Package 'treeDbalance'

July 22, 2025

Title Computation of 3D Tree Imbalance

Version 1.0.1

Author Mareike Fischer [aut],
Sophie Kersting [aut, cre],
Luise Kühn [aut],
Jule Möller [ctr]

Maintainer Sophie Kersting <sophie_kersting@gmx.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>, 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.
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

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).

Contents

---

all3DImbalIndices	*Calculation of integral-based distance- and angle-based 3D imbalance indices*

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 an-
gle" 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))
```

---

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
```

---

extendPhylo	*Useful extensions to the phylo format*

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.

`getNodesAtDepth` - 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)
```

---

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 ceiling(l/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

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)).
point	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 = FALSE)

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

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   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.

Optional: Add attribute 'edge.type' and/or 'edge.color' (character vectors e.g., c("d","b","l","l") and c("red","blue", "red", "gray30")) to the pylo3D object to plot the edges differently.

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.

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).

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.

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).

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 = FALSE)
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)
```

Index