, Cetacea, and Mesonychidae. *American Museum Novitates*, 3344, 53.
- Giles2015** GILES, S., FRIEDMAN, M. and BRAZEAU, M. D. 2015. Osteichthyan-like cranial conditions in an Early Devonian stem gnathostome. *Nature*, 520, 82–85.
- Griswold1999** GRISWOLD, C. E., CODDINGTON, J. A., PLATNICK, N. I. and FORSTER, R. R. 1999. Towards a phylogeny of entelegyne spiders (Araneae, Araneomorphae, Entelegynae). *Journal of Arachnology*, 27, 53–63.
- Liljeblad2008** LILJEBLAD, J., RONQUIST, F., NIEVES-ALDREY, J. L., FONTAL-CAZALLA, F., ROS-FARRE, P., GAITROS, D. and PUJADE-VILLAR, J. 2008. A fully web-illustrated morphological phylogenetic study of relationships among oak gall wasps and their closest relatives (Hymenoptera: Cynipidae).
- Loconte1991** LOCONTE, H. and STEVENSON, D. W. 1991. Cladistics of the Magnoliidae. *Cladistics*, 7, 267–296.
- Longrich2010** LONGRICH, N. R., SANKEY, J. and TANKE, D. 2010. *Texacephale langstoni*, a new genus of pachycephalosaurid (Dinosauria: Ornithischia) from the upper Campanian Aguja Formation, southern Texas, USA. *Cretaceous Research*, 31, 274–284.
- OMeara2014** O’MEARA, R. N. and THOMPSON, R. S. 2014. Were There Miocene Meridiolestidans? Assessing the phylogenetic placement of *Necrolestes patagonensis* and the presence of a 40 million year Meridiolestidan ghost lineage. *Journal of Mammalian Evolution*, 21, 271–284.
- Rougier2012** ROUGIER, G. W., WIBLE, J. R., BECK, R. M. D. and APESTEGUIA, S. 2012. The Miocene mammal *Necrolestes* demonstrates the survival of a Mesozoic nontherian lineage into the late Cenozoic of South America. *Proceedings of the National Academy of Sciences*, 109, 20053–8.
- Sharkey2011** SHARKEY, M. J., CARPENTER, J. M., VILHELMSSEN, L., HERATY, J., LILJEBLAD, J., DOWLING, A. P. G., SCHULMEISTER, S., MURRAY, D., DEANS, A. R., RONQUIST, F., KROGMANN, L. and WHEELER, W. C. 2012. Phylogenetic relationships among superfamilies of Hymenoptera. *Cladistics*, 28, 80–112.
- Sundue2010** SUNDUE, M. A., ISLAM, M. B. and RANKER, T. A. 2010. Systematics of Grammitid Ferns (Polypodiaceae): Using Morphology and Plastid Sequence Data to Resolve the Circumscriptions of *Melpomene* and the Polyphyletic Genera *Lellingeria* and *Terpsichore*. *Systematic Botany*, 35, 701–715.

- Vinther2008** VINTHER, J., VAN ROY, P. and BRIGGS, D. E. G. 2008. Machaeridians are Palaeozoic armoured annelids. *Nature*, 451, 185–188.
- Wilson2003** WILSON, G. D. F. and EDGEcombe, G. D. 2003. The Triassic isopod *Protamphisopus wianamattensis* (Chilton) and comparison by extant taxa (Crustacea, Phreatoicoidea). *Journal of Paleontology*, 77, 454–470.
- Wortley2006** WORTLEY, A. H. and SCOTLAND, R. W. 2006. The effect of combining molecular and morphological data in published phylogenetic analyses. *Systematic Biology*, 55, 677–685.
- Zanol2014** ZANOL, J., HALANYCH, K. M. and FAUCHALD, K. 2014. Reconciling taxonomy and phylogeny in the bristleworm family Eunicidae (Polychaete, Annelida). *Zoologica Scripta*, 43, 79–100.
- Zhu2013** ZHU, M., YU, X., AHLBERG, P. E., CHOO, B., LU, J., QIAO, T., QU, Q., ZHAO, W., JIA, L., BLOM, H. and ZHU, Y. 2013. A Silurian placoderm with osteichthyan-like marginal jaw bones. *Nature*, 502, 188–193.

## References

- Brazeau MD, Guillaume T, Smith MR (2019). “An algorithm for morphological phylogenetic analysis with inapplicable data.” *Systematic Biology*, **68**, 619–631. doi: [10.1093/sysbio/syy083](https://doi.org/10.1093/sysbio/syy083).

---

inapplicable.phyData *Thirty Datasets with Inapplicable data*

---

## Description

These are the datasets used to evaluate the behaviour of the inapplicable algorithm in Brazeau, Guillaume and Smith (2017).

## Usage

```
inapplicable.phyData
```

## Format

An object of class `list` of length 30.

## Details

The name of each item corresponds to the datasets listed below. Datasets are sorted into two subsets, each sorted alphabetically; the first subset comprise simpler datasets with faster processing times. The value is the dataset in `phyDat` format.

**Source**

- Agnarsson2004** AGNARSSON, I. 2004. Morphological phylogeny of cobweb spiders and their relatives (Araneae, Araneoidea, Theridiidae). *Zoological Journal of the Linnean Society*, 141, 447–626.
- Capa2011** CAPA, M., HUTCHINGS, P., AGUADO, M. T. and BOTT, N. J. 2011. Phylogeny of Sabellidae (Annelida) and relationships with other taxa inferred from morphology and multiple genes. *Cladistics*, 27, 449–469.
- DeAssis2011** DE ASSIS, J. E. and CHRISTOFFERSEN, M. L. 2011. Phylogenetic relationships within Maldanidae (Capitellida, Annelida), based on morphological characters. *Systematics and Biodiversity*, 9, 233–245.
- OLeary1999** O'LEARY, M. A. and GEISLER, J. H. 1999. The position of Cetacea within Mammalia: phylogenetic analysis of morphological data from extinct and extant taxa. *Systematic Biology*, 48, 455–490.
- Rousset2004** ROUSSET, V., ROUSE, G. W., SIDDALL, M. E., TILLIER, A. and PLEIJEL, F. 2004. The phylogenetic position of Siboglinidae (Annelida) inferred from 18S rRNA, 28S rRNA and morphological data. *Cladistics*, 20, 518–533.
- Sano2011** SANO, M. and AKIMOTO, S.-I. 2011. Morphological phylogeny of gall-forming aphids of the tribe Eriosomatini (Aphididae: Eriosomatinae). *Systematic Entomology*, 36, 607–627.
- Sansom2010** SANSOM, R. S., FREEDMAN, K., GABBOTT, S. E., ALDRIDGE, R. J. and PURNELL, M. A. 2010. Taphonomy and affinity of an enigmatic Silurian vertebrate, *Jamoytius kerwoodi* White. *Palaeontology*, 53, 1393–1409.
- Schulze2007** SCHULZE, A., CUTLER, E. B. and GIRIBET, G. 2007. Phylogeny of sipunculan worms: A combined analysis of four gene regions and morphology. *Molecular Phylogenetics and Evolution*, 42, 171–92.
- Shultz2007** SHULTZ, J. W. 2007. A phylogenetic analysis of the arachnid orders based on morphological characters. *Zoological Journal of the Linnean Society*, 150, 221–265.
- Wetterer2000** WETTERER, A. L., ROCKKMAN, M. V. and SIMMONS, N. B. 2000. Phylogeny of phyllostomid bats (Mammalia: Chiroptera): data from diverse morphological systems, sex chromosomes, and restriction sites. *Bulletin of the American Museum of Natural History*, 248, 1–200.
- Wills2012** WILLS, M. A., GERBER, S., RUTA, M. and HUGHES, M. 2012. The disparity of priapulid, archaeopriapulid and palaeoscolecid worms in the light of new data. *Journal of Evolutionary Biology*, 25, 2056–2076.
- Aguado2009** AGUADO, M. T. and SAN MARTIN, G. 2009. Phylogeny of Syllidae (Polychaeta) based on morphological data. *Zoologica Scripta*, 38, 379–402.
- Aria2015** ARIA, C., CARON, J. B. and GAINES, R. 2015. A large new leanchoiliid from the Burgess Shale and the influence of inapplicable states on stem arthropod phylogeny. *Palaeontology*, 58, 629–660.
- Asher2005** ASHER, R. J. and HOFREITER, M. 2006. Tenrec phylogeny and the noninvasive extraction of nuclear DNA. *Systematic biology*, 55, 181–94.
- Baker2009** BAKER, W. J., SAVOLAINEN, V., ASMUSSEN-LANGE, C. B., CHASE, M. W., DRANSFIELD, J., FOREST, F., HARLEY, M. M., UHL, N. W. and WILKINSON, M. 2009. Complete generic-level phylogenetic analyses of palms (Arecaceae) with comparisons of supertree and supermatrix approaches. *Systematic Biology*, 58, 240–256.

- Bouchenak2010** BOUCHENAK-KHELLADI, Y., VERBOOM, G. A., SAVOLAINEN, V. and HODKINSON, T. R. 2010. Biogeography of the grasses (Poaceae): a phylogenetic approach to reveal evolutionary history in geographical space and geological time. *Botanical Journal of the Linnean Society*, 162, 543–557.
- Conrad2008** CONRAD, J. L. 2008. Phylogeny And Systematics Of Squamata (Reptilia) Based On Morphology. *Bulletin of the American Museum of Natural History*, 310, 1–182.
- Dikow2009** DIKOW, T. 2009. A phylogenetic hypothesis for Asilidae based on a total evidence analysis of morphological and DNA sequence data (Insecta: Diptera: Brachycera: Asiloidea). *Organisms Diversity and Evolution*, 9, 165–188.
- Eklund2004** EKLUND, H., DOYLE, J. A. and HERENDEEN, P. S. 2004. Morphological phylogenetic analysis of living and fossil Chloranthaceae. *International Journal of Plant Sciences*, 165, 107–151.
- Geisler2001** GEISLER, J. H. 2001. New morphological evidence for the phylogeny of Artiodactyla, Cetacea, and Mesonychidae. *American Museum Novitates*, 3344, 53.
- Giles2015** GILES, S., FRIEDMAN, M. and BRAZEAU, M. D. 2015. Osteichthyan-like cranial conditions in an Early Devonian stem gnathostome. *Nature*, 520, 82–85.
- Griswold1999** GRISWOLD, C. E., CODDINGTON, J. A., PLATNICK, N. I. and FORSTER, R. R. 1999. Towards a phylogeny of entelegyne spiders (Araneae, Araneomorphae, Entelegynae). *Journal of Arachnology*, 27, 53–63.
- Liljeblad2008** LILJEBLAD, J., RONQUIST, F., NIEVES-ALDREY, J. L., FONTAL-CAZALLA, F., ROS-FARRE, P., GAITROS, D. and PUJADE-VILLAR, J. 2008. A fully web-illustrated morphological phylogenetic study of relationships among oak gall wasps and their closest relatives (Hymenoptera: Cynipidae).
- Loconte1991** LOCONTE, H. and STEVENSON, D. W. 1991. Cladistics of the Magnoliidae. *Cladistics*, 7, 267–296.
- Longrich2010** LONGRICH, N. R., SANKEY, J. and TANKE, D. 2010. *Texacephale langstoni*, a new genus of pachycephalosaurid (Dinosauria: Ornithischia) from the upper Campanian Aguja Formation, southern Texas, USA. *Cretaceous Research*, 31, 274–284.
- OMeara2014** O’MEARA, R. N. and THOMPSON, R. S. 2014. Were There Miocene Meridiolestidans? Assessing the phylogenetic placement of *Necrolestes patagonensis* and the presence of a 40 million year Meridiolestidan ghost lineage. *Journal of Mammalian Evolution*, 21, 271–284.
- Rougier2012** ROUGIER, G. W., WIBLE, J. R., BECK, R. M. D. and APESTEGUIA, S. 2012. The Miocene mammal *Necrolestes* demonstrates the survival of a Mesozoic nontherian lineage into the late Cenozoic of South America. *Proceedings of the National Academy of Sciences*, 109, 20053–8.
- Sharkey2011** SHARKEY, M. J., CARPENTER, J. M., VILHELMSSEN, L., HERATY, J., LILJEBLAD, J., DOWLING, A. P. G., SCHULMEISTER, S., MURRAY, D., DEANS, A. R., RONQUIST, F., KROGMANN, L. and WHEELER, W. C. 2012. Phylogenetic relationships among superfamilies of Hymenoptera. *Cladistics*, 28, 80–112.
- Sundue2010** SUNDUE, M. A., ISLAM, M. B. and RANKER, T. A. 2010. Systematics of Grammitid Ferns (Polypodiaceae): Using Morphology and Plastid Sequence Data to Resolve the Circumscriptions of *Melpomene* and the Polyphyletic Genera *Lellingeria* and *Terpsichore*. *Systematic Botany*, 35, 701–715.



- Vinther2008** VINTHER, J., VAN ROY, P. and BRIGGS, D. E. G. 2008. Machaeridians are Palaeozoic armoured annelids. *Nature*, 451, 185–188.
- Wilson2003** WILSON, G. D. F. and EDGECOMBE, G. D. 2003. The Triassic isopod *Protamphisopus wianamattensis* (Chilton) and comparison by extant taxa (Crustacea, Phreatoicoidea). *Journal of Paleontology*, 77, 454–470.
- Wortley2006** WORTLEY, A. H. and SCOTLAND, R. W. 2006. The effect of combining molecular and morphological data in published phylogenetic analyses. *Systematic Biology*, 55, 677–685.
- Zanol2014** ZANOL, J., HALANYCH, K. M. and FAUCHALD, K. 2014. Reconciling taxonomy and phylogeny in the bristleworm family Eunicidae (Polychaete, Annelida). *Zoologica Scripta*, 43, 79–100.
- Zhu2013** ZHU, M., YU, X., AHLBERG, P. E., CHOO, B., LU, J., QIAO, T., QU, Q., ZHAO, W., JIA, L., BLOM, H. and ZHU, Y. 2013. A Silurian placoderm with osteichthyan-like marginal jaw bones. *Nature*, 502, 188–193.

## References

- Brazeau MD, Guillaume T, Smith MR (2019). “An algorithm for morphological phylogenetic analysis with inapplicable data.” *Systematic Biology*, 68, 619–631. doi: [10.1093/sysbio/syy083](https://doi.org/10.1093/sysbio/syy083).

---

InfoAmounts

*Amount of information in each character*

---

## Description

As presently implemented, this function requires that there be no ambiguous tokens and two applicable tokens, '1' and '2'.

## Usage

```
InfoAmounts(tokenTable, precision = 1e+06, warn = TRUE)
```

## Arguments

tokenTable	A matrix, where each row corresponds to a character, each column to a tip, and each entry to the value (1 or 2) of the character at that tip.
precision	number of random trees to generate when calculating Profile curves
warn	Boolean (TRUE/FALSE): display warnings when concavity functions are generated by approximation.

## Value

information content of each extra step, in bits

## Author(s)

Martin R. Smith

---

IWScore

*Implied weights parsimony Score*


---

**Description**

Calculate a tree's Parsimony score with a given dataset using implied weights (Goloboff 1997).

**Usage**

```
IWScore(tree, dataset, concavity = 10, ...)
```

```
IWInitMorpho(dataset)
```

```
IWDestroyMorpho(dataset)
```

**Arguments**

tree	A tree of class <code>phylo</code> .
dataset	Dataset of class <code>phyDat</code> . The dataset should have been prepared using <code>dataset &lt;- PrepareDataIW(dataset)</code> ; if this step has not been completed, the dataset will be (time-consumingly) prepared within the function. In subsidiary functions, the dataset will have been initialized using <code>IWInitMorpho</code> , and must be destroyed using <code>IWDestroyMorpho</code> .
concavity	A numeric value to use as the concavity constant (k) in implied weighting. Defaults to 10; TNT sets a default of 3, but this is too low in some circumstances (Smith, 2019).
...	unused; allows additional parameters specified within ... to be received by the function without throwing an error.

**Value**

The 'fit',  $h / h + k$ , where h is the amount of homoplasy ('extra steps') and k is a constant (the 'concavity constant')

**Functions**

- `IWInitMorpho`: Initialize dataset by adding `morphoObjs` and `min.length` properties.
- `IWDestroyMorpho`: Free memory from `morphoObjs` initialized by `IWScoreMorpho()`.

**Author(s)**

Martin R. Smith

## References

- Goloboff PA (1997). “Self-weighted optimization: tree searches and character state reconstructions under implied transformation costs.” *Cladistics*, **13**(3), 225–245. <http://dx.doi.org/10.1111/j.1096-0031.1997.tb00317.x>.
- 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

Other tree scoring: [CharacterLength\(\)](#), [Fitch\(\)](#), [MorphyTreeLength\(\)](#)

## Examples

```
data(referenceTree)
data(congreveLamsdellMatrices)
dataset <- PrepareDataIW(congreveLamsdellMatrices[[42]])
IWScore(referenceTree, dataset)
```

---

IWScoreMorphy

*Profile Parsimony Score*

---

## Description

Calculate a tree’s Profile Parsimony score with a given dataset, after Faith and Trueman (2001).

## Usage

```
IWScoreMorphy(
  parent,
  child,
  dataset,
  concavity = 10L,
  minLength = attr(dataset, "min.length"),
  ...
)

ProfileScore(tree, dataset)

ProfileScoreMorphy(parent, child, dataset, ...)

ProfileInitMorphy(dataset)

ProfileDestroyMorphy(dataset)
```

**Arguments**

parent	Integer vector corresponding to the first column of the edge matrix of a tree of class <code>phylo</code> , i.e. <code>tree\$edge[, 1]</code> .
child	Integer vector corresponding to the second column of the edge matrix of a tree of class <code>phylo</code> , i.e. <code>tree\$edge[, 2]</code> .
dataset	Dataset of class <code>phyDat</code> . The dataset should have been prepared using <code>dataset &lt;- PrepareDataProfile(dataset)</code> ; if this step has not been completed, the dataset will be (time-consumingly) prepared within the function. In subsidiary functions, the dataset will have been initialized using <code>ProfileInitMorphy</code> , must be destroyed using <code>ProfileDestroyMorphy</code> .
concavity	A numeric value to use as the concavity constant ( $k$ ) in implied weighting. Defaults to 10; TNT sets a default of 3, but this is too low in some circumstances (Smith, 2019).
minLength	Integer vector specifying the minimum length possible for each character in dataset, perhaps calculated using <code>MinimumLength</code> .
...	unused; allows additional parameters specified within ... to be received by the function without throwing an error.
tree	A tree of class <code>phylo</code> .

**Value**

Zero minus the profile score (because the optimization algorithm treats smaller numbers as better)

**Functions**

- `IWScoreMorphy`: Scorer for Implied Weighting dataset.
- `ProfileScoreMorphy`: Scorer for initialized dataset.
- `ProfileInitMorphy`: Initialize dataset by adding `morphyObjs` attribute.
- `ProfileDestroyMorphy`: Free memory from `morphyObjs` initialized by `ProfileScoreMorphy()`.

**Author(s)**

Martin R. Smith

**References**

Faith DP, Trueman JWH (2001). "Towards an inclusive philosophy for phylogenetic inference." *Systematic Biology*, **50**(3), 331–350. doi: [10.1080/10635150118627](https://doi.org/10.1080/10635150118627).

**Examples**

```
data(referenceTree)
data(congreveLamsdellMatrices)
# In actual use, the dataset should be prepared with a much higher
# precision: try 1e+06?
# Of course, gaining higher precision takes substantially longer.
dataset <- suppressWarnings(
```

```
PrepareDataProfile(congreveLamsdellMatrices[[42]], precision = 1e+03))
ProfileScore(referenceTree, dataset)
```

---

 Jackknife

*Parsimony Ratchet*


---

### Description

Ratchet uses the parsimony ratchet (Nixon 1999) to search for a more parsimonious tree.

### Usage

```
Jackknife(
  tree,
  dataset,
  resampleFreq = 2/3,
  InitializeData = PhyDat2Morph,
  CleanUpData = UnloadMorph,
  TreeScorer = MorphyLength,
  EdgeSwapper = TBRSwap,
  jackIter = 5000L,
  searchIter = 4000L,
  searchHits = 42L,
  verbosity = 1L,
  ...
)

Ratchet(
  tree,
  dataset,
  InitializeData = PhyDat2Morph,
  CleanUpData = UnloadMorph,
  TreeScorer = MorphyLength,
  Bootstrapper = MorphyBootstrap,
  swappers = list(TBRSwap, SPRSwap, NNISwap),
  BootstrapSwapper = if (class(swappers) == "list") swappers[[length(swappers)]] else
    swappers,
  returnAll = FALSE,
  stopAtScore = NULL,
  stopAtPeak = FALSE,
  stopAtPlateau = 0L,
  ratchIter = 100,
  ratchHits = 10,
  searchIter = 4000,
  searchHits = 42,
  bootstrapIter = searchIter,
```

```
bootstrapHits = searchHits,
verbosity = 1L,
suboptimal = 1e-08,
...
)

ProfileRatchet(
  tree,
  dataset,
  swappers = list(TBRSwap, SPRSwap, NNISwap),
  BootstrapSwapper = if (class(swappers) == "list") swappers[[length(swappers)]] else
    swappers,
  returnAll = FALSE,
  stopAtScore = NULL,
  stopAtPeak = FALSE,
  stopAtPlateau = 0L,
  ratchIter = 100,
  ratchHits = 10,
  searchIter = 2000,
  searchHits = 40,
  bootstrapIter = searchIter,
  bootstrapHits = searchHits,
  verbosity = 1L,
  suboptimal = 1e-08,
  ...
)

IWRatchet(
  tree,
  dataset,
  concavity = 10,
  swappers = list(TBRSwap, SPRSwap, NNISwap),
  BootstrapSwapper = if (class(swappers) == "list") swappers[[length(swappers)]] else
    swappers,
  returnAll = FALSE,
  stopAtScore = NULL,
  stopAtPeak = FALSE,
  stopAtPlateau = 0L,
  ratchIter = 100,
  ratchHits = 10,
  searchIter = 2000,
  searchHits = 40,
  bootstrapIter = searchIter,
  bootstrapHits = searchHits,
  verbosity = 1L,
  suboptimal = 1e-08,
  ...
)
```

```
MultiRatchet(  
  tree,  
  dataset,  
  ratchHits = 10,  
  searchIter = 500,  
  searchHits = 20,  
  verbosity = 0L,  
  swappers = list(RootedNNISwap),  
  nSearch = 10,  
  stopAtScore = NULL,  
  ...  
)
```

```
IWMultiRatchet(  
  tree,  
  dataset,  
  ratchHits = 10,  
  concavity = 4,  
  searchIter = 500,  
  searchHits = 20,  
  verbosity = 0L,  
  swappers = list(RootedNNISwap),  
  nSearch = 10,  
  suboptimal = suboptimal,  
  stopAtScore = NULL,  
  ...  
)
```

```
RatchetConsensus(  
  tree,  
  dataset,  
  ratchHits = 10,  
  searchIter = 500,  
  searchHits = 20,  
  verbosity = 0L,  
  swappers = list(RootedNNISwap),  
  nSearch = 10,  
  stopAtScore = NULL,  
  ...  
)
```

```
IWRatchetConsensus(  
  tree,  
  dataset,  
  ratchHits = 10,  
  concavity = 4,  
  searchIter = 500,
```

```

    searchHits = 20,
    verbosity = 0L,
    swappers = list(RootedNNISwap),
    nSearch = 10,
    suboptimal = suboptimal,
    stopAtScore = NULL,
    ...
)

```

## Arguments

tree	A tree of class <a href="#">phylo</a> .
dataset	a dataset in the format required by <code>TreeScorer()</code> .
resampleFreq	Double between 0 and 1 stating proportion of characters to resample
InitializeData	Function that sets up data object to prepare for tree search. The function will be passed the dataset parameter. Its return value will be passed to <code>TreeScorer()</code> and <code>CleanUpData()</code> .
CleanUpData	Function to destroy data object on function exit. The function will be passed the value returned by <code>InitializeData()</code> .
TreeScorer	function to score a given tree. The function will be passed three parameters, corresponding to the parent and child entries of a tree's edge list, and a dataset.
EdgeSwapper	a function that rearranges a parent and child vector, and returns a list with modified vectors; for example <a href="#">SPRSwap()</a> .
jackIter	Integer specifying number of jackknife iterations to conduct
searchIter	Integer specifying maximum rearrangements to perform on each bootstrap or ratchet iteration. To override this value for a single swapper function, set e.g. <code>attr(SwapperFunction, 'searchIter') &lt;-99</code>
searchHits	Integer specifying maximum times to hit best score before terminating a tree search within a ratchet iteration. To override this value for a single swapper function, set e.g. <code>attr(SwapperFunction, 'searchHits') &lt;-99</code>
verbosity	Level of detail to display in console: larger numbers provide more verbose feedback to the user.
...	further arguments to pass to <code>TreeScorer()</code> (e.g. <a href="#">TipsAreColumns</a> , dataset).
Bootstrapper	Function to perform bootstrapped rearrangements of tree. First arguments will be an <code>edgeList</code> and a dataset, initialized using <code>InitializeData()</code> . Should return a rearranged <code>edgeList</code> .
swappers	A list of functions to use to conduct edge rearrangement during tree search. Provide functions like <a href="#">NNISwap</a> to shuffle root position, or <a href="#">RootedTBRSwap</a> if the position of the root should be retained. You may wish to use extreme swappers (such as TBR) early in the list, and a more subtle rearranger (such as NNI) later in the list to make incremental tinkering once an almost-optimal tree has been found.
BootstrapSwapper	Function such as <a href="#">RootedNNISwap</a> to use to rearrange trees within <code>Bootstrapper()</code> .



<code>returnAll</code>	Set to TRUE to report all MPTs encountered during the search, perhaps to analyse consensus.
<code>stopAtScore</code>	stop search as soon as this score is hit or beaten.
<code>stopAtPeak</code>	Logical specifying whether to terminate search once a subsequent iteration recovers a sub-optimal score. Useful with methods that return all trees one rearrangement from the current tree, such as <code>AllTBR()</code> . Will be overridden if a passed function has an attribute <code>stopAtPeak</code> set by <code>attr(FunctionName, 'stopAtPeak') &lt;-TRUE</code> .
<code>stopAtPlateau</code>	Integer. If > 0, tree search will terminate if the score has not improved after <code>stopAtPlateau</code> iterations. Useful with methods that return all trees one rearrangement from the current tree, such as <code>AllTBR()</code> . Will be overridden if a passed function has an attribute <code>stopAtPlateau</code> set by <code>attr(FunctionName, 'stopAtPlateau') &lt;-TRUE</code> .
<code>ratchIter</code>	Stop when this many ratchet iterations have been performed.
<code>ratchHits</code>	Stop when this many ratchet iterations have found the same best score.
<code>bootstrapIter</code>	Integer specifying maximum rearrangements to perform on each bootstrap iteration (default: <code>searchIter</code> ).
<code>bootstrapHits</code>	Integer specifying maximum times to hit best score on each bootstrap iteration (default: <code>searchHits</code> ).
<code>suboptimal</code>	retain trees that are suboptimal by this score. Defaults to 1e-08 to counter rounding errors.
<code>concavity</code>	A numeric value to use as the concavity constant (k) in implied weighting. Defaults to 10; TNT sets a default of 3, but this is too low in some circumstances (Smith, 2019).
<code>nSearch</code>	Number of Ratchet searches to conduct (for <code>RatchetConsensus()</code> )

### Value

a list of trees recovered after jackknife iterations

This function returns a tree modified by parsimony ratchet iterations.

`MultiRatchet()` returns a list of optimal trees produced by `nSearch` ratchet searches, from which a consensus tree can be generated using `ape::consensus()` or `TreeTools::ConsensusWithout()`.

`IWMultiRatchet` returns a list of optimal trees produced by `nSearch` Ratchet searches, using implied weighting.

### Functions

- `Jackknife`: Jackknife resampling. Note that at present this assumes that `InitializeData` will return a morphy object; if this doesn't hold for you, please let me know and I'll make the function more general.
- `ProfileRatchet`: Shortcut for Ratchet search under Profile Parsimony
- `IWRatchet`: Shortcut for Ratchet search using implied weights
- `RatchetConsensus`: deprecated alias for `MultiRatchet()`
- `IWRatchetConsensus`: deprecated alias for `MultiRatchet()`

**Author(s)**

Martin R. Smith

Martin R. Smith

**References**

- Nixon KC (1999). “The Parsimony Ratchet, a new method for rapid parsimony analysis.” *Cladistics*, **15**(4), 407–414. ISSN 0748-3007, doi: [10.1111/j.10960031.1999.tb00277.x](https://doi.org/10.1111/j.10960031.1999.tb00277.x).
- 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**

- [JackLabels\(\)](#): Label nodes of a tree with jackknife supports.
- [TreeSearch\(\)](#)
- Adapted from [pratchet](#) in the **phangorn** package.

**Examples**

```
data('Lobo', package='TreeTools')
njtree <- TreeTools::NJTree(Lobo.phy)
# Increase value of ratchetIter and searchHits to do a proper search
quickResult <- Ratchet(njtree, Lobo.phy, ratchetIter = 2, searchHits = 3)

# Plot result (legibly)
oldPar <- par(mar = rep(0, 4), cex = 0.75)
plot(quickResult)
par(oldPar)

# IW search is currently much slower:
quickIWRResult <- IWRatchet(quickResult, Lobo.phy, concavity = 2.5,
                           ratchetIter = 1, searchIter = 25, searchHits = 2,
                           swappers = RootedTBRSwap, verbosity = 5)
```

---

JackLabels

---

*Label nodes with jackknife support values*


---

**Description**

Label nodes with jackknife support values

**Usage**

```
JackLabels(  
  tree,  
  jackTrees,  
  add = FALSE,  
  adj = 0,  
  col = NULL,  
  frame = "none",  
  pos = 2L,  
  ...  
)
```

**Arguments**

tree	A tree of class <a href="#">phylo</a> .
jackTrees	A list or multiPhylo object containing trees generated by <a href="#">Jackknife()</a> .
add	Logical specifying whether to add the labels to an existing plot.
adj, col, frame, pos, ...	Parameters to pass to <a href="#">nodelabels</a> .

**Value**

A named vector specifying the proportion of jackknife trees consistent with each node in tree, as plotted.

**Author(s)**

[Martin R. Smith](#) ([martin.smith@durham.ac.uk](mailto:martin.smith@durham.ac.uk))

**See Also**

[Jackknife\(\)](#): Generate trees by jackknife resampling

**Examples**

```
library('TreeTools') # for as.phylo  
  
# jackTrees will usually be generated with Jackknife(), but for simplicity:  
jackTrees <- as.phylo(1:100, 8)  
  
JackLabels(as.phylo(0, 8), jackTrees)
```

---

LogisticPoints	<i>Logistic Points Extract points from a fitted model</i>
----------------	---

---

**Description**

Logistic Points Extract points from a fitted model

**Usage**

```
LogisticPoints(x, fittedModel)
```

**Arguments**

x	an integer vector giving x co-ordinates.
fittedModel	a fitted model, perhaps generated using <code>nls(cumP ~ SSlogis(nSteps, Asym, xmid, scal))</code> .

**Value**

values of y co-ordinates corresponding to the x co-ordinates provided

**Author(s)**

Martin R. Smith

---

MinimumLength	<i>Minimum length</i>
---------------	-----------------------

---

**Description**

The smallest length that a character can obtain on any tree.

**Usage**

```
MinimumLength(states)
```

```
MinimumSteps(states)
```

**Arguments**

states	Integer vector listing the tokens that may be present at each tip along a single character, with each token represented as a binary digit; e.g. a value of 11 (= $2^0 + 2^1 + 2^3$ ) means that the tip may have tokens 0, 1 or 3. Inapplicable tokens should be denoted with the integer 0 (not $2^0$ ). Tokens that are ambiguous for an inapplicable and an applicable state are not presently supported; for an approximate value, denote such ambiguity with the integer 0.
--------	--

**Value**

An integer specifying the minimum number of steps that the character must contain.

**Author(s)**

Martin R. Smith

**Examples**

```
{
  data('inapplicable.datasets')
  myPhyDat <- inapplicable.phyData[[4]]
  class(myPhyDat) # phyDat object
  # load your own data with
  # my.PhyDat <- as.phyDat(read.nexus.data('filepath'))
  # or Windows users can select a file interactively using:
  # my.PhyDat <- as.phyDat(read.nexus.data(choose.files()))

  # Convert list of character codings to an array
  myData <- vapply(myPhyDat, I, myPhyDat[[1]])

  # Convert phyDat's representation of states to binary
  myContrast <- attr(myPhyDat, 'contrast')
  tokens <- colnames(myContrast)
  binaryContrast <- integer(length(tokens))
  tokenApplicable <- tokens != '-'
  binaryContrast[tokenApplicable] <- 2 ^ (seq_len(sum(tokenApplicable)) - 1)
  binaryValues <- apply(myContrast, 1,
    function (row) sum(binaryContrast[as.logical(row)]))
  myStates <- matrix(binaryValues[myData], nrow=nrow(myData),
    ncol=ncol(myData), dimnames=dimnames(myData))

  # Finally, work out minimum steps
  apply(myStates, 1, MinimumLength)
}
```

**Description**

Ratchet bootstrapper

**Usage**

```
MorphyBootstrap(
  edgeList,
  morphyObj,
  EdgeSwapper = NNISwap,
  maxIter,
  maxHits,
  verbosity = 1L,
  stopAtPeak = FALSE,
  stopAtPlateau = 0L,
  ...
)
```

```
ProfileBootstrap(
  edgeList,
  dataset,
  EdgeSwapper = NNISwap,
  maxIter,
  maxHits,
  verbosity = 1L,
  ...
)
```

```
IWBootstrap(
  edgeList,
  dataset,
  concavity = 10L,
  EdgeSwapper = NNISwap,
  maxIter,
  maxHits,
  verbosity = 1L,
  ...
)
```

**Arguments**

edgeList	a list containing the following: - vector of integers corresponding to the parent of each edge in turn - vector of integers corresponding to the child of each edge in turn - (optionally) score of the tree - (optionally, if score provided) number of times this score has been hit
morphyObj	Object of class morphy, perhaps created with <a href="#">PhyDat2Morphy()</a> .
EdgeSwapper	a function that rearranges a parent and child vector, and returns a list with modified vectors; for example <a href="#">SPRSwap()</a> .
maxIter	maximum number of iterations to perform in tree search
maxHits	maximum number of hits to accomplish in tree search
verbosity	Level of detail to display in console: larger numbers provide more verbose feedback to the user.

stopAtPeak	Logical specifying whether to terminate search once a subsequent iteration recovers a sub-optimal score. Useful with methods that return all trees one rearrangement from the current tree, such as <a href="#">AllTBR()</a> . Will be overridden if a passed function has an attribute stopAtPeak set by <code>attr(FunctionName, 'stopAtPeak') &lt;-TRUE</code> .
stopAtPlateau	Integer. If > 0, tree search will terminate if the score has not improved after stopAtPlateau iterations. Useful with methods that return all trees one rearrangement from the current tree, such as <a href="#">AllTBR()</a> . Will be overridden if a passed function has an attribute stopAtPlateau set by <code>attr(FunctionName, 'stopAtPlateau') &lt;-TRUE</code> .
...	further parameters to send to <code>TreeScorer()</code>
dataset	A phylogenetic data matrix of class <a href="#">phyDat</a> , whose names correspond to the labels of any accompanying tree.
concavity	A numeric value to use as the concavity constant (k) in implied weighting. Defaults to 10; TNT sets a default of 3, but this is too low in some circumstances (Smith, 2019).

**Value**

A tree that is optimal under a random sampling of the original characters

**Functions**

- `ProfileBootstrap`: Bootstrapper for Profile Parsimony
- `IWBootstrap`: Bootstrapper for Implied weighting

**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).

---

MorphyWeights

*Report the character weightings associated with a Morphy object*


---

**Description**

Report the character weightings associated with a Morphy object

**Usage**

```
MorphyWeights(morphyObj)
```

**Arguments**

morphyObj      Object of class morphy, perhaps created with [PhyDat2Morphy\(\)](#).

**Value**

a matrix of dimensions (2, number of characters); row 1 lists the exact rates specified by the user; row 2 the approximate (integral) weights used by MorphyLib

**Author(s)**

Martin R. Smith

**See Also**

Other Morphy API functions: [MorphyErrorCheck\(\)](#), [PhyDat2Morphy\(\)](#), [SetMorphyWeights\(\)](#), [SingleCharMorphy\(\)](#), [UnloadMorphy\(\)](#), [mpl\\_apply\\_tipdata\(\)](#), [mpl\\_attach\\_rawdata\(\)](#), [mpl\\_attach\\_symbols\(\)](#), [mpl\\_delete\\_Morphy\(\)](#), [mpl\\_delete\\_rawdata\(\)](#), [mpl\\_first\\_down\\_recon\(\)](#), [mpl\\_first\\_up\\_recon\(\)](#), [mpl\\_get\\_charac\\_weight\(\)](#), [mpl\\_get\\_num\\_charac\(\)](#), [mpl\\_get\\_num\\_internal\\_nodes\(\)](#), [mpl\\_get\\_numtaxa\(\)](#), [mpl\\_get\\_symbols\(\)](#), [mpl\\_init\\_Morphy\(\)](#), [mpl\\_new\\_Morphy\(\)](#), [mpl\\_second\\_down\\_recon\(\)](#), [mpl\\_second\\_up\\_recon\(\)](#), [mpl\\_set\\_charac\\_weight\(\)](#), [mpl\\_set\\_num\\_internal\\_nodes\(\)](#), [mpl\\_set\\_parsim\\_t\(\)](#), [mpl\\_translate\\_error\(\)](#), [mpl\\_update\\_lower\\_root\(\)](#), [mpl\\_update\\_tip\(\)](#), [summary.morphyPtr\(\)](#)

---

NNI

*Nearest Neighbour Interchange (NNI)*

---

**Description**

Performs a single iteration of the nearest-neighbour interchange algorithm. Based on the corresponding phangorn function, but re-coded to improve speed.

**Usage**

```
NNI(tree, edgeToBreak = NULL)
```

```
NNISwap(parent, child, nTips = (length(parent)/2L) + 1L, edgeToBreak = NULL)
```

```
RootedNNI(tree, edgeToBreak = NULL)
```

```
RootedNNISwap(
  parent,
  child,
  nTips = (length(parent)/2L) + 1L,
  edgeToBreak = NULL
)
```

**Arguments**

`tree` A tree of class [phylo](#).

`edgeToBreak` (optional) integer specifying the index of an edge to bisect/prune, generated randomly if not specified. Alternatively, set to -1 to return a complete list of all trees one step from the input tree.



parent	Integer vector corresponding to the first column of the edge matrix of a tree of class <code>phylo</code> , i.e. <code>tree\$edge[,1]</code> .
child	Integer vector corresponding to the second column of the edge matrix of a tree of class <code>phylo</code> , i.e. <code>tree\$edge[,2]</code> .
nTips	(optional) Number of tips.

### Details

Branch lengths are not supported.

All nodes in a tree must be bifurcating; `ape::collapse.singles` and `ape::multi2di` may help.

### Value

Returns a tree with class `phylo` (if `returnAll = FALSE`) or a set of trees, with class `multiPhylo` (if `returnAll = TRUE`).

a list containing two elements, corresponding in turn to the rearranged parent and child parameters

a list containing two elements, corresponding in turn to the rearranged parent and child parameters

### Functions

- `NNISwap`: faster version that takes and returns parent and child parameters
- `RootedNNI`: Perform NNI rearrangement, retaining position of root
- `RootedNNISwap`: faster version that takes and returns parent and child parameters

### Author(s)

Martin R. Smith

### References

The algorithm is summarized in Felsenstein J (2004). *Inferring phylogenies*. Sinauer Associates, Sunderland, Massachusetts.

### See Also

Other tree rearrangement functions: `SPR()`, `TBR()`

### Examples

```
tree <- ape::rtree(20, br = NULL)
NNI(tree)
NNI(tree, edgeToBreak = -1)
```

---

PhyDat2Morphy	<i>Initialize a Morphy Object from a phyDat object</i>
---------------	--

---

**Description**

Creates a new Morphy object with the same size and characters as the phyDat object

**Usage**

```
PhyDat2Morphy(phy)
```

**Arguments**

phy                    An object of class [phyDat](#).

**Value**

A pointer to an initialized Morphy object.

**Author(s)**

Martin R. Smith

**See Also**

Other Morphy API functions: [MorphyErrorCheck\(\)](#), [MorphyWeights\(\)](#), [SetMorphyWeights\(\)](#), [SingleCharMorphy\(\)](#), [UnloadMorphy\(\)](#), [mpl\\_apply\\_tipdata\(\)](#), [mpl\\_attach\\_rawdata\(\)](#), [mpl\\_attach\\_symbols\(\)](#), [mpl\\_delete\\_Morphy\(\)](#), [mpl\\_delete\\_rawdata\(\)](#), [mpl\\_first\\_down\\_recon\(\)](#), [mpl\\_first\\_up\\_recon\(\)](#), [mpl\\_get\\_charac\\_weight\(\)](#), [mpl\\_get\\_num\\_charac\(\)](#), [mpl\\_get\\_num\\_internal\\_nodes\(\)](#), [mpl\\_get\\_numtaxa\(\)](#), [mpl\\_get\\_symbols\(\)](#), [mpl\\_init\\_Morphy\(\)](#), [mpl\\_new\\_Morphy\(\)](#), [mpl\\_second\\_down\\_recon\(\)](#), [mpl\\_second\\_up\\_recon\(\)](#), [mpl\\_set\\_charac\\_weight\(\)](#), [mpl\\_set\\_num\\_internal\\_nodes\(\)](#), [mpl\\_set\\_parsim\\_t\(\)](#), [mpl\\_translate\\_error\(\)](#), [mpl\\_update\\_lower\\_root\(\)](#), [mpl\\_update\\_tip\(\)](#), [summary.morphyPtr\(\)](#)

---

PrepareDataProfile	<i>Prepare data for Profile Parsimony</i>
--------------------	---

---

**Description**

Prepare data for Profile Parsimony

**Usage**

```
PrepareDataProfile(dataset, precision = 1e+06, warn = TRUE)
```

```
PrepareDataIW(dataset)
```

**Arguments**

dataset	dataset of class phyDat
precision	number of random trees to generate when calculating Profile curves. With 22 tokens (taxa): - Increasing precision from 4e+05 to 4e+06 reduces error by a mean of 0.005 bits for each step after the first (max = 0.11 bits, sd=0.017 bits) - Increasing precision from 1e+06 to 4e+06 reduces error by a mean of 0.0003 bits for each step after the first (max = 0.046 bits, sd=0.01 bits)
warn	Boolean (TRUE/FALSE): display warnings when concavity functions are generated by approximation.

**Value**

An object of class phyDat, with additional attributes. PrepareDataProfile adds the attributes:

- `info.amounts`: details the information represented by each character when subject to N additional steps.
- `split.sizes`: The size of the splits implied by each character
- `bootstrap`: The character vector `c('info.amounts', 'split.sizes')`, indicating attributes to sample when bootstrapping the dataset (e.g. in Ratchet searches).

PrepareDataIW adds the attribute:

- `min.length`: The minimum number of steps that must be present in each transformation series.

**Functions**

- `PrepareDataIW`: Prepare data for implied weighting

**Author(s)**

Martin R. Smith; written with reference to `phangorn::prepareDataFitch()`

---

RandomMorphyTree      *Random postorder tree*

---

**Description**

Random postorder tree

**Usage**

RandomMorphyTree(nTip)

**Arguments**

nTip                    Integer specifying the number of tips to include in the tree (minimum 2).

**Value**

A list with three elements, each a vector of integers, respectively containing:

- The parent of each tip and node, in order
- The left child of each node
- The right child of each node.

---

RandomTreeScore	<i>Parsimony score of random postorder tree</i>
-----------------	---

---

**Description**

Parsimony score of random postorder tree

**Usage**

```
RandomTreeScore(nTip, morphyObj)
```

**Arguments**

nTip	number of tips (minimum 3)
morphyObj	Object of class morphy, perhaps created with <a href="#">PhyDat2Morphy()</a> .

**Value**

the parsimony score of a random tree, for the given Morphy object.

---

RearrangeEdges	<i>Rearrange edges of a phylogenetic tree</i>
----------------	---

---

**Description**

RearrangeEdges() performs the specified edge rearrangement on a matrix that corresponds to the edges of a phylogenetic tree, returning the score of the new tree. Will generally be called from within a tree search function.

**Usage**

```
RearrangeEdges(
  parent,
  child,
  dataset,
  TreeScorer = MorphyLength,
  EdgeSwapper,
  scoreToBeat = TreeScorer(parent, child, dataset, ...),
  iter = "?",
  hits = 0L,
  verbosity = 0L,
  ...
)
```

**Arguments**

parent	Integer vector corresponding to the first column of the edge matrix of a tree of class <code>phylo</code> , i.e. <code>tree\$edge[, 1]</code> .
child	Integer vector corresponding to the second column of the edge matrix of a tree of class <code>phylo</code> , i.e. <code>tree\$edge[, 2]</code> .
dataset	Third argument to pass to <code>TreeScorer</code> .
TreeScorer	function to score a given tree. The function will be passed three parameters, corresponding to the <code>parent</code> and <code>child</code> entries of a tree's edge list, and a dataset.
EdgeSwapper	a function that rearranges a parent and child vector, and returns a list with modified vectors; for example <code>SPRSwap()</code> .
scoreToBeat	Double giving score of input tree.
iter	iteration number of calling function, for reporting to user only.
hits	Integer giving number of times the input tree has already been hit.
verbosity	Level of detail to display in console: larger numbers provide more verbose feedback to the user.
...	further arguments to pass to <code>TreeScorer()</code> (e.g. <code>TipsAreColumns</code> , <code>dataset</code> ).

**Details**

`RearrangeTree()` performs one tree rearrangement of a specified type, and returns the score of the tree (with the given dataset). It also reports the number of times that this score was hit in the current function call.

**Value**

This function returns a list with two to four elements, corresponding to a binary tree: - 1. Integer vector listing the parent node of each edge; - 2. Integer vector listing the child node of each edge; - 3. Score of the tree; - 4. Number of times that score has been hit.

**Author(s)**

Martin R. Smith

### Examples

```
data('Lobo', package='TreeTools')
tree <- TreeTools::NJTree(Lobo.phy)
edge <- tree$edge
parent <- edge[, 1]
child <- edge[, 2]
dataset <- PhyDat2Morph(Lobo.phy)
RearrangeEdges(parent, child, dataset, EdgeSwapper = RootedNNISwap)
```

---

referenceTree

*Tree topology for matrix simulation*

---

### Description

The tree topology used to generate the matrices in [congreveLamsdellMatrices](#) Congreve & Lamsdell (2016)

### Usage

```
referenceTree
```

### Format

A single phylogenetic tree saved as an object of class `phylo`

### Source

<https://dx.doi.org/10.1111/pala.12236>

### References

Congreve CR, Lamsdell JC (2016). “Implied weighting and its utility in palaeontological datasets: a study using modelled phylogenetic matrices.” *Palaeontology*, **59**, 447–465. doi: [10.1111/pala.12236](https://doi.org/10.1111/pala.12236).  
Congreve CR, Lamsdell JC (2016). “Data from: Implied weighting and its utility in palaeontological datasets: a study using modelled phylogenetic matrices.” *Dryad Digital Repository*, doi:10.5061/dryad.7dq0j. doi: [10.5061/dryad.7dq0j](https://doi.org/10.5061/dryad.7dq0j).

### Examples

```
data(referenceTree)
plot(referenceTree)
```

---

SetMorphyWeights	<i>Set the character weightings associated with a Morphy object</i>
------------------	---

---

**Description**

Set the character weightings associated with a Morphy object

**Usage**

```
SetMorphyWeights(weight, morphyObj, checkInput = TRUE)
```

**Arguments**

weight	A vector listing the new weights to be applied to each character
morphyObj	Object of class morphy, perhaps created with <a href="#">PhyDat2Morphy()</a> .
checkInput	Whether to sanity-check input data before applying. Defaults to TRUE to protect the user from crashes.

**Value**

SetMorphyWeights() returns the Morphy error code generated when applying tipData.

**Author(s)**

Martin R. Smith

**See Also**

Other Morphy API functions: [MorphyErrorCheck\(\)](#), [MorphyWeights\(\)](#), [PhyDat2Morphy\(\)](#), [SingleCharMorphy\(\)](#), [UnloadMorphy\(\)](#), [mpl\\_apply\\_tipdata\(\)](#), [mpl\\_attach\\_rawdata\(\)](#), [mpl\\_attach\\_symbols\(\)](#), [mpl\\_delete\\_Morphy\(\)](#), [mpl\\_delete\\_rawdata\(\)](#), [mpl\\_first\\_down\\_recon\(\)](#), [mpl\\_first\\_up\\_recon\(\)](#), [mpl\\_get\\_charac\\_weight\(\)](#), [mpl\\_get\\_num\\_charac\(\)](#), [mpl\\_get\\_num\\_internal\\_nodes\(\)](#), [mpl\\_get\\_numtaxa\(\)](#), [mpl\\_get\\_symbols\(\)](#), [mpl\\_init\\_Morphy\(\)](#), [mpl\\_new\\_Morphy\(\)](#), [mpl\\_second\\_down\\_recon\(\)](#), [mpl\\_second\\_up\\_recon\(\)](#), [mpl\\_set\\_charac\\_weight\(\)](#), [mpl\\_set\\_num\\_internal\\_nodes\(\)](#), [mpl\\_set\\_parsim\\_t\(\)](#), [mpl\\_translate\\_error\(\)](#), [mpl\\_update\\_lower\\_root\(\)](#), [mpl\\_update\\_tip\(\)](#), [summary.morphyPtr\(\)](#)

---

SingleCharMorphy	<i>Morphy object from single character</i>
------------------	--

---

**Description**

Morphy object from single character

**Usage**

```
SingleCharMorphy(char)
```

**Arguments**

char                    State of each character at each tip in turn, in a format that will be converted to a character string by `paste0(char, ';', collapse='')`.

**Value**

A pointer to an object of class `morphObj`. Don't forget to unload it when you've finished with it: `morphObj <-UnloadMorphy(morphObj)`.

**Author(s)**

Martin R. Smith

**See Also**

Other Morphy API functions: `MorphyErrorCheck()`, `MorphyWeights()`, `PhyDat2Morphy()`, `SetMorphyWeights()`, `UnloadMorphy()`, `mpl_apply_tipdata()`, `mpl_attach_rawdata()`, `mpl_attach_symbols()`, `mpl_delete_Morphy()`, `mpl_delete_rawdata()`, `mpl_first_down_recon()`, `mpl_first_up_recon()`, `mpl_get_charac_weight()`, `mpl_get_num_charac()`, `mpl_get_num_internal_nodes()`, `mpl_get_numtaxa()`, `mpl_get_symbols()`, `mpl_init_Morphy()`, `mpl_new_Morphy()`, `mpl_second_down_recon()`, `mpl_second_up_recon()`, `mpl_set_charac_weight()`, `mpl_set_num_internal_nodes()`, `mpl_set_parsim_t()`, `mpl_translate_error()`, `mpl_update_lower_root()`, `mpl_update_tip()`, `summary.morphyPtr()`

---

 SPR

*Subtree Pruning and Rearrangement (SPR)*


---

**Description**

Perform one SPR rearrangement on a tree

**Usage**

```
SPR(tree, edgeToBreak = NULL, mergeEdge = NULL)
```

```
SPRSwap(
  parent,
  child,
  nEdge = length(parent),
  nNode = nEdge/2L,
  edgeToBreak = NULL,
  mergeEdge = NULL
)
```

```
RootedSPR(tree, edgeToBreak = NULL, mergeEdge = NULL)
```

```
RootedSPRSwap(
  parent,
```



```

    child,
    nEdge = length(parent),
    nNode = nEdge/2L,
    edgeToBreak = NULL,
    mergeEdge = NULL
  )

```

### Arguments

tree	A tree of class <a href="#">phylo</a> .
edgeToBreak	the index of an edge to bisect, generated randomly if not specified.
mergeEdge	the index of an edge on which to merge the broken edge.
parent	Integer vector corresponding to the first column of the edge matrix of a tree of class <a href="#">phylo</a> , i.e. <code>tree\$edge[, 1]</code> .
child	Integer vector corresponding to the second column of the edge matrix of a tree of class <a href="#">phylo</a> , i.e. <code>tree\$edge[, 2]</code> .
nEdge	(optional) integer specifying the number of edges of a tree of class <a href="#">phylo</a> , i.e. <code>dim(tree\$edge)[1]</code>
nNode	(optional) Number of nodes.

### Details

Equivalent to `kSPR` in the `phangorn` package, but faster. Note that rearrangements that only change the position of the root WILL be returned by `SPR`. If the position of the root is irrelevant (as in Fitch parsimony, for example) then this function will occasionally return a functionally equivalent topology. `RootIrrelevantSPR` will search tree space more efficiently in these cases. Branch lengths are not (yet) supported.

All nodes in a tree must be bifurcating; [ape::collapse.singles](#) and [ape::multi2di](#) may help.

### Value

This function returns a tree in `phyDat` format that has undergone one `SPR` iteration.

a list containing two elements, corresponding in turn to the rearranged parent and child parameters

a list containing two elements, corresponding in turn to the rearranged parent and child parameters

### Functions

- `SPRswap`: faster version that takes and returns parent and child parameters
- `RootedSPR`: Perform `SPR` rearrangement, retaining position of root
- `RootedSPRswap`: faster version that takes and returns parent and child parameters

### Author(s)

Martin R. Smith

**References**

The SPR algorithm is summarized in Felsenstein J (2004). *Inferring phylogenies*. Sinauer Associates, Sunderland, Massachusetts.

**See Also**

- [RootedSPR\(\)](#): useful when the position of the root node should be retained.

Other tree rearrangement functions: [NNI\(\)](#), [TBR\(\)](#)

**Examples**

```
{
tree <- ape::rtree(20, br=FALSE)
SPR(tree)
}
```

---

SuccessiveApproximations

*Tree Search using Successive Approximations*

---

**Description**

Searches for a tree that is optimal under the Successive Approximations criterion.

**Usage**

```
SuccessiveApproximations(
  tree,
  dataset,
  outgroup = NULL,
  k = 3,
  maxSuccIter = 20,
  ratchetHits = 100,
  searchHits = 50,
  searchIter = 500,
  ratchetIter = 5000,
  verbosity = 0,
  suboptimal = 0.1
)
```

**Arguments**

tree	A tree of class <a href="#">phylo</a> .
dataset	A phylogenetic data matrix of class <a href="#">phyDat</a> , whose names correspond to the labels of any accompanying tree.
outgroup	if not NULL, taxa on which the tree should be rooted

k	Constant for successive approximations, see Farris 1969 p. 379
maxSuccIter	maximum iterations of successive approximation
ratchetHits	maximum hits for parsimony ratchet
searchHits	maximum hits in tree search
searchIter	maximum iterations in tree search
ratchetIter	maximum iterations of parsimony ratchet
verbosity	integer (default 0) specifying how much detail to print to stdout
suboptimal	retain trees that are this proportion less optimal than the optimal tree

**Value**

SuccessiveApproximations() returns a list of class multiPhylo containing optimal (and slightly suboptimal, if suboptimal > 0) trees.

---

summary.morphyPtr      *Details the attributes of a morphy object*

---

**Description**

Details the attributes of a morphy object

**Usage**

```
## S3 method for class 'morphyPtr'
summary(object, ...)
```

**Arguments**

object	A Morphy object
...	any other parameters...

**Value**

A list detailing the number of taxa, internal nodes, and characters and their weights.

**Author(s)**

Martin R. Smith

**See Also**

Other Morphy API functions: [MorphyErrorCheck\(\)](#), [MorphyWeights\(\)](#), [PhyDat2Morphy\(\)](#), [SetMorphyWeights\(\)](#), [SingleCharMorphy\(\)](#), [UnloadMorphy\(\)](#), [mpl\\_apply\\_tipdata\(\)](#), [mpl\\_attach\\_rawdata\(\)](#), [mpl\\_attach\\_symbols\(\)](#), [mpl\\_delete\\_Morphy\(\)](#), [mpl\\_delete\\_rawdata\(\)](#), [mpl\\_first\\_down\\_recon\(\)](#), [mpl\\_first\\_up\\_recon\(\)](#), [mpl\\_get\\_charac\\_weight\(\)](#), [mpl\\_get\\_num\\_charac\(\)](#), [mpl\\_get\\_num\\_internal\\_nodes\(\)](#), [mpl\\_get\\_numtaxa\(\)](#), [mpl\\_get\\_symbols\(\)](#), [mpl\\_init\\_Morphy\(\)](#), [mpl\\_new\\_Morphy\(\)](#), [mpl\\_second\\_down\\_recon\(\)](#), [mpl\\_second\\_up\\_recon\(\)](#), [mpl\\_set\\_charac\\_weight\(\)](#), [mpl\\_set\\_num\\_internal\\_nodes\(\)](#), [mpl\\_set\\_parsim\\_t\(\)](#), [mpl\\_translate\\_error\(\)](#), [mpl\\_update\\_lower\\_root\(\)](#), [mpl\\_update\\_tip\(\)](#)

---

TBR

*Tree bisection and reconnection (TBR)*


---

### Description

TBR performs a single random TBR iteration.

### Usage

```
TBR(tree, edgeToBreak = NULL, mergeEdges = NULL)
```

```
TBRSwap(
  parent,
  child,
  nEdge = length(parent),
  edgeToBreak = NULL,
  mergeEdges = NULL
)
```

```
TBRMoves(
  parent,
  child,
  nEdge = length(parent),
  avoid = NULL,
  retainRoot = FALSE
)
```

```
AllTBR(parent, child, nEdge = length(parent), avoid = NULL, retainRoot = FALSE)
```

```
RootedTBR(tree, edgeToBreak = NULL, mergeEdges = NULL)
```

```
RootedTBRSwap(
  parent,
  child,
  nEdge = length(parent),
  edgeToBreak = NULL,
  mergeEdges = NULL
)
```

### Arguments

tree	A bifurcating tree of class <a href="#">phylo</a> , with all nodes resolved;
edgeToBreak	(optional) integer specifying the index of an edge to bisect/prune, generated randomly if not specified. Alternatively, set to -1 to return a complete list of all trees one step from the input tree.

mergeEdges	(optional) vector of length 1 or 2, listing edge(s) to be joined: In SPR, this is where the pruned subtree will be reconnected. In TBR, these edges will be reconnected (so must be on opposite sides of edgeToBreak); if only a single edge is specified, the second will be chosen at random
parent	Integer vector corresponding to the first column of the edge matrix of a tree of class <code>phylo</code> , i.e. <code>tree\$edge[,1]</code> .
child	Integer vector corresponding to the second column of the edge matrix of a tree of class <code>phylo</code> , i.e. <code>tree\$edge[,2]</code> .
nEdge	(optional) Number of edges.
avoid	Integer vector specifying which edges should not be broken
retainRoot	logical specifying whether taxa may be swapped across the root

### Details

Branch lengths are not (yet) supported.

All nodes in a tree must be bifurcating; `ape::collapse.singles` and `ape::multi2di` may help.

### Value

This function returns a tree in `phyDat` format that has undergone one TBR iteration.

a list containing two elements, corresponding in turn to the rearranged parent and child parameters

a matrix with two columns, each row listing an edge that can be broken and an edge into which it can be merged

a list of trees, in parent-child format

### Functions

- `TBRSwap`: faster version that takes and returns parent and child parameters
- `TBRMoves`: Possible TBR moves
- `AllTBR`: All unique trees one TBR move away
- `RootedTBR`: Perform TBR rearrangement, retaining position of root
- `RootedTBRSwap`: faster version that takes and returns parent and child parameters

### Author(s)

Martin R. Smith

### References

The TBR algorithm is summarized in Felsenstein J (2004). *Inferring phylogenies*. Sinauer Associates, Sunderland, Massachusetts.

### See Also

`RootedTBR()`: useful when the position of the root node should be retained.

Other tree rearrangement functions: `NNI()`, `SPR()`

**Examples**

```
{
  library('ape')
  tree <- rtree(20, br=NULL)
  TBR(tree)
}
```

---

 UnloadMorphy

*Destroy a Morphy Object*


---

**Description**

Destroys a previously-created Morphy object.

**Usage**

```
UnloadMorphy(morphObj)
```

**Arguments**

`morphObj`      Object of class `morphy`, perhaps created with `PhyDat2Morphy()`.

**Details**

Best practice is to call `morphObj <- UnloadMorphy(morphObj)` Failure to do so will cause a crash if `UnloadMorphy()` is called on an object that has already been destroyed

**Value**

Morphy error code, decipherable using `mpl_translate_error`

**Author(s)**

Martin R. Smith

**See Also**

Other Morphy API functions: `MorphyErrorCheck()`, `MorphyWeights()`, `PhyDat2Morphy()`, `SetMorphyWeights()`, `SingleCharMorphy()`, `mpl_apply_tipdata()`, `mpl_attach_rawdata()`, `mpl_attach_symbols()`, `mpl_delete_Morphy()`, `mpl_delete_rawdata()`, `mpl_first_down_recon()`, `mpl_first_up_recon()`, `mpl_get_charac_weight()`, `mpl_get_num_charac()`, `mpl_get_num_internal_nodes()`, `mpl_get_numtaxa()`, `mpl_get_symbols()`, `mpl_init_Morphy()`, `mpl_new_Morphy()`, `mpl_second_down_recon()`, `mpl_second_up_recon()`, `mpl_set_charac_weight()`, `mpl_set_num_internal_nodes()`, `mpl_set_parsim_t()`, `mpl_translate_error()`, `mpl_update_lower_root()`, `mpl_update_tip()`, `summary.morphyPtr()`

---

WithOneExtraStep	<i>Number of trees with one extra step</i>
------------------	--

---

**Description**

Number of trees with one extra step

**Usage**

WithOneExtraStep(splits)

**Arguments**

splits      A vector of integers listing the number of tips in each of a number of tree splits (e.g. bipartitions). For example,  $c(3, 5)$  states that a character divides a set of eight tips into a group of three and a group of five.

# Index

## \* Morphy API functions

MorphyWeights, 31  
PhyDat2Morphy, 34  
SetMorphyWeights, 39  
SingleCharMorphy, 39  
summary.morphyPtr, 43  
UnloadMorphy, 46

## \* datasets

congreveLamsdellMatrices, 5  
inapplicable.citations, 8  
inapplicable.datasets, 11  
inapplicable.phyData, 14  
referenceTree, 38

## \* tree generation functions

RandomMorphyTree, 35

## \* tree rearrangement functions

NNI, 32  
SPR, 40  
TBR, 44

## \* tree scoring

CharacterLength, 4  
Fitch, 6  
IWScore, 18

## \* tree

ICSteps, 7  
IWScore, 18  
IWScoreMorphy, 19  
Jackknife, 21

AllSPR, 3

AllTBR (TBR), 44

AllTBR(), 25, 31

ape::collapse.singles, 33, 41, 45

ape::consensus(), 25

ape::multi2di, 33, 41, 45

AsBinary, 3

CharacterLength, 4, 7, 19

congreveLamsdellMatrices, 5, 38

Evaluate, 6

FastCharacterLength (CharacterLength), 4

Fitch, 5, 6, 19

FitchSteps (CharacterLength), 4

ICSteps, 7

inapplicable.citations, 8

inapplicable.datasets, 11

inapplicable.phyData, 14

InfoAmounts, 17

IWBootstrap (MorphyBootstrap), 29

IWDestroyMorphy (IWScore), 18

IWInitMorphy (IWScore), 18

IWMultiRatchet (Jackknife), 21

IWRatchet (Jackknife), 21

IWRatchetConsensus (Jackknife), 21

IWScore, 5, 7, 18

IWScoreMorphy, 19

Jackknife, 21

Jackknife(), 27

JackLabels, 26

JackLabels(), 26

LogisticPoints, 28

MinimumLength, 20, 28

MinimumSteps (MinimumLength), 28

MorphyBootstrap, 29

MorphyErrorCheck, 32, 34, 39, 40, 43, 46

MorphyTreeLength, 5, 7, 19

MorphyWeights, 31, 34, 39, 40, 43, 46

mpl\_apply\_tipdata, 32, 34, 39, 40, 43, 46

mpl\_attach\_rawdata, 32, 34, 39, 40, 43, 46

mpl\_attach\_symbols, 32, 34, 39, 40, 43, 46

mpl\_delete\_Morphy, 32, 34, 39, 40, 43, 46

mpl\_delete\_rawdata, 32, 34, 39, 40, 43, 46

mpl\_first\_down\_recon, 32, 34, 39, 40, 43, 46

mpl\_first\_up\_recon, 32, 34, 39, 40, 43, 46



- mpl\_get\_charac\_weight, [32, 34, 39, 40, 43, 46](#)
- mpl\_get\_num\_charac, [32, 34, 39, 40, 43, 46](#)
- mpl\_get\_num\_internal\_nodes, [32, 34, 39, 40, 43, 46](#)
- mpl\_get\_numtaxa, [32, 34, 39, 40, 43, 46](#)
- mpl\_get\_symbols, [32, 34, 39, 40, 43, 46](#)
- mpl\_init\_Morphy, [32, 34, 39, 40, 43, 46](#)
- mpl\_new\_Morphy, [32, 34, 39, 40, 43, 46](#)
- mpl\_second\_down\_recon, [32, 34, 39, 40, 43, 46](#)
- mpl\_second\_up\_recon, [32, 34, 39, 40, 43, 46](#)
- mpl\_set\_charac\_weight, [32, 34, 39, 40, 43, 46](#)
- mpl\_set\_num\_internal\_nodes, [32, 34, 39, 40, 43, 46](#)
- mpl\_set\_parsim\_t, [32, 34, 39, 40, 43, 46](#)
- mpl\_translate\_error, [32, 34, 39, 40, 43, 46](#)
- mpl\_update\_lower\_root, [32, 34, 39, 40, 43, 46](#)
- mpl\_update\_tip, [32, 34, 39, 40, 43, 46](#)
- MultiRatchet (Jackknife), [21](#)
- NNI, [32, 42, 45](#)
- NNISwap, [24](#)
- NNISwap (NNI), [32](#)
- paste0, [40](#)
- phyDat, [4, 6, 31, 34, 42](#)
- PhyDat2Morphy, [32, 34, 39, 40, 43, 46](#)
- PhyDat2Morphy(), [30, 31, 36, 39, 46](#)
- phylo, [3, 4, 6, 18, 20, 24, 27, 32, 33, 37, 41, 42, 44, 45](#)
- pratchet, [26](#)
- PrepareDataIW, [18](#)
- PrepareDataIW (PrepareDataProfile), [34](#)
- PrepareDataProfile, [20, 34](#)
- ProfileBootstrap (MorphyBootstrap), [29](#)
- ProfileDestroyMorphy (IWScoreMorphy), [19](#)
- ProfileInitMorphy (IWScoreMorphy), [19](#)
- ProfileRatchet (Jackknife), [21](#)
- ProfileScore (IWScoreMorphy), [19](#)
- ProfileScoreMorphy (IWScoreMorphy), [19](#)
- RandomMorphyTree, [35](#)
- RandomTreeScore, [36](#)
- Ratchet (Jackknife), [21](#)
- RatchetConsensus (Jackknife), [21](#)
- read.nexus.data, [11](#)
- RearrangeEdges, [36](#)
- referenceTree, [5, 38](#)
- RootedNNI (NNI), [32](#)
- RootedNNISwap, [24](#)
- RootedNNISwap (NNI), [32](#)
- RootedSPR (SPR), [40](#)
- RootedSPR(), [42](#)
- RootedSPRSwap (SPR), [40](#)
- RootedTBR (TBR), [44](#)
- RootedTBR(), [45](#)
- RootedTBRSwap, [24](#)
- RootedTBRSwap (TBR), [44](#)
- SetMorphyWeights, [32, 34, 39, 40, 43, 46](#)
- SingleCharMorphy, [32, 34, 39, 39, 43, 46](#)
- SPR, [33, 40, 45](#)
- SPRSwap (SPR), [40](#)
- SPRSwap(), [24, 30, 37](#)
- SuccessiveApproximations, [42](#)
- summary.morphyPtr, [32, 34, 39, 40, 43, 46](#)
- TBR, [33, 42, 44](#)
- TBRMoves (TBR), [44](#)
- TBRSwap (TBR), [44](#)
- TipsAreColumns, [24, 37](#)
- TreeSearch(), [7, 26](#)
- TreeTools::ConsensusWithout(), [25](#)
- UnloadMorphy, [32, 34, 39, 40, 43, 46](#)
- WithOneExtraStep, [47](#)