

Package ‘treeDbalance’

May 8, 2026

Title Computation of 3D Tree Imbalance

Version 1.2.0

Maintainer Sophie Kersting <sophie.kersting@uni-greifswald.de>

Description The main goal of the R package 'treeDbalance' is to provide functions for the computation of several measurements of 3D node imbalance and their respective 3D tree imbalance indices, as well as to introduce the new 'phylo3D' format for rooted 3D tree objects. Moreover, it encompasses an example dataset of 3D models of 63 beans in 'phylo3D' format. Please note that this R package was developed alongside the project described in the manuscript 'Measuring 3D tree imbalance of plant models using graph-theoretical approaches' by M. Fischer, S. Kersting, and L. Kühn (2023) <[doi:10.48550/arXiv.2307.14537](https://doi.org/10.48550/arXiv.2307.14537)>, which provides precise mathematical definitions of the measurements. Furthermore, the package contains several helpful functions, for example, some auxiliary functions for computing the ancestors, descendants, and depths of the nodes, which ensures that the computations can be done in linear time, or functions that convert existing formats of 3D tree models of other software into the 'phylo3D' format. Moreover, it comprises functions to extract the graph-theoretical topology without vertices of in- and out-degree 1 of rooted 3D trees as well as to adapt node enumerations to the common 'phylo' format. Most functions of 'treeDbalance' require as input a rooted tree in the 'phylo3D' format, an extended 'phylo' format (as introduced in the R package 'ape' 1.9 in November 2006). Such a 'phylo3D' object must have at least two new attributes next to those required by the 'phylo' format: 'node.coord', the coordinates of the nodes, as well as 'edge.weight', the literal weight or volume of the edges. Optional attributes are 'edge.diam', the diameter of the edges, and 'edge.length', the length of the edges. For visualization purposes one can also specify 'edge.type', which ranges from normal cylinder to bud to leaf, as well as 'edge.color' to change the color of the edge depiction. This project was supported by the joint research project DIG-IT! funded by the European Social Fund (ESF), reference: ESF/14-BM-A55-0017/19, and the Ministry of Education, Science and Culture of Mecklenburg-Western Pomerania, Germany, as well as by

the project ArtIGROW, which is a part of the WIR!-Alliance 'ArtIFARM – Artificial Intelligence in Farming' funded by the German Federal Ministry of Education and Research (FKZ: 03WIR4805).

License GPL-3

Depends R (>= 3.5.0)

Imports rgl, R.matlab

Encoding UTF-8

LazyData true

NeedsCompilation no

RoxygenNote 7.3.3

Author Mareike Fischer [aut] (ORCID: <<https://orcid.org/0000-0002-9429-0859>>),
 Sophie Kersting [aut, cre] (ORCID:
 <<https://orcid.org/0000-0002-1038-9246>>),
 Luise Kühn [aut] (ORCID: <<https://orcid.org/0000-0002-0158-8286>>),
 Jule Möller [ctr]

Repository CRAN

Date/Publication 2026-02-24 18:40:02 UTC

Contents

all3DImbalIndices	3
combined3DIndex	6
enum2depthwise	7
example3Dtrees	9
extendPhylo	10
extractTopology	12
imbalProfile	13
imbalSubdiv_A	14
imbalSubdiv_mu	16
makePhylo3DBalanced	17
plotPhylo3D	18
print.phylo3D	21
qsm2phylo3D	22
sketchPhylo3D	23

Index

25

all3DImbalIndices	<i>Calculation of integral-based distance- and angle-based 3D imbalance indices</i>
-------------------	---

Description

all3DImbalIndices - This serves as a wrapper function to calculate a 3D imbalance index value of a 3D tree in phylo3D format according to the specified imbalance measurement and weighting scheme.

If problems occur with the estimation of the integrals, try to increase the optional parameters `rel.tolerance` and `max.subdiv`.

A_Index - Calculates the 3D imbalance index "weighted integral-based centroid angle" of a 3D tree in phylo3D format using either the edge weights or the edge lengths as weights.

If problems occur with the estimation of the integrals, try to increase the optional parameters `rel.tolerance` and `max.subdiv`.

alpha_Index - Calculates the 3D imbalance index "weighted integral-based minimal centroid angle" of a 3D tree in phylo3D format using either the edge weights or the edge lengths as weights.

If problems occur with the estimation of the integrals, try to increase the optional parameters `rel.tolerance` and `max.subdiv`.

M_Index - Calculates the 3D imbalance index "weighted integral-based expanded relative centroid distance" of a 3D tree in phylo3D format using either the edge weights or the edge lengths as weights.

If problems occur with the estimation of the integrals, try to increase the optional parameters `rel.tolerance` and `max.subdiv`.

mu_Int - Calculates the 3D imbalance index "weighted integral-based relative centroid distance" of a 3D tree in phylo3D format using either the edge weights or the edge lengths as weights.

If problems occur with the estimation of the integrals, try to increase the optional parameters `rel.tolerance` and `max.subdiv`.

imbalInt_e - Calculates the integral of the node imbalance values over all possible subdividing nodes on an edge.

imbalProfile_e - Calculates the node imbalance values for a given set of subdivisions of an edge.

Usage

```
all3DImbalIndices(
  tree,
  imbal_type,
  weight = "edge_weight",
  rel.tolerance = 1e-10,
  max.subdiv = 200L
)
```

```
A_Index(tree, weight = "edge_weight", rel.tolerance = 1e-10, max.subdiv = 200L)
```

```
alpha_Index(
```

```

    tree,
    weight = "edge_weight",
    rel.tolerance = 1e-10,
    max.subdiv = 200L
)

M_Index(tree, weight = "edge_weight", rel.tolerance = 1e-10, max.subdiv = 200L)

mu_Index(
  tree,
  weight = "edge_weight",
  rel.tolerance = 1e-10,
  max.subdiv = 200L
)

imbalInt_e(
  p,
  v,
  centr_v,
  centr_v_weight,
  edge_weight,
  imbal_type,
  rel.tolerance = 1e-08,
  max.subdiv = 200L
)

imbalProfile_e(xs, p, v, centr_v, centr_v_weight, edge_weight, imbal_type)

```

Arguments

tree	A rooted tree in phylo3D format (no special node enumeration required, except that nodes are numbered from 1 to $ V $ = the total number of nodes). There must be at least 2 nodes, i.e., one edge. The attributes 'node.coord' and 'edge.weight' are strictly required.
imbal_type	Specifies which node imbalance measurement should be used. Available are: "A" - centroid angle "alpha" - minimal centroid angle "M" - expanded relative centroid distance "mu" - relative centroid distance
weight	Specifies how the node imbalance values should be weighted. Available weighting methods are: "edge_weight" (default) -> Imbalance with regards to the total edge weight. "edge_length" -> Imbalance with regards to the total edge length.
rel.tolerance	(Optional) Numeric value which specifies the relative tolerance which should be used for estimating the integral using stats::integrate. Set to 1e-10 by default (the stats::integrate default value is approx 3e-16).
max.subdiv	(Optional) Integer value which specifies the maximal number of interval subdivisions for estimating the integral using stats::integrate. Set to 200 by default

	(stats::integrate default value 100).
p	Numeric vector of size 3 (3D coordinates of parent node p).
v	Numeric vector of size 3 (3D coordinates of node v).
centr_v	Numeric vector of size 3 (3D coordinates of the centroid of the pending subtree of node v).
centr_v_weight	Numeric value ≥ 0 (weight of the pending subtree of node v).
edge_weight	Numeric value ≥ 0 (weight of the edge (p, v)).
xs	Numeric vector with values between 0 (included) and 1 (excluded). Set of edge subdivisions.

Value

all3DImbalIndices Numeric value indicating the internal 3D imbalance according to the chosen method.

A_Index Numeric value in the interval between 0 (included) and π (excluded). A value near π indicates a higher degree and near 0 a lower degree of asymmetry.

alpha_Index Numeric value in the interval between 0 (included) and $\pi/2$ (excluded). A value near $\pi/2$ indicates a higher degree and near 0 a lower degree of asymmetry.

M_Index Numeric value in the interval between 0 (included) and 1 (excluded). A value near 1 indicates a higher degree and near 0 a lower degree of asymmetry.

mu_Index Numeric value in the interval between 0 (included) and 1 (excluded). A value near 1 indicates a higher degree and near 0 a lower degree of asymmetry.

imbalInt_e Numeric value (0 minimal value, higher values indicate a higher degree of asymmetry).

imbalProfile_e Numeric vector of imbalance values (0 minimal value, higher values indicate a higher degree of asymmetry) for the edge subdivisions indicated by input xs.

Author(s)

Sophie Kersting

Examples

```
tree <- treeDbalance::extendPhylo(treeDbalance::example3Dtrees$bean09)
all3DImbalIndices(tree, imbal_type = "A", weight = "edge_length")
A_Index(tree, weight = "edge_weight")
alpha_Index(tree)
M_Index(tree, weight = "edge_length")
mu_Index(tree, weight = "edge_length")
imbalInt_e(
  p = c(0, 0, 1), v = c(0, 0, 0), centr_v = c(0.5, 0, 0),
  centr_v_weight = 1, edge_weight = 1, imbal_type = "mu"
)
imbalProfile_e(
  xs = c(0, 0.2, 0.4), p = c(1, 1, 0), v = c(0, 0, 0), centr_v = c(0.5, 0, 0),
  centr_v_weight = 1, edge_weight = 1, imbal_type = "A"
)
```

 combined3DIndex

Calculation of combined 3D imbalance indices

Description

combined3DIndex - Calculates either the pure root imbalance value with regard to a specified vertical axis or the combined 3D imbalance index value of a 3D tree in phylo3D format. The latter is a weighted mean of the integral-based 3D imbalance index value (i.e., A_Index, alpha_Index, M_Index, or mu_Index with edge length or edge weight based weighting) as well as the root imbalance value.

Usage

```
combined3DIndex(
  tree,
  imbal_type,
  weight = "edge_weight",
  root_weight_factor = 1,
  vertical_axis = c(0, 0, 1)
)
```

Arguments

tree	A rooted tree in phylo3D format (no special node enumeration required, except that nodes are numbered from 1 to $ V $ = the total number of nodes). There must be at least 2 nodes, i.e., one edge. The attributes 'node.coord' and 'edge.weight' are strictly required.
imbal_type	Specifies which node imbalance measurement should be used. Available are: "A" - centroid angle "alpha" - minimal centroid angle "M" - expanded relative centroid distance "mu" - relative centroid distance
weight	Specifies how the node imbalance values should be weighted. Available weighting methods are: "edge_weight" (default) -> Imbalance with regards to the total edge weight. "edge_length" -> Imbalance with regards to the total edge length.
root_weight_factor	Numeric value >0 (default 1), which specifies the weight of the root imbalance value in the weighted mean. For example, a weight of 1 means that both the imbalance index value of the whole tree as well as the root imbalance value contribute equally, i.e., the unweighted mean of the two values is returned. For a larger value, the influence of the root imbalance value increases. If the weight is set to infinity (Inf), then the pure root imbalance value is returned.

`vertical_axis` Numeric vector of length 3 (default (0,0,1)), which specifies the given vertical axis for the given tree model. For example, use the default (0,0,1) for models that grow straight upwards (e.g., trees) and (0,0,-1) for models that grow downwards (e.g., roots). The vector (0,0,0) is not allowed.

Value

`combined3DIndex` Numeric value in the interval between 0 (included) and 1 (excluded). A value near 1 indicates a higher degree and near 0 a lower degree of asymmetry.

Author(s)

Sophie Kersting

Examples

```
tree <- treeDbalance::extendPhylo(treeDbalance::example3Dtrees$bean09)
combined3DIndex(tree,
  imbal_type = "A", weight = "edge_weight",
  root_weight_factor = 2, vertical_axis = c(0, 0, 1)
)
combined3DIndex(tree,
  imbal_type = "A", root_weight_factor = Inf,
  vertical_axis = c(0, 0, 1)
)
```

enum2depthwise

Change the node enumeration of trees in phylo(3D) format

Description

`enum2depthwise` - Changes the enumeration of the nodes of a tree to a depthwise (top-down) enumeration that complies with the phylo format's requirements.

The input tree must have its nodes enumerated with 1,...,|V| (with |V| being the total number of nodes). `enum2depthwise` changes the node enumeration s.t. it fulfills the other enumeration requirements of the phylo format like:

- leaves are enumerated with 1,...,n, (with n being the number of leaves)
- the root has number n+1,
- and the rule for inner nodes: descendants have higher numbers than their ancestors.

The edge enumeration and order remains untouched. The function works for phylo as well as phylo3D objects and possible extensions like 'node.ancs', 'node.descs', etc. remain intact.

`enum2cladewise` - Changes the enumeration of the nodes of a tree to a cladewise enumeration (i.e. starting from the root we follow the rule: Go to left child; if that doesn't exist or was already visited go to the right child) that complies with the phylo format's requirements.

The input tree must have its nodes enumerated with 1,...,|V| (with |V| being the total number of nodes). `enum2depthwise` changes the node enumeration s.t. it fulfills the other enumeration requirements of the phylo format like:

- leaves are enumerated with 1,...,n, (with n being the number of leaves)

- the root has number $n+1$,
 - and the rule for inner nodes: descendants have higher numbers than their ancestors.
 The edge enumeration and order remains untouched. The function works for phylo as well as phylo3D objects and possible extensions like 'node.ancs', 'node.descs', etc. remain intact.

enum2_1toV - Changes the enumeration of the nodes of a tree such that they are enumerated with $1, \dots, |V|$ (with $|V|$ being the total number of nodes). Then enum2cladewise or enum2depthwise can be used to switch to a specific enumeration. The edge enumeration and order remains untouched. The input tree must have its nodes enumerated, i.e. the 'edge' attribute must be a numeric matrix (with no missing values). Note that the attribute 'node.coords' is allowed, but then the old node enumeration must point to the correct coordinates. All other rows of the coordinate matrix will be ignored and deleted.

Usage

```
enum2depthwise(tree)
```

```
enum2cladewise(tree)
```

```
enum2_1toV(tree_bad_enum)
```

Arguments

tree	A rooted tree in (extended) phylo(3D) format (no special node enumeration required, except that nodes are numbered from 1 to $ V $ = the total number of nodes). There must be at least 2 nodes, i.e. one edge.
tree_bad_enum	A rooted tree in a rough phylo(3D) format (no special node enumeration required). There must be at least 2 nodes, i.e. one edge.

Value

enum2depthwise(tree) Tree in (extended) phylo(3D) format as before but with node enumeration conforming to the phylo format standard.

enum2cladewise(tree) Tree in (extended) phylo(3D) format as before but with node enumeration conforming to the phylo format standard.

enum2_1toV(tree) Tree in format as before but with nodes enumerated from 1 to $|V|$.

Author(s)

Sophie Kersting

Examples

```
tree <- treeDbalance::example3Dtrees$bean21
# Visualize enumeration: plotPhylo3D(tree, show_node_enum = TRUE)
tree_depthwise <- enum2depthwise(tree)
# Visualize: addPhylo3D(tree_depthwise, offset=c(2,0,0), show_node_enum = TRUE)
tree_cladewise <- enum2cladewise(tree)
# Visualize: addPhylo3D(tree_cladewise, offset=c(4,0,0), show_node_enum = TRUE)
bad_tree <- list(
```

```
edge = matrix(c(1, 13, 1, -7, 13, 4), byrow = TRUE, ncol = 2),
tip.label = c("", ""), Nnode = 2
)
better_tree <- enum2_1toV(bad_tree)
better_tree$edge
bad_3Dtree <- list(
  edge = matrix(c(1, 2, 1, 5, 2, 4), byrow = TRUE, ncol = 2),
  tip.label = c("", ""), Nnode = 2,
  node.coord = matrix(c(
    1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4,
    5, 5, 5, 6, 6, 6
  ), byrow = TRUE, ncol = 3),
  edge.weight = c(1, 1, 1)
)
better_3Dtree <- enum2_1toV(bad_3Dtree)
better_3Dtree$edge
better_3Dtree$node.coord
```

example3Dtrees

Examples of rooted 3D trees

Description

Contains the 3D models of 63 beans in phylo3D format. The single models can be accessed with their IDs, e.g. "bean09" for the bean with ID 9. Note that all IDs from 01 to 66 are available except IDs 8, 29, and 60.

Usage

```
data(example3Dtrees)
```

Format

list

Examples

```
data(example3Dtrees)
treeDbalance::example3Dtrees$bean09
```

Description

extendPhylo - Extends a tree in phylo or phylo3D format, i.e., adds or updates several useful attributes of the tree that facilitate various computations and allow it to be independent of a certain node enumeration. These are: information on descendants, ancestors, and depths as well as on the centroids of all pending subtrees. The latter is only computed if the tree is in phylo3D format and as such contains the attributes 'node.coord' and 'edge.weight'.

getDescs - Creates a matrix with two rows, the second contains in the i -th entry the index of the first row in which the descendants of node i start. Use the function getChildren to quickly retrieve the direct descendants of each node.

getChildren - Creates a vector containing the direct children of a node. If the method indicates that also edges should be returned, this function will also return the number (identifier) of the incoming edge of each child.

getDescendants - Creates a vector containing all descendants of a node.

getAncs - Creates a matrix that contains the parent (direct ancestor) of node i as well as the corresponding edge number in column i .

getNodeDepths - Creates a matrix with three rows: The first contains the nodes ordered by increasing depth. The second contains the indices at which the next depth starts in the first row, i.e., these first two rows are similar to the output matrix of getDescs. The last row contains the depth of each node.

getNodeAtDepth - Creates a vector containing the nodes at a certain depth.

getLeaves - Creates a logical vector that indicates if the i -th node is a leaf.

getSubtrCentr - Calculates the centroid of each pending subtree. Returns a matrix containing the 3D coordinates (3 columns) where row i gives the position of the centroid of T_i , the pending subtree rooted in node i .

getDistFromRoot - Creates a vector containing the length of the path from the node to the root, i.e., the sum of the corresponding edge lengths.

getDistFromLeaf - Creates a vector containing the length of the path from the node to the nearest descendant leaf, i.e., the sum of the corresponding edge lengths.

getIncEdgeLens - Returns the length of the incoming edge of every node as a numeric vector.

getIncEdgeWeights - Returns the weight of the incoming edge of every node as a numeric vector.

Usage

```
extendPhylo(tree)
```

```
getDescs(tree)
```

```
getChildren(tree, node, method = "onlyNodes")
```

```

getDescendants(tree, node)

getAncs(tree)

getNodeDepths(tree)

getNodesAtDepth(tree, depth)

getLeaves(tree)

getSubtrCentr(tree)

getDistFromRoot(tree)

getDistFromLeaf(tree)

getIncEdgeLens(tree)

getIncEdgeWeights(tree)

```

Arguments

tree	A rooted tree in phylo3D format (no special node enumeration required, except that nodes are numbered from 1 to $ V $ = the total number of nodes). There must be at least 2 nodes, i.e., one edge. The attributes 'node.coord' and 'edge.weight' are strictly required.
node	Numeric/integer value representing a node of the tree.
method	A string specifying if only descending nodes or also descending edges should be returned. Can be one of 'onlyNodes' or 'alsoEdges'.
depth	An integer value representing the depth of interest in the tree. A depth of 0 indicates the root layer, 1 the layer of its children, and so forth.

Value

extendPhylo Tree in extended phylo(3D) format, i.e., phylo(3D) format with further attributes.

getDescs Numeric matrix with 2 rows.

getChildren Depending on the method the function either returns an integer vector containing the direct descendants of a node or an integer matrix with two rows, the first containing the direct descendants and the second the corresponding incoming edges.

getAncs Integer matrix with 2 rows. The first row contains the direct ancestor of each node, the second row the incoming edge of this node, i.e., the edge that leads to its ancestor.

getNodeDepths Numeric matrix with 3 rows.

getNodesAtDepth Integer/numeric vector containing all nodes at the desired depth.

getLeaves Creates a logical vector that indicates if the i -th node is a leaf, TRUE for leaf and FALSE for interior node.

getSubtrCentr Numeric matrix with 2 columns.

getDistFromRoot Integer/numeric vector containing the length of the path from each node to the root.

getDistFromLeaf Integer/numeric vector containing the length of the path from each node to its nearest descendant leaf.

getIncEdgeLens Numeric vector containing the length of the incoming edge of each node, i.e., the length of the edge from its direct ancestor to the node itself.

getIncEdgeWeights Numeric vector containing the weight of the incoming edge of each node, i.e., the weight of the edge from its direct ancestor to the node itself.

Author(s)

Sophie Kersting

Examples

```
tree <- treeDbalance::example3Dtrees$bean09
ext_tree <- extendPhylo(tree)
getDescs(tree)
getChildren(ext_tree, 3, method = "alsoEdges")
getDescendants(ext_tree, 3)
getAncs(tree)
getNodeDepths(tree)
getNodesAtDepth(tree, 4)
getLeaves(tree)
getSubtrCentr(ext_tree)
getDistFromRoot(ext_tree)
getDistFromLeaf(ext_tree)
getIncEdgeLens(tree)
getIncEdgeWeights(tree)
```

extractTopology

Extract the non-3D topology of trees in phylo(3D) format

Description

extractTopology - Extracts the topology of a rooted tree in phylo or phylo3D format by ignoring edge subdivisions, i.e. nodes with both in- and out-degree of 1. The nodes have to be enumerated from 1 to $|V|$ (otherwise use enum2_1toV()).

The function works for trees in phylo as well as phylo3D format as existing information on the coordinates or edge weights is dropped anyway.

Edge lengths in the topology are based on the sum of the edge lengths of the corresponding paths in the original tree.

Usage

```
extractTopology(tree)
```

Arguments

tree A rooted tree in (extended) phylo(3D) format (no special node enumeration required, except that nodes are numbered from 1 to $|V|$ = the total number of nodes). There must be at least 2 nodes, i.e. one edge.

Value

extractTopology(tree) Tree in phylo format with cladewise node enumeration in which all interior nodes have at least out-degree 2.

Author(s)

Sophie Kersting, Luise Kühn

Examples

```
tree <- treeDbalance::example3Dtrees$bean09
top <- extractTopology(tree)
```

imbalProfile

Calculation of 3D imbalance profiles

Description

imbalProfile - Calculates the database for a 3D imbalance profile of a 3D tree in phylo3D format for any of the four node imbalance approaches: relative centroid distance, expanded relative centroid distance, centroid angle, or minimal centroid angle. It is also used as a basis to visualize the imbalance in a 3D plot, i.e., functions plotImbalPhylo3D and addImbalPhylo3D.

The profile is computed with regards to the z-coordinate (height), path length to the root, and path length to the nearest descendant leaf of the nodes.

The accuracy can be determined by defining the maximal section length `max.seclen`, which means that an edge with length l will be subdivided $\text{ceiling}(l/\text{max.seclen})-1$ times into parts of equal length and all subdividing nodes will be analyzed for their node imbalance. For example, an edge (p, v) of length 3 with a maximal section length of 1 would be subdivided twice and would then be evaluated for three points in total: for v itself and for the two subdivisions at $1/3$ and $2/3$ of the edge length.

Usage

```
imbalProfile(tree, imbal_type, max.seclen)
```

Arguments

tree A rooted tree in phylo3D format (no special node enumeration required, except that nodes are numbered from 1 to $|V|$ = the total number of nodes). There must be at least 2 nodes, i.e., one edge. The attributes 'node.coord' and 'edge.weight' are strictly required.

imbal_type	Specifies which node imbalance measurement should be used. Available are: "A" - centroid angle "alpha" - minimal centroid angle "M" - expanded relative centroid distance "mu" - relative centroid distance
max.seclen	Numeric value >0 that specifies the maximal section length.

Value

imbalProfile Numeric matrix with five columns. The rows each represent the values of a single tree node or edge subdivision. The first column contains the z-coordinate (height), the second the root path length, the third the (nearest) descendant leaf path length, and the fourth the imbalance value. The fifth column stores the number of the corresponding edge.

Author(s)

Sophie Kersting

Examples

```
tree <- treeDbalance::extendPhylo(treeDbalance::example3Dtrees$bean09)
imbalProfile(tree, imbal_type = "mu", max.seclen = 1)
```

imbalSubdiv_A

Calculation of the centroid angles

Description

imbalSubdiv_A - Calculates the node imbalance value "centroid angle" of a vertex which subdivides the edge (p, v) at $v + x \cdot (p - v)$ with $x \in [0, 1]$. For example, we can obtain the node imbalance value of v if $x = 0$, and $x = 0.5$ would indicate a subdividing node exactly in the middle of v and p .

Attention: If $x = 1$, this function will not calculate the node imbalance value of p with respect to its incoming edge but with respect to the edge (p, v) itself. This enables us to estimate the node imbalance integrals over the entire edge length.

imbalSubdiv_alpha - Calculates the node imbalance value "minimal centroid angle" of a vertex which subdivides the edge (p, v) at $v + x \cdot (p - v)$ with $x \in [0, 1]$. For example, we can obtain the node imbalance value of v if $x = 0$, and $x = 0.5$ would indicate a subdividing node exactly in the middle of v and p .

Attention: If $x = 1$, this function will not calculate the node imbalance value of p with respect to its incoming edge but with respect to the edge (p, v) itself. This enables us to estimate the node imbalance integrals over the entire edge length.

angle3dVec - Calculates the angle in the interval $[0, \pi]$ between two 3D vectors a and b . Note that the function returns 0 if one entry vector is $(0, 0, 0)$.

Usage

```
imbalSubdiv_A(x, p, v, centr_v, centr_v_weight, edge_weight)

imbalSubdiv_alpha(x, p, v, centr_v, centr_v_weight, edge_weight)

angle3dVec(a, b)
```

Arguments

x	Numeric value $\in [0, 1]$ which indicates where on the edge (p, v) the subdivision takes place: $v + x \cdot (p - v)$.
p	Numeric vector of size 3 (3D coordinates of parent node p).
v	Numeric vector of size 3 (3D coordinates of node v).
centr_v	Numeric vector of size 3 (3D coordinates of the centroid of the pending subtree of node v).
centr_v_weight	Numeric value ≥ 0 (weight of the pending subtree of node v).
edge_weight	Numeric value ≥ 0 (weight of the edge (p, v)).
a	Numeric vector of size 3 (e.g., 3D coordinates).
b	Numeric vector of size 3 (e.g., 3D coordinates).

Value

imbalSubdiv_A Numeric value $\in [0, \pi]$ (higher values indicate a higher degree of asymmetry).

imbalSubdiv_alpha Numeric value $\in [0, \pi/2]$ (higher values indicate a higher degree of asymmetry).

angle3dVec Numeric value in $[0, \pi]$.

Author(s)

Sophie Kersting, Luise Kühn

Examples

```
imbalSubdiv_A(
  x = 0.5, p = c(1, 0, 1), v = c(0, 0, 0), centr_v = c(0.5, 0, 0),
  centr_v_weight = 1, edge_weight = 1
)
imbalSubdiv_alpha(
  x = 0.5, p = c(1, 0, 1), v = c(0, 0, 0), centr_v = c(0.5, 0, 0),
  centr_v_weight = 1, edge_weight = 1
)
angle3dVec(a = c(1, 0, 0), b = c(0, 1, 0)) # right angle = pi/2 = 1.5707...
```

imbalSubdiv_mu

Calculation of the centroid distances

Description

imbalSubdiv_mu - Calculates the node imbalance value "relative centroid distance" of a vertex which subdivides the edge (p, v) at $v + x \cdot (p - v)$ with $x \in [0, 1]$. For example, we can obtain the node imbalance value of v if $x = 0$, and $x = 0.5$ would indicate a subdividing node exactly in the middle of v and p .

Attention: If $x = 1$, this function will not calculate the node imbalance value of p with respect to its own incoming edge but with respect to the edge (p, v) itself. This enables us to estimate the node imbalance integrals over the entire edge length.

imbalSubdiv_M - Calculates the node imbalance value "expanded relative centroid distance" of a vertex which subdivides the edge (p, v) at $v + x \cdot (p - v)$ with $x \in [0, 1]$. For example, we can obtain the node imbalance value of v if $x = 0$, and $x = 0.5$ would indicate a subdividing node exactly in the middle of v and p .

Attention: If $x = 1$, this function will not calculate the node imbalance value of p with respect to its own incoming edge but with respect to the edge (p, v) itself. This enables us to estimate the node imbalance integrals over the entire edge length.

dist3dToLine - Calculates the distance of a *point* to the infinite line between two points a and b in 3D space.

cross3d_prod - Calculates the cross-product of two 3D vectors.

Usage

```
imbalSubdiv_mu(x, p, v, centr_v, centr_v_weight, edge_weight)
```

```
imbalSubdiv_M(x, p, v, centr_v, centr_v_weight, edge_weight)
```

```
dist3dToLine(point, a, b)
```

```
cross3d_prod(a, b)
```

Arguments

<code>x</code>	Numeric value $\in [0, 1]$ which indicates where on the edge (p, v) the subdivision takes place: $v + x \cdot (p - v)$.
<code>p</code>	Numeric vector of size 3 (3D coordinates of parent node p).
<code>v</code>	Numeric vector of size 3 (3D coordinates of node v).
<code>centr_v</code>	Numeric vector of size 3 (3D coordinates of the centroid of the pending subtree of node v).
<code>centr_v_weight</code>	Numeric value ≥ 0 (weight of the pending subtree of node v).
<code>edge_weight</code>	Numeric value ≥ 0 (weight of the edge (p, v)).
<code>point</code>	Numeric vector of size 3 (e.g. 3D coordinates).

- a Numeric vector of size 3 (e.g. 3D coordinates).
 b Numeric vector of size 3 (e.g. 3D coordinates).

Value

imbalSubdiv_mu Numeric value $\in [0, 1]$ (higher values indicate a higher degree of asymmetry).

imbalSubdiv_M Numeric value $\in [0, 2]$ (higher values indicate a higher degree of asymmetry).

dist3dToLine Numeric value.

cross3d_prod Numeric vector of size 3.

Author(s)

Sophie Kersting, Luise Kühn

Examples

```
imbalSubdiv_mu(
  x = 0.5, p = c(1, 0, 1), v = c(0, 0, 0), centr_v = c(0.5, 0, 0),
  centr_v_weight = 1, edge_weight = 1
)
imbalSubdiv_M(
  x = 0.5, p = c(1, 0, 1), v = c(0, 0, 0), centr_v = c(0.5, 0, 0),
  centr_v_weight = 1, edge_weight = 1
)
dist3dToLine(point = c(1, 1, 1), a = c(0, 0, 0), b = c(1, 2, 2)) # 0.47140...
cross3d_prod(a = c(1, -1, 1), b = c(1, 2, 2)) # c(-4, -1, 3)
```

makePhylo3DBalanced *Turn a rooted 3D tree into one of its balanced versions*

Description

makePhylo3DBalanced - Creates a balanced version of a rooted 3D tree in phylo3D format. From the leaves of lowest depth to the root, each node is turned into a balanced node by rotating its pending subtree such that it is in line with its incoming edge. The edge and subtree lengths and weights as well as the angles between the outgoing edges of a single node (the sister edges) are left intact.

Note that this function yields only one of many possible balanced version of the given rooted 3D tree (most often the minimal tree under the aforementioned requirements is not unique).

rotate3dVec - Rotates a vector in 3D space for a given angle and rotation axis.

Usage

```
makePhylo3DBalanced(tree)
```

```
rotate3dVec(angle, axis, vec)
```

Arguments

tree	A rooted tree in phylo3D format (no special node enumeration required, except that nodes are numbered from 1 to $ V $ = the total number of nodes). There must be at least 2 nodes, i.e., one edge. The attributes 'node.coord' and 'edge.weight' are strictly required.
angle	Angle for the rotation.
axis	Rotation axis.
vec	Numeric vector of size 3 (3D coordinates of the vector that shall be rotated).

Value

makePhylo3DBalanced Tree in phylo3D format which is balanced with regards to all four node imbalance measurements and their corresponding imbalance indices.

rotate3dVec Numeric vector of size 3 (3D coordinates of the rotated vector).

Author(s)

Sophie Kersting, Luise Kühn

Examples

```
tree <- treeDbalance::extendPhylo(treeDbalance::example3Dtrees$bean22)
tree_bal <- makePhylo3DBalanced(tree)
rotate3dVec(angle = pi / 2, axis = c(0, -1, 0), vec = c(5, 0, 0)) # approx. (0,0,5)
round(rotate3dVec(angle = pi / 2, axis = c(0, -1, 0), vec = c(5, 0, 0)), 15)
```

plotPhylo3D

Plot a phylo3D object

Description

plotPhylo3D - Plots a phylo3D object using functions of the package 'rgl'.

addPhylo3D - This function plots a phylo3D object without any coordinate axis or adds the tree to an existing plot (e.g., for plotPhylo3D).

plotImbalPhylo3D - Plots a phylo3D object using functions of the package 'rgl'. Moreover, it uses either brightness or a color scale to indicate the imbalance.

Edge sections are shown darker or red with higher degree of imbalance and brighter or cyan if they are balanced. This function does not use the parameter `edge.color`

Attention: Edges of type 'bud' or 'leaf' will always be depicted as balanced, because they should represent leaf edges that are by definition always balanced.

addImbalPhylo3D - This function plots a phylo3D object without any coordinate axis or adds the tree to an existing plot (e.g., for plotImbalPhylo3D). Moreover, it uses either brightness or a color scale to indicate the imbalance.

Edge sections are shown darker or red with higher degree of imbalance and brighter or cyan if they are balanced. This function does not use the parameter `edge.color`.

Attention: Edges of type 'bud' or 'leaf' will always be depicted as balanced because they should represent leaf edges that are always balanced.

Usage

```

plotPhylo3D(tree, show_node_enum = NULL)

addPhylo3D(tree, offset = c(0, 0, 0), show_node_enum = NULL)

plotImbalPhylo3D(
  tree,
  imbal_type,
  max.seclen,
  color.imbal = TRUE,
  show.gradient = FALSE
)

addImbalPhylo3D(
  tree,
  offset = c(0, 0, 0),
  imbal_type,
  max.seclen,
  color.imbal = TRUE,
  show.gradient = FALSE
)

```

Arguments

- | | |
|----------------|--|
| tree | <p>A rooted tree in phylo3D format (no special node enumeration required, except that nodes are numbered from 1 to V = the total number of nodes). There must be at least 2 nodes, i.e., one edge. The attributes 'node.coord' and 'edge.weight' are strictly required.</p> <p>Optional: Add attribute 'edge.type' and/or 'edge.color' (character vectors e.g., <code>c("d","b","l","l")</code> and <code>c("red","blue", "red", "gray30")</code>) to the phylo3D object to plot the edges differently.</p> <p>The i-th entry of 'edge.type' has to contain the type of the i-th edge. The following types are available: "d"= default cylinder, "b" = bud, and "l"=leaf. If 'edge.type' is not given, every edge is by default visualized as a cylinder. Similar for the attribute 'edge.color'. If it is not given, every edge is depicted as "gray30" by default.</p> <p>Furthermore, the attribute 'edge.diam' can also be added optionally to set the edge diameters. Otherwise, the edge diameters will be calculated based on the edge lengths and the edge weights (treated as volume).</p> |
| show_node_enum | <p>Character (vector) that specifies which node numbers should be depicted (none by default, NULL). This helps to identify specific nodes and edges. Available are the following options (can be combined, e.g. <code>c("leaf_numbers", "inner_numbers")</code>).</p> <p>Use "leaf_numbers" to display the number of each leaf.</p> <p>Use "leaf_labels" to display the label of each leaf.</p> <p>Use "inner_numbers" to display the number of each interior node.</p> |
| offset | <p>Numeric vector of length 3, contains 3D coordinates by which the phylo object should be shifted (default = <code>c(0,0,0)</code>, i.e., no shift).</p> |

imbal_type	Specifies which node imbalance measurement should be used. Available are: "A" - centroid angle "alpha" - minimal centroid angle "M" - expanded relative centroid distance "mu" - relative centroid distance
max.seclen	Numeric value >0 that specifies the maximal section length.
color.imbal	Boolean value (default TRUE). If true, colors are used to depict the imbalance. Otherwise, a grayscale image is produced.
show.gradient	Boolean value (default FALSE). If true the color or grayscale gradient is depicted.

Value

plotPhylo3D No return value, called for side effects (plotting).

addPhylo3D No return value, called for side effects (plotting).

plotImbalPhylo3D No return value, called for side effects (plotting).

addImbalPhylo3D No return value, called for side effects (plotting).

Author(s)

Sophie Kersting

Examples

```
tree <- list(
  edge = matrix(c(1, 2, 2, 3), byrow = TRUE, ncol = 2),
  tip.label = "", Nnode = 2,
  node.coord = matrix(c(0, 0, 0, 1, 2, 2, 3, 1, 3),
    byrow = TRUE,
    ncol = 3
  ),
  edge.weight = c(0.5, 0.25), edge.type = c("d", "l")
)
class(tree) <- "phylo3D"
# Alternatively try: tree <- treeDbalance::example3Dtrees$bean09
plotPhylo3D(tree, show_node_enum = "leaf_numbers")
addPhylo3D(tree, offset = c(1, 1, 0))
plotImbalPhylo3D(tree,
  imbal_type = "mu", max.seclen = 0.5, color.imbal = TRUE,
  show.gradient = FALSE
)
addImbalPhylo3D(tree,
  imbal_type = "mu", offset = c(1, 0, 0), max.seclen = 0.5,
  color.imbal = FALSE, show.gradient = FALSE
)
```

print.phylo3D *Print, summary and plot methods for phylo3D objects*

Description

print.phylo3D prints the contents of an object of class phylo3D. It provides a brief summary of the object structure and its contents.

plot.phylo3D creates a 2D projection plot of a phylo3D object, using the X and Z coordinates of each node. Each edge is drawn as a line segment between parent and child nodes.

For more elaborate 3D visualizations use plotPhylo3D() or addPhylo3D(), plotImbalPhylo3D() or addImbalPhylo3D(), or sketchPhylo3D().

summary.phylo3D provides a summary of an object of class phylo3D. It offers a high-level overview of the contents and their structure.

Usage

```
## S3 method for class 'phylo3D'
print(x, ...)

## S3 method for class 'phylo3D'
plot(x, edge_col = "black", edge_lwd_scale = 1, show_leaf_label = FALSE, ...)

## S3 method for class 'phylo3D'
summary(object, ...)
```

Arguments

x	An object of class phylo3D.
...	Additional arguments passed to plot().
edge_col	Color for the edges (default "black").
edge_lwd_scale	Scaling factor (default 1) for line width for the edges.
show_leaf_label	Logical (default FALSE), specifying if tip labels should be depicted.
object	An object of class phylo3D.

Value

print.phylo3D The input phylo3D object, returned invisibly. Called primarily for its side effect of printing a human-readable summary to the console.

plot.phylo3D The input phylo3D object, returned invisibly. This function is called for its side effect of displaying a plot.

summary.phylo3D An object of class summary.phylo3D, which contains summary information about the phylo3D object.

Examples

```
# Printing a 'phylo3D' object:
tree <- treeDbalance::extendPhylo(treeDbalance::example3Dtrees$bean09)
tree # same as print(tree)
# Simple plot of a 'phylo3D' object:
plot(tree, edge_col = "blue", show_leaf_label = TRUE)
# Summary of a 'phylo3D' object:
summary(tree)
str(summary(tree))
```

qsm2phylo3D

Convert QSM format to a phylo3D object

Description

qsm2phylo3D - Wrapper function that converts the output .mat-file of the software treeQSM (version 2.0 or 2.4.x) to a rooted tree in phylo3D format.

qsmList2phylo3D - Converts a list containing QSM data to a rooted tree in phylo3D format.

Usage

```
qsm2phylo3D(file, version = "2.4.x", setConnection2zero = TRUE)
```

```
qsmList2phylo3D(qsm_list, setConnection2zero = TRUE)
```

Arguments

file	File name (with path if not located in working directory), e.g.: "C:/...path.../myMatlabQSMfile.mat"
version	Specifies the version of the treeQSM .mat-file. Available are - "2.0" for version 2.0 and - "2.4.x" for version 2.4.1 and 2.4.2.
setConnection2zero	Logical value (default TRUE) indicating if the width of the edges, which connect two consecutive cylinders, should be set to zero. If FALSE, then the radius is set to the radius of the "child" cylinder.
qsm_list	List containing the QSM-data, with the attributes "radius", "length", "start", "axis", "parent" and "extension".

Value

qsm2phylo3D(file) A rooted tree in phylo3D format.

qsmList2phylo3D(qsmList) A rooted tree in phylo3D format.

Author(s)

Sophie Kersting

Examples

```
mat_file <- system.file("extdata", "exampleQSM.mat",
  package = "treeDbalance"
)
new_phylo <- qsm2phylo3D(file = mat_file)
new_phylo <- qsmList2phylo3D(NULL)
```

sketchPhylo3D

Sketch a phylo3D object

Description

sketchPhylo3D - Sketches a phylo3D object using functions of the package 'rgl'. This is a much faster but rougher version of plotPhylo3D as the resulting sketch is using lines instead of cylinders and the user can specify in various ways that only a portion of the edges will be depicted. This is intended to get a quick overview of a phylo3D object that consists of thousands of edges and would therefore take very long to plot regularly.

addSketchPhylo3D - This function sketches a phylo3D object without any coordinate axis or adds the tree to an existing sketch (e.g. for sketchPhylo3D). This is a much faster but rougher version of addPhylo3D as the resulting sketch is using lines instead of cylinders and the user can specify in various ways that only a portion of the edges will be depicted. This is intended to get a quick overview of a phylo3D object that consists of thousands of edges and would therefore take very long to plot regularly.

Usage

```
sketchPhylo3D(
  tree,
  show_node_enum = FALSE,
  ratio_of_edges = 1,
  number_of_edges = NULL,
  min_edge_diam = NULL,
  lwd_factor = 100
)
```

```
addSketchPhylo3D(
  tree,
  offset = c(0, 0, 0),
  show_node_enum = FALSE,
  ratio_of_edges = 1,
  number_of_edges = NULL,
  min_edge_diam = NULL,
  lwd_factor = 100
)
```

Arguments

tree	<p>A rooted tree in phylo3D format (no special node enumeration required, except that nodes are numbered from 1 to V = the total number of nodes). There must be at least 2 nodes, i.e. one edge. The attributes 'node.coord' and 'edge.weight' are strictly required.</p> <p>Optional: Add attribute 'edge.color' (character vector, e.g. c("red", "blue", "red", "gray30", ...)) to the phylo3D object to sketch the edges differently. If it is not given, every edge is depicted as "gray30" by default.</p> <p>The attribute 'edge.diam' can be added optionally to set the edge diameters, otherwise the edge diameters will be calculated based on the edge lengths and the edge weights (treated as volume).</p>
show_node_enum	A boolean value (default FALSE). If true each node of the visualized phylo3D object is marked with its number. This helps to identify specific nodes and edges.
ratio_of_edges	A numeric value (>0 and ≤ 1 , default 1) that indicates the ratio of edges that should be visualized (beginning with the widest edges). This can be ignored, i.e. set to NULL, if 'number_of_edges' or 'min_edge_diam' is set. If the parameters are conflicting: 'ratio_of_edges' trumps 'number_of_edges' trumps 'min_edge_diam'.
number_of_edges	An integer value (>0 , default NULL) indicating how many of the edges should be visualized (beginning with the widest edges). This can be ignored, i.e. left at NULL, if 'ratio_of_edges' or 'min_edge_diam' is set. If the parameters are conflicting: 'ratio_of_edges' trumps 'number_of_edges' trumps 'min_edge_diam'.
min_edge_diam	A numeric value (default NULL) that indicates that only edges with equal or larger diameter should be visualized. This can be ignored, i.e. left at NULL, if 'number_of_edges' is set. If the parameters are conflicting: 'ratio_of_edges' trumps 'number_of_edges' trumps 'min_edge_diam'.
lwd_factor	A numeric value (>0 , default 100) to increase or decrease the line width.
offset	Numeric vector of length 3, contains 3D coordinates by which the phylo object should be shifted (default = c(0,0,0), i.e. no shift).

Author(s)

Sophie Kersting

Examples

```
tree <- treeDbalance::example3Dtrees$bean09
sketchPhylo3D(tree, show_node_enum = TRUE, ratio_of_edges = 0.8)
addSketchPhylo3D(tree,
  offset = c(1, 1, 0), ratio_of_edges = NULL,
  number_of_edges = 200
)
```

Index

* datasets

- example3Dtrees, 9
- A_Index (all3DImbalIndices), 3
- addImbalPhylo3D (plotPhylo3D), 18
- addPhylo3D (plotPhylo3D), 18
- addSketchPhylo3D (sketchPhylo3D), 23
- all3DImbalIndices, 3
- alpha_Index (all3DImbalIndices), 3
- angle3dVec (imbalSubdiv_A), 14
- combined3DIndex, 6
- cross3d_prod (imbalSubdiv_mu), 16
- dist3dToLine (imbalSubdiv_mu), 16
- enum2_1toV (enum2depthwise), 7
- enum2cladewise (enum2depthwise), 7
- enum2depthwise, 7
- example3Dtrees, 9
- extendPhylo, 10
- extractTopology, 12
- getAncs (extendPhylo), 10
- getChildren (extendPhylo), 10
- getDescendants (extendPhylo), 10
- getDescs (extendPhylo), 10
- getDistFromLeaf (extendPhylo), 10
- getDistFromRoot (extendPhylo), 10
- getIncEdgeLens (extendPhylo), 10
- getIncEdgeWeights (extendPhylo), 10
- getLeaves (extendPhylo), 10
- getNodeDepths (extendPhylo), 10
- getNodesAtDepth (extendPhylo), 10
- getSubtrCentr (extendPhylo), 10
- imbalInt_e (all3DImbalIndices), 3
- imbalProfile, 13
- imbalProfile_e (all3DImbalIndices), 3
- imbalSubdiv_A, 14
- imbalSubdiv_alpha (imbalSubdiv_A), 14
- imbalSubdiv_M (imbalSubdiv_mu), 16
- imbalSubdiv_mu, 16
- M_Index (all3DImbalIndices), 3
- makePhylo3DBalanced, 17
- mu_Index (all3DImbalIndices), 3
- plot.phylo3D (print.phylo3D), 21
- plotImbalPhylo3D (plotPhylo3D), 18
- plotPhylo3D, 18
- print.phylo3D, 21
- qsm2phylo3D, 22
- qsmList2phylo3D (qsm2phylo3D), 22
- rotate3dVec (makePhylo3DBalanced), 17
- sketchPhylo3D, 23
- summary.phylo3D (print.phylo3D), 21