Package ‘rase’

August 29, 2016

Title Range Ancestral State Estimation for Phylogeography and Comparative Analyses

Version 0.2-22

Date 2016-01-13

Depends R (>= 2.14.0), ape (>= 3.1-0)

Imports mvtnorm (>= 1.0-0), polyCub (>= 0.5-0), spatstat (>= 1.36-0), rgl, sm, methods, grDevices, stats,

Suggests sp, ggmcmc, rpanel, tcltk

Description The rase package primarily implements the Range Ancestral State Estimation for phylo-
geography described in Quintero, I., Keil, P., Jetz, W., & Crawford, F. W. (2015). Historical Bio-
geography Using Species Geographical Ranges. Systematic biology, 64(6), 1059-1073. It also includes Bayesian inference of ancestral states under a Brownian Motion model of character evolution and Maximum Likelihood estimation of rase for n-dimen-
sional data. Visualizing functions in 3D are implemented using the rgl package.

License GPL (>= 2)

NeedsCompilation no

Author Ignacio Quintero [aut, cre],
Forrest W. Crawford [aut],
Petr Keil [aut]

Maintainer Ignacio Quintero <ignacio.quintero@yale.edu>

Repository CRAN

Date/Publication 2016-01-13 22:41:22

R topics documented:

rase-package ..................................................... 2
add.dens .......................................................... 3
add.polygons ..................................................... 5
bm_ase ............................................................. 6
data.for.3d ........................................................ 8
name.poly ........................................................... 9
phylo.3d ............................................................ 10
Description
The rase package primarily implements the Range Ancestral State Estimation for phylogeography described in Quintero, I., Keil, P., Jetz, W., Crawford, F. W. 2015 Historical Biogeography Using Species Geographical Ranges. Systematic Biology. doi: 10.1093/sysbio/syv057. It also includes Bayesian inference of ancestral states under a Brownian Motion model of character evolution and Maximum Likelihood estimation of rase for n-dimensional data. Visualizing functions in 3D are implemented using the rgl package.

Details

Package: rase
Type: Package
Version: 0.2-2
Date: 2015-08-12
License: GLP (>=2)

Author(s)
Ignacio Quintero, Forrest Crawford, Petr Keil
Maintainer: Ignacio Quintero <ignacio.quintero@yale.edu>

References

Examples
# Here the application in the paper of Quintero et al.,
# on the Psophia trumpeters
# is shown using rase package.

# load data
data(rase_data, package = 'rase')

## Not run:
# check the data we are going to use
# the phylogenetic tree
psophia_tree

# the GPC polygons of Psophia distribution.
psohia_poly

# Species names of polygons (in order)
pnames <- c('dextralis', 'viridis', 'leucoptera', 'interjecta',
            'obscura', 'crepitans', 'ochroptera', 'napensis')

# name the polygons
psophia_poly <- name.poly(psophia_poly, psophia_tree,
                            poly.names = pnames)

# Run rase for 10 iterations
rase_results <- rase(psophia_tree, psophia_poly, niter = 100)

# Run with higher number of iterations
# rase_results <- rase(psophia_tree, polygons)

# Check the results
str(rase_results)

# post-MCMC handling
rase_results_for_ggmcmc <- post.mcmc(rase_results, burnin=0, thin = 1)

# plot the densities for dispersal rates using ggmcmc
require(ggmcmc)
ggs_traceplot(rase_results_for_ggmcmc, family = 'sigma')

## End(Not run)

---

**add.dens**

*Plot the Posterior Density for each Ancestor in 3D Space*

**Description**

Plots a user-specified credible interval from the posterior density for each ancestor according to `rase` results. It is mapped in 3D space, where the x/y axes are longitude/latitude, and the z axis is time. Although it works by itself, the posterior densities are intended to overlay an existing `phylo.3d` plot (see example).
Usage
add.dens(df3, res, nlevels = 20, z.scale = 1, col = c(1:nnode), ...)

Arguments
df3 an object from data.for.3d.
res the output from a rase run.
nlevels the credible interval to be plotted for every posterior probabilities. The nlevels
is the inverse of the credible interval (CI) to be plotted (e.g., nlevels = 10 will
plot the 90% CI, nlevels = 20 (the default) will plot the 95% CI).
z.scale the scale of the time axis (vertical axis). Effectively, the time variable is multipliers by this number. It is very useful for visualization purposes. It should
 correspond with the z.scale argument in phylo.3d
col vector of colors for the polygons.
... further arguments to be passed to polygon3d.

Author(s)
Ignacio Quintero

References

See Also
data.for.3d, phylo.3d, add.polygons

Examples
# load the mcmc results from rase run, polygons and tree
data(rase_data, package = 'rase')

pnames <- c('dextralis', 'viridis', 'leucoptera', 'interjecta',
'obscura', 'crepitans', 'ochroptera', 'napensis')
psophia_poly <- name.poly(psophia_poly, psophia_tree, poly.names = pnames)

# transform the data for 3d plotting
df3 <- data.for.3d(mcmc, psophia_tree, psophia_poly)

## Not run:
# plot the tree in 3D
phylo.3d(df3, z.scale = 10, pts = TRUE)

# add the polygons representing the tip distributions
add.polygons(df3)

# add the posterior density at each node of the 3d tree
add.polygons

```
if(requireNamespace("rpanel") & requireNamespace("tcltk"))
  add.dens(df3, mcmc, z.scale = 10, col = c(2:8))

## End(Not run)
```

### add.polygons

Plot the Distributions for the Phylogenetic Tips in 3D

**Description**

Plots the polygon surfaces of the tips of the phylogeny. It is mapped in 3D space, where the x/y axes are longitude/latitude, and the z axis is time. Although it works by itself, the polygons are intended to overlay an existing `phylo.3d` plot (see example).

**Usage**

```
add.polygons(df3, axes = 2, ...)
```

**Arguments**

- `df3`: an object from `data.for.3d`.
- `axes`: number of axes to be plotted (1 = 'x', 2 = 'x' & 'y', 3 = 'x', 'y' & 'z').
- `...`: further arguments to be passed to `polygon3d`.

**Details**

The polygons do not have to be non-overlapping, however, when they do, they “flicker”.

**Author(s)**

Petr Keil

**References**


**See Also**

`data.for.3d`, `phylo.3d`, `add.dens`
Examples

```r
# load the mcmc results from rase run, polygons and tree
data(rase_data, package = 'rase')

pnames <- c('dextralis', 'viridis', 'leucoptera', 'interjecta',
            'obscura', 'crepitans', 'ochroptera', 'napensis')
psophia_poly <- name.poly(psophia_poly, psophia_tree, poly.names = pnames)

# transform the data for 3d plotting
df3 <- data.frame(mcmc, psophia_tree, psophia_poly)

## Not run:
# plot the tree in 3D
phylo.3d(df3, z.scale = 10, pts = TRUE)

# add the polygons representing the tip distributions
add.polygons(df3)

# add the posterior density at each node of the 3d tree
if(requireNamespace("rpanel") && requireNamespace("tcltk"))
    add.dens(df3, mcmc, z.scale = 10, col = c(2:8))

## End(Not run)
```

bm_ase

Bayesian Ancestral State Estimation with Points

Description

Performs a bayesian two-dimensional ancestral state estimation with single values as input (contrast with \texttt{rase}) according to a Brownian Motion model of trait evolution (or dispersal for phylogeography). It uses Gibbs sampling to approximate the posterior distribution.

Usage

```
bm_ase(tree, values, niter = 1e3, logevery = 10,
       sigma2_scale = 0.05, screenlog = TRUE, params0 = NA)
```

Arguments

- **tree**: phylogenetic tree of class "phylo".
- **values**: 2-dimensional trait values (e.g., coordinates for phylogeography). Should be a data.frame with two columns named x & y, or a list with two elements named x & y.
- **niter**: number of MCMC iterations. By default \texttt{niter} = 1000.
- **logevery**: iteration cycle to print current iteration. By default \texttt{logevery} = 10.
sigma2_scale optional. Window proposal for sigma2x & sigma2y.
screenlog if TRUE (default), prints current iteration every logevery to the screen.
params0 optional. A vector of initial parameter values in the following order: x ancestors, y ancestors, sigma2x and sigma2y. If params0 = NA (default), an initial Maximum Likelihood optimization using ace provides the starting parameter values.

Value
returns a matrix where every column represents one parameter. The first columns (i.e., nX_x) give the ancestral locations for trait x in the order of nodes in the tree (see the phylo class for details), followed by the ancestral locations of trait y (i.e., nX_y), and the rate parameter in x (sigma2x) and y (sigma2y).

Author(s)
Forrest Crawford, Ignacio Quintero

References

See Also
For the maximum likelihood version see point.like.bm; for the incorporation of polygon uncertainty see rase. For post-mcmc handling see post.mcmc.

Examples

```r
### Create some data to be analyzed

# Number of taxa
ntaxa <- 10

# Known parameters
mean_x <- 0
mean_y <- 0
sigma_x <- 1
sigma_y <- 1

# Create a random tree
tree <- ape::rtree(n = ntaxa)

# Create random data according to tree structure
x_locs <- as.numeric(mvtnorm::rmvnorm(1, rep(mean_x, ntaxa), sigma=sigma_x*vcv(tree)))
y_locs <- as.numeric(mvtnorm::rmvnorm(1, rep(mean_y, ntaxa), sigma=sigma_y*vcv(tree)))
values = list(x = x_locs, y = y_locs)
## Not run:
# run bm_ase for 10 iterations
```
Transform Output for 3D Plotting

**Description**

Takes the output from a *rase* run, the corresponding phylogenetic tree and tip polygonal distributions to create and object used for the different 3D plotting functions.

**Usage**

\[
dataNforNSdHresL treeL polygons)\]

**Arguments**

- **res**: the output file from a *rase* run.
- **tree**: phylogenetic tree of class "phylo" used as input for *rase*.
- **polygons**: list of polygons in *owin.object* format used as input for *rase*.

**Value**

Returns an object required by the plotting functions *phylo.3d*, *add.dens* and *add.polygons*. It is effectively a list with the following elements: $xyz$ containing the 3D coordinates, $edge$ containing the tree$edge$ and $pol$ containing the polygons.

**Author(s)**

Petr Keil, Ignacio Quintero

**References**


**See Also**

*phylo.3d*, *add.dens*, *add.polygons*, *rase*
Examples

```r
# load the mcmc results from rase run, polygons and tree
data(rase_data, package = 'rase')

pnames <- c('dextralis', 'viridis', 'leucoptera', 'interjecta',
'obscura', 'crepitans', 'ochroptera', 'napensis')
psophia_poly <- name.poly(psophia_poly, psophia_tree, poly.names = pnames)

# transform the data for 3d plotting
df3 <- data.for.3d(mcmc, psophia_tree, psophia_poly)

## Not run:
## plot the tree in 3D
phylo.3d(df3, z.scale = 10, pts = TRUE)

## add the polygons representing the tip distributions
add.polygons(df3)

## add the posterior density at each node of the 3d tree
if(requireNamespace("rpanel") && requireNamespace("tcltk"))
  add.dens(df3, mcmc, z.scale = 10, col = c(2:8))
## End(Not run)
```

name.poly

### Name & Order Polygons According to the Phylogeny

**Description**

Names polygons and orders them according to the phylogenetic tree. If they match the tree perfectly, the following message should appear: "tip labels and polygon names match and are in the same order".

**Usage**

```r
name.poly(polygons, tree, poly.names = NA)
```

**Arguments**

- `polygons` : list of polygons in `owin.object` format.
- `tree` : phylogenetic tree of class "phylo".
- `poly.names` : optional. Vector of names for the polygons. They should be in the same order as the polygons. If `poly.names = NA`, the function assumes the polygons are already named and only checks if they match the tip labels of the phylogenetic tree.
Value

Returns the polygon list with the names of the tip values in order.

Author(s)

Ignacio Quintero

See Also

shape.to.rase

Examples

```r
# attach polygon data
data(rase_data, package = 'rase')

## Not run:
# polygons already in format used by rase.
psophia_poly

# check the phylogenetic tree
psophia_tree

# Species names of polygons (in order)
pnames <- c('dextralis', 'viridis', 'leucoptera', 'interjecta',
            'obscura', 'crepitans', 'ochroptera', 'napensis')

# check tree's' tip labels
psophia_tree$tip.label

# name the polygons
polygons <- name.poly(psophia_poly, psophia_tree, poly.names = pnames)

## End(Not run)
```

phylo.3d

Plots a Phylogenetic Tree in 3D

Description

Plots a phylogenetic tree in 3D using rgl. It is mapped in 3D space, where the x/y axes are longitude/latitude, and the z axis is time. It is used as the first plot function to visualize the results of rase with add.dens and add.polygons (see example).

Usage

```r
phylo.3d(df3, z.scale = 1, pts = TRUE, ...)
```
phylo.3d

Arguments

- **df3**: an object from `data.for.3d`.
- **z.scale**: the scale of the time axis (vertical axis). Effectively, the time variable is multiplied by this number. It is very useful for visualization purposes. It should correspond with the `z.scale` argument in `add.dens`.
- **pts**: if TRUE, points in the vertices are plotted.
  ... further arguments to be passed to `lines3d`.

Author(s)

Petar Keil

References


See Also

`data.for.3d`, `add.dens`, `add.polygons`

Examples

```r
# load the mcmc results from rase run, polygons and tree
data(rase_data, package = 'rase')

pnames <- c('dextralis', 'viridis', 'leucoptera', 'interjecta',
            'obscura', 'crepitans', 'ochroptera', 'napensis')
psophia_poly <- name.poly(psophia_poly, psophia_tree, poly.names = pnames)

# transform the data for 3d plotting
df3 <- data.for.3d(mcmc, psophia_tree, psophia_poly)

## Not run:
# plot the tree in 3D
phylo.3d(df3, z.scale = 10, pts = TRUE)

# add the polygons representing the tip distributions
add.polygons(df3)

# add the posterior density at each node of the 3d tree
if(requireNamespace("rpanel") & requireNamespace("tcltk"))
  add.dens(df3, mcmc, z.scale = 10, col = c(2:8))

## End(Not run)
```
Description

Estimates Most Recent Common Ancestor (MRCA) states and the Brownian rate according to a Brownian Motion model of trait evolution using Maximum Likelihood. It can be used for several dimensions simultaneously. For 1-dimensional traits, we recommend the more stable `ace` function.

Usage

```r
point.like_bm(tree, values, start_values = NA, dimen = NA)
```

Arguments

- `tree` phylogenetic tree of class "phylo".
- `values` a list, with each element being one-dimensional values for one trait.
- `start_values` Optional. A vector of starting values for the Maximum Likelihood optimization. The function only estimates the MRCA and the rates for each trait, and that should be the order of the input starting values.
- `dimen` Optional. Number of dimensions (number of traits being analyzed). If `dimen = NA`, dimensions are obtained from `values`.

Value

Returns a list with the following components:

- `mrcas` Most recent common ancestor estimates for each trait (in the same order of the input values).
- `rates` Brownian motion rate parameters for each trait (in the same order of the input values).
- `nlm.details` A list with the results from nlm optimization. For details see `nlm`.

Author(s)

Ignacio Quintero

References


See Also

`ace`, ML taking into account ranges `ranges.like_bm`. 
Examples

```r
# Number of taxa to simulate tree
# Number of taxa
ntaxa <- 10

# Known parameters
mean_x <- 0
mean_y <- 0
sigma_x <- 1
sigma_y <- 1

# Create a random tree
tree <- ape::rtree(n = ntaxa)

# Create random data according to tree structure
x_locs <- as.numeric(mvtnorm::rmvnorm(1, rep(mean_x, ntaxa), sigma=sigma_x*vcv(tree)))
y_locs <- as.numeric(mvtnorm::rmvnorm(1, rep(mean_y, ntaxa), sigma=sigma_y*vcv(tree)))
values <- list(x_locs, y_locs)

## Not run:
# run point.like.bm
bm_results = point.like.bm(tree, values)
## End(Not run)
```

---

**Post-MCMC Handling**

**Description**

Takes the result from an mcmc run (e.g., output from `rase`) and applies a burnin and a thinning. Optionally it can be transformed into a **ggs** object that the ggmcmc package can use (for mcmc visualization).

**Usage**

```r
post.mcmc(res, burnin = 1000, thin = 10, as.ggmcmc = TRUE)
```

**Arguments**

- `res` The output from any mcmc run (e.g., output from `rase`). It is effectively a matrix where each column is a different parameter and each row is one iteration of the mcmc sampler.
- `burnin` Number of iterations to disregard as burnin.
- `thin` Sample every ith iteration. If no iterations are to be discarded, it should be `thin = 1`. Note that if `thin = 0`, then no iteration is saved at all.
- `as.ggmcmc` If TRUE (default), the object will be transformed into a **ggs** object.
Value

If `as.ggmcmc = TRUE` (the default), the function returns a `ggs` object. This can be used as input to the `ggs_` functions in the `ggmcmc` package. If `as.ggmcmc = FALSE`, the function returns a matrix where each column is a different parameter and each row is one iteration of the mcmc sampler.

Author(s)

Ignacio Quintero

See Also

`rase`, `bm_ase`, `ggs`

Examples

```r
# attach the data
data(rase_data, package = 'rase')

## Not run:
# Explore the data
str(mcmc)

# discard the first 200 iterations as burnin
# and keep every 10th iteration.
gg_dat <- post.mcmc(mcmc, burnin=200, thin = 10, as.ggmcmc = TRUE)

# density using ggmcmc for dispersal rates (sigma).
require(ggmcmc)
ggs_density(gg_dat, family = 'sigma')

## End(Not run)
```

ranges.like.bm  

*n-dimensional Maximum Likelihood of Range Brownian Motion*

Description

Estimates Most Recent Common Ancestor (MRCA) states and the Brownian rate according to a Brownian Motion model of trait evolution using Maximum Likelihood. The input for each trait is a one-dimensional range for every tip of the phylogenetic tree (e.g., if your trait is weight, instead of a 1 value input for each tip, usually the average, you can input the whole range of weights for that tip). It can be used for several dimensions simultaneously.

Usage

`ranges.like.bm(tree, lower_bounds, upper_bounds, start_values = NA, dimen = NA)`
Arguments

tree phylogenetic tree of class "phylo".
lower_bounds a list, with each element being one-dimensional lower bounds for a trait.
upper_bounds a list, with each element being one-dimensional upper bounds for a trait.
start_values Optional. A vector of starting values for the Maximum Likelihood optimization. The function only estimates the MRCA and the rates for each trait, and that should be the order of the input starting values.
dimen Optional. Number of dimensions (number of traits being analyzed). If dimen = NA, dimensions are obtained from values.

Value

Returns a list with the following components:
mrcas Most recent common ancestor estimates for each trait
rates Brownian motion rate parameters for each trait.

Author(s)

Ignacio Quintero

References


See Also

Contrast with point.like.bm

Examples

# Number of taxa to simulate tree
ntaxa <- 10

# Known parameters
# for three dimension
mean_x <- 0
mean_y <- 0
mean_z <- 0

sigma2x <- 1
sigma2y <- 2
sigma2z <- 3

# Create a random tree
tree <- ape::rtree(n = ntaxa)
# Create random data according to tree structure
x_locs <- as.numeric(mvtnorm::rmvnorm(1, rep(mean_x,ntaxa), sigma=sigma2x*vcv(tree)))
y_locs <- as.numeric(mvtnorm::rmvnorm(1, rep(mean_y,ntaxa), sigma=sigma2y*vcv(tree)))
z_locs <- as.numeric(mvtnorm::rmvnorm(1, rep(mean_z,ntaxa), sigma=sigma2z*vcv(tree)))

# random shifts to the mean to create ranges
xshift <- rexp(ntaxa)
yshift <- rexp(ntaxa)
zshift <- rexp(ntaxa)
x_lower <- x_locs - xshift
x_upper <- x_locs + xshift
y_lower <- y_locs - yshift
y_upper <- y_locs + yshift
z_lower <- z_locs - zshift
z_upper <- z_locs + zshift

# store rectangles: this is the trait data:
lower_bounds <- list(xl=x_lower, yl=y_lower,zl=z_lower)
upper_bounds <- list(xu=x_upper, yu=y_upper,zu=z_upper)

## Not run:
#run range.like_bm
range_results <- ranges.like_bm(tree, lower_bounds, upper_bounds)
## End(Not run)

rase

## Bayesian Range Ancestral State Estimation of Polygons

### Description
Performs a bayesian two-dimensional ancestral state estimation with polygonal distributions as input (contrast with \texttt{bm_ase}) according to a Brownian Motion model of trait evolution (or dispersal for phylogeography). It uses Gibbs sampling to approximate the posterior distribution. See reference at the end for more detailed information.

### Usage
\[
\text{rase(tree, polygons, niter = 1000, logevery = 10,}
\sigma_2 \text{scale} = 0.05, \text{screenlog = TRUE, params0 = NA, nGQ = 20)}
\]

### Arguments
\begin{itemize}
\item \texttt{tree} \hspace{1cm} phylogenetic tree of class \texttt{"phylo"}.
\item \texttt{polygons} list of polygons in \texttt{owin.object} format.
\item \texttt{niter} number of MCMC iterations. By default \texttt{niter = 1000}.
\item \texttt{logevery} iteration cycle to print current iteration. By default \texttt{logevery = 10}.
\item \texttt{sigma2_scale} optional. window proposal for \texttt{sigma2x} & \texttt{sigma2y}.
\item \texttt{screenlog} if \texttt{TRUE} (default), prints current iteration every \texttt{logevery} to the screen.
\end{itemize}
params0
optional. A vector of initial parameter values in the following order: x ancestors, y ancestors, sigma2x and sigma2y. If params0 = NA (default), an initial Maximum Likelihood optimization with polygon centroids using ace provides the starting parameter values.

nGQ
degree of the one-dimensional Gauss-Legendre quadrature rule (default = 20) as given by polyCub. SV in package polyCub. Bigger numbers make the integration more precise, but takes longer.

Value
returns a matrix where every column represents one parameter. The first columns (i.e., nx_x; where X = node 1, ..., node i) give the ancestral locations for trait x in the order of nodes in the tree (see the phylo class for details), followed by the ancestral locations of trait y (i.e., nx_y), and the rate parameter in x (sigma2x) and y (sigma2y).

Author(s)
Forrest Crawford, Ignacio Quintero

References

See Also
Contrast with the point ancestral state estimation bm_ase. For the maximum likelihood version of ranges see ranges.like.bm. For post-mcmc handling see post.mcmc.

Examples
# Here the application in the paper of Quintero et al.,
# on the Psophia trumpeters
# is shown using rase package.

#load data
data(rase_data, package = 'rase')

## Not run:
# check the data we are going to use
# the phylogenetic tree
psophia_tree

# the GPC polygons of Psophia distribution.
psophia_poly

# Species names of polygons (in order)
pnames <- c('dextralis', 'viridis', 'leucoptera', 'interjecta',
           'obscura', 'crepitans', 'ochroptera', 'napensis')

# name the polygons
rasp.hpoly <- name.poly(ras.hpoly, ras.hpoly, ras.hpoly, poly.names = pnames)

# Run ras for 10 iterations
rasr.results <- rasr(ras.hpoly, ras.hpoly, niter = 100)
# Run with higher number of iterations
# rase_results <- rase(ras.hpoly, polygons)

# Check the results
str(rase_results)

# post-MCMC handling
rase_results_for_ggmcmc <- post.mcmc(rase_results, burnin=0, thin = 1)

#plot the densities for dispersal rates using ggmcmc
require(ggmcmc)
ggs_traceplot(rase_results_for_ggmcmc, family = 'sigma')

### End(Not run)

### rasr.slice

---

#### Ancestral State Estimation At Any Given Time Slice

**Description**

Estimates the geographical location of ancestors (at branches or, much less likely, at nodes) at any given point in time integrating over a rasr. It first uses tree.slice to identify the branches that the slice intersects with and then uses MCMC sampling to approximate the posterior distribution of the ancestor locations.

**Usage**

rasr.slice(tree, slice, res, polygons,
params0 = NA, niter = 1000, logevery = 10, nGQ = 20)

**Arguments**

- **tree**
  - phylogenetic tree of class "phylo". The same used in the coderasr run.
- **slice**
  - the time at which to slice. It should be in the same units of the phylogenetic tree.
- **res**
  - output from rasr. Ideally, it should be the post-handled mcmc (i.e., already applied burnin and thinning, if applicable; see post.mcmc).
- **polygons**
  - list of polygons in owin.object format used as input for rasr.
- **params0**
  - optional. A vector of initial parameter values in the following order: x ancestors for each branch in the order given by tree.slice, followed by y ancestors in the same order.
- **niter**
  - number of MCMC iterations. By default niter = 1000.
- **logevery**
  - iteration cycle to print current iteration. By default logevery = 10.
rase.slice

degree of the one-dimensional Gauss-Legendre quadrature rule (default = 20) as given by `polyCub.SV` in package polyCub. Bigger numbers make the integration more precise, but takes longer.

**Value**

returns a matrix where every column represents one parameter. The first columns (i.e., bX.x; where X = branch 1, ..., branch i) give the ancestral locations for trait x in the order given by `tree.slice`, followed by the ancestral locations of trait y (i.e., bX.y).

**Author(s)**

Ignacio Quintero Forrest Crawford

**References**


**See Also**

tree.slice, rase

**Examples**

```r
# load data
data(rase_data, package = 'rase')

## Not run:
# check the data we are going to use

# examine the mcmc result from rase
# after it has been applied a burnin
# phase and a thinning
str(mcmc)

# the phylogenetic tree used in the rase run
psophia_tree

# the polygons used in the rase run
str(psophia_poly)

# define the slice of time;
# for now, let's say 1 Million
# years ago (Ma)
slice <- 1

# Species names of polygons (in order)
pnames <- c('dextralis', 'viridis', 'leucoptera', 'interjecta',
           'obscura', 'crepitans', 'ochroptera', 'napensis')

# name the polygons
```
### rase_data

Psophia Data Used for rase Package

#### Description


#### Usage

```r
rase_data
```

#### Format

- `psophia_shp` contains the distributions of Psophia species as shapefiles in the format `SpatialPolygonsDataFrame`.
- `psophia_poly` contains the distributions of Psophia species as a list of polygons in `owin.object` format used as input for `rase`. `psophia_tree` contains Psophia's phylogenetic tree in `phylo` format. `mcmc` contains the MCMC chain of rase run over the Psophia data after a 2000 burnin period and sampling every 10 iterations.

#### Source


shape.to.rase

Transform a Shapefile into owin.object Format

Description
Transforms a Shapefile in format SpatialPolygonsDataFrame to the format owin.object used by rase.

Usage
shape.to.rase(shape_poly)

Arguments
shape_poly an object of class SpatialPolygonsDataFrame.

Value
Returns a list of polygons in owin.object format.

Author(s)
Ignacio Quintero

See Also
readShapePoly

Examples
## Not run:
# the Shapefiles for the distributions of the tips of the tree
data(rase_data, package = 'rase')

psophia_shp

# Transform the shapefile into the format used by rase
polygons = shape.to.rase(psophia_shp)

# check the structure of the polygons
polygons

## End(Not run)
tree.slice

Identify Tree Branches That Intersect With a Given Time Slice.

Description
Identifies the branches in a phylogenetic tree that the slice intersects with. It is used by rase.slice.

Usage
tree.slice(tree, slice)

Arguments
- tree: phylogenetic tree of class "phylo".
- slice: the time at which to slice. It should be in the same units of the phylogenetic tree.

Details
The tree must be fully dichotomous and all the tips should be extant lineages (time = 0).

Value
returns a matrix where each row is an intersecting branch with the following columns: two integers associating the branch (i.e., ancestor, descendant), the age of the ancestral node, and the age of the descendant node.

Author(s)
Ignacio Quintero

References

See Also
rase.slice

Examples
```r
#load data
data(rase_data, package = 'rase')

## Not run:

# a phylogenetic tree used in the rase run
psophia_tree
```
# define the slice of time;
# say, 1 Million years ago (Ma)
slice <- 1

# slice the tree
slice_results <- tree.slice(psophia_tree, slice = slice)

# check results
slice_results

## End(Not run)
Index

*Topic datasets
  rase_data, 20

*Topic package
  rase-package, 2

ace, 7, 12, 17
add.dens, 3, 5, 8, 10, 11
add.polygons, 4, 5, 8, 10, 11

bm_ase, 6, 14, 16, 17
data.for.3d, 4, 5, 8, 11
ggs, 13, 14
lines3d, 11

mcmc(rase_data), 20

name.poly, 9
nlm, 12, 15

owin.object, 8, 9, 16, 18, 20, 21

phylo, 20
phylo.3d, 3–5, 8, 10
point.like.bm, 7, 12, 15
polyCub.SV, 17, 19
polygon3d, 4, 5
post.mcmc, 7, 13, 17, 18
psophia_poly(rase_data), 20
psophia_shp(rase_data), 20
psophia_tree(rase_data), 20

ranges.like.bm, 12, 14, 17
rase, 6–8, 13, 14, 16, 18–21
rase-package, 2
rase.slice, 18, 22
rase_data, 20
readShapePoly, 21
shape.to.rase, 10, 21

SpatialPolygonsDataFrame, 20, 21
tree.slice, 18, 19, 22