Package: MorphSim (via r-universe)

June 5, 2026

Type Package

Title Simulate Discrete Character Data along Phylogenetic Trees

Version 1.2.0

Description Tools to simulate morphological traits along phylogenetic trees with branch lengths representing evolutionary distance or time. Includes functions for visualizing evolutionary processes along trees and within morphological character matrices.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Imports ape, FossilSim, stats, graphics, phangorn

Suggests TreeSim, testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Depends R (>= 3.5)

LazyData true

NeedsCompilation no

Author Laura Mulvey [aut, cre, cph], Tim Brandler [aut, cph], Alessio Capobianco [aut, cph], Rachel Warnock [aut, cph], Joelle Barido-Sottani [aut, cph]

Maintainer Laura Mulvey <`laura.mulvey479@gmail.com`>

Config/pak/sysreqs cmake libglpk-dev make libicu-dev libuv1-dev libxml2-dev

Repository https://laumul.r-universe.dev

Date/Publication 2026-05-06 16:57:20 UTC

RemoteUrl https://github.com/cran/MorphSim

RemoteRef HEAD

RemoteSha b0a9a99dac78d0ad1f07c3e855301a6adb0f1549

Contents

`combine.morpho` *Combine Two Morpho Objects*

Description

This function merges two ‘morpho‘ objects, combining their sequences, model parameters, and transition histories, while ensuring tree and fossil consistency.

Usage

```
combine.morpho(x, y)
```

Arguments

`x`	A ‘morpho‘ object.
`y`	A ‘morpho‘ object.

Value

A combined ‘morpho‘ object.

Examples

```
phy <- ape::rtree(10)

# simulate characters along the branches of the tree
morpho1 <-  sim.morpho(tree = phy,
                           k = c(2,3,4),
                           trait.num = 20,
                           ancestral = TRUE,
                           partition = c(10,5,5),
                           ACRV = "gamma",
                           variable = TRUE,
                           ACRV.ncats = 4,
                           define.Q = NULL)

morpho2 <-  sim.morpho(tree = phy,
                           k = c(2,3,4),
                           trait.num = 20,
                           ancestral = TRUE,
                           partition = c(10,5,5),
                           ACRV = "gamma",
                           variable = TRUE,
                           ACRV.ncats = 4,
                           define.Q = NULL)

combined <- combine.morpho(morpho1, morpho2)
```

`convergent_evol` *Determines the number of convergently evolved traits*

Description

Identifies which traits have evolved independently multiple times (convergent evolution) in a morpho object.

Usage

```
convergent_evol(data = NULL)
```

Arguments

`data`	A morpho object

Value

A data.frame listing convergent traits, their state, and number of transitions

Examples

```
phy <- ape::rtree(10)
morpho_data <- sim.morpho(tree = phy, k = 2, trait.num = 20)
convergent_evol(morpho_data)
```

`find_path_to_tip` *Determines the route (nodes and branches) for a tip in a phylogenetic tree*

Description

Traverses the tree to determine the evolutionary path (branches) from root to a given tip.

Usage

```
find_path_to_tip(tree, tip)
```

Arguments

`tree`	A phylogenetic tree of class `phylo`
`tip`	Tip label (character)

Value

A matrix with columns `parent` and `child` representing the path

`get.convergent` *Identify convergent traits*

Description

Identifies convergent evolution in a morpho object. When called without a trait, returns a summary of which traits are convergent. When called with a specific trait, returns tip groupings by state and evolutionary origin.

Usage

```
get.convergent(data, trait = NULL)
```

Arguments

`data`	A morpho object
`trait`	The trait number to examine. If `NULL` (default), returns a summary.

Value

A data frame. Without `trait`: trait index and number of convergent states. With `trait`: state, origin, tips, and whether the state is convergent.

Examples

```
phy <- ape::rtree(10)
morpho_data <- sim.morpho(tree = phy, k = 2, trait.num = 20)
get.convergent(morpho_data)
get.convergent(morpho_data, trait = 3)
```

`get.matrix` *Get Specific Morphological Matrix*

Description

Return data frame of morphological matrix

Usage

```
get.matrix(data, seq)
```

Arguments

`data`	A morpho object
`seq`	The sequence data you want to extract

Value

A data frame with columns edge (branch number), state, and hmin (where along the branch the transition occurred)

Examples

```
phy <- ape::rtree(10)
morpho_data <- sim.morpho(tree = phy, k = 2, trait.num = 20)
get.matrix(morpho_data, seq = "tips")
```

`get.reconstructed` *Add reconstructed tree and matrix to morpho object*

Description

This function should be called if you want to access the reconstructed tree and character matrix directly from the morpho object. It adds the reconstructed tree in Newick format ($trees$Recon), containing only the sampled lineages, and a character matrix ($sequences$recon) containing only the taxa present in that reconstructed tree.

Usage

```
get.reconstructed(data)
```

Arguments

`data`	‘morpho object‘ containing a fossil object

Value

a ‘morpho object‘

Examples

```
# simulate tree
lambda = 0.1
mu = 0.05
tips = 10
t = TreeSim::sim.bd.taxa(n = tips, numbsim = 1, lambda = lambda, mu = mu)[[1]]

# Simulate fossils and extant taxa
rate = 0.1 # poisson sampling rate
f = FossilSim::sim.fossils.poisson(rate = rate, tree = t, root.edge = FALSE)
rho = 0.5
f2 = FossilSim::sim.extant.samples(fossils = f, tree = t, rho = rho)
morpho_data <-  sim.morpho(k = c(2,3),
                           time.tree = t,
                           trait.num = 6,
                           ancestral = TRUE,
                           br.rates = 0.1,
                           partition = c(4,2),
                           ACRV = "gamma",
                           variable = TRUE,
                           ACRV.ncats = 4,
                           fossil = f2)

re <- get.reconstructed(morpho_data)
```

`get.transitions` *Get Tranisition History*

Description

Returns the full transition history for a given trait, including the root state.

Usage

```
get.transitions(data, trait)
```

Arguments

`data`	A morpho object
`trait`	The trait number .

Value

A data frame with columns edge (branch number), state, and hmin (where along the branch the transition occurred)

Examples

```
phy <- ape::rtree(10)
morpho_data <- sim.morpho(tree = phy, k = 2, trait.num = 20)
get.transitions(morpho_data, trait = 2)
```

`get_gamma_rates` *Get discrete gamma rates*

Description

Computes a set of discrete gamma rates for rate variation across sites or characters. This function is adapted from the `phangorn` package.

Usage

```
get_gamma_rates(alpha, k)
```

Arguments

`alpha`	Numeric. The shape parameter of the gamma distribution.
`k`	Integer. The number of rate categories.

Value

Numeric vector of length k representing the discrete gamma rates.

Examples

```
get_gamma_rates(alpha = 0.5, k = 4)
```

`get_lognormal_rates` *Get discrete log-normal rates*

Description

Computes a set of discrete log-normal rates for rate variation across sites or characters. The rates are normalized so that the mean rate equals 1.

Usage

```
get_lognormal_rates(meanlog, sdlog, k)
```

Arguments

`meanlog`	Numeric. Mean on the log scale.
`sdlog`	Numeric. Standard deviation on the log scale.
`k`	Integer. Number of rate categories.

Value

Numeric vector of length k representing the discrete log-normal rates.

Examples

```
get_lognormal_rates(meanlog = 0, sdlog = 1, k = 4)
```

`morpho` *Morpho object*

Description

Create a morpho object.

Usage

```
morpho(
  sequences = NULL,
  trees = NULL,
  model = NULL,
  transition_history = NULL,
  root.states = NULL,
  fossil = NULL
)

as.morpho(
  sequences,
  trees,
  model = NULL,
  transition_history = NULL,
  root.states = NULL,
  fossil = NULL
)

is.morpho(sequences)
```

Arguments

`sequences`	A list containing all of the sequences simulated. This can contain sequences for taxa at the tips or the tree, along the nodes, and if present, for sampled ancestors (SA)
`trees`	A list containing the trees and branch lengths used for the simulation. EvolTree contains a phylogenetic tree with branch lengths representing evolutionary distance. TimeTree (if present) contains the same tree with branch lengths in unit of time. BrRates can either be a single value, when simulating under a strict clock, or a vector of values representing the rate/branch
`model`	A list containing all model attributes. Model specifies the components specified to simulate under. RateVar contains the relative values drawn from the specified distribution. RateVarTrait species the rate used to simulate each trait
`transition_history`	The constant character transitions along the branches
`root.states`	A vector supplying the root state for each character
`fossil`	Fossil object used to simulate data

Value

An object of class "`morpho`" containing the simulated morphological data and associated information. The object includes the simulated sequences, phylogenetic trees and branch rates used for the simulation, model parameters, root states, fossil information (if provided), and the character transition history.

Examples

```
phy <- ape::rtree(10)
morpho_data <- sim.morpho(tree = phy, k = 2, trait.num = 5)
is.morpho(morpho_data) # TRUE
```

`morpho_data` *Example morpho dataset*

Description

Small example dataset for testing morpho functions.

Usage

```
data(morpho_data)
```

Format

A list with components:

sequences List of sequences for each taxon
trees List containing the evolutionary tree

`morphsim_fossilsim` *Match sampled ancestor labels*

Description

Match the sampled ancestor labels from `Morphsim` and `Fossilsim`

Usage

```
morphsim_fossilsim(data = NULL)
```

Arguments

`data`	Morpho object containing fossils

Value

A character matrix mapping sampled ancestor labels between the naming conventions used by `Morphsim` and `Fossilsim`

Examples

```
data(morpho_data)
morphsim_fossilsim <- function(data = morpho_data)
```

`plot.morpho` *Plot full evolutionary history*

Description

This function creates a plot showing continuous evolution of discrete traits.

Usage

```
## S3 method for class 'morpho'
plot(
  x = NULL,
  trait = 1,
  timetree = TRUE,
  show.fossil = TRUE,
  reconstructed = TRUE,
  show.convergent = TRUE,
  root.edge = FALSE,
  edge.width = 1,
  label.offset = 0.01,
  e.cex = 0.5,
  f.cex = 1,
  box.cex = 4,
 col = c("#fdfdfd", "lightgray", "lightblue", "pink", "yellow", "green", "orange"),
  col.timescale = "darkgrey",
  ...
)
```

Arguments

`x`	A morpho object
`trait`	The trait number to plot.
`timetree`	TRUE or FALSE. Indicate whether you want to plot a time tree or not. Default = FALSE (uses distance tree if FALSE).
`show.fossil`	Plot the fossil along the tree. Default = FALSE.
`reconstructed`	Plot the reconstructed tree. Default = FALSE.
`show.convergent`	When convergent = TRUE, transition boxes are highlighted for any state that arose independently more than once during the simulation. This includes transitions that were subsequently reversed on descendant branches
`root.edge`	If TRUE plot the root edge. Default = FALSE.
`edge.width`	Width of the branches.
`label.offset`	Distance of tip label to tree tips.
`e.cex`	Size of extant taxa.

`f.cex`	Size of fossils.
`box.cex`	Size of traits on plot
`col`	A vector of colors that should be the same length or longer than the number of different character states (k). If not specified, the traits from 0 to 6 can be differentiated.
`col.timescale`	A single color for the timescale. Default = "darkgrey".
`...`	Other arguments to be passed to methods, such as graphical parameters.

Value

No return value, called for its side effect of producing a plot.

Examples

```
# simulate a phylogenetic tree
data(morpho_data)
plot(morpho_data, trait = 4, timetree = FALSE, show.fossil = FALSE,
     root.edge = FALSE, reconstructed = FALSE)
```

`plotMorphoGrid` *Plots morphological matrix*

Description

This function plots the full morphological matrix assocaited with the character data at the tips of a tree. Requires a morpho object as input.

Usage

```
plotMorphoGrid(
  data = NULL,
  timetree = FALSE,
  seq = "tips",
  num.trait = "all",
 col = c("lavender", "white", "lightskyblue1", "pink", "gold2", "forestgreen", "coral")
)
```

Arguments

`data`	A morpho object
`timetree`	TRUE or FALSE Indicate whether you want to plot a time tree or not. default FALSE, uses distance tree if FALSE
`seq`	the sequence data to plot: "tips", "nodes", "SA", or "recon"
`num.trait`	default is set to "all" which plots all traits in black font. If you want to focus on a specific trait set it here, e.g. num.trait = 1 and this trait will be highlighted

`col`	A vector of colors that should be the same length or longer than the number of different character states (k). if not specified, the traits from 0 to 6 can be differentiated

Value

No return value, called for its side effect of producing a plot.

Examples

```
data(morpho_data)
# plot the character matrix
plotMorphoGrid(data = morpho_data, seq = "tips", num.trait = "all")
```

`reconstruct.matrix` *Get reconstructed matrix*

Description

This function returns the moprhological matrix for tips in the reconstructed tree.

Usage

```
reconstruct.matrix(data)
```

Arguments

`data`	A ‘morpho‘ object with fossil data

`reconstruct.tree` *Color branches for plotting a reconstructed tree*

Description

This function generates colors for branches when plotting a reconstructed tree from a morpho object containing fossil data. Branches that are part of the reconstructed tree or have fossils along them are colored black; all others are grey.

Usage

```
reconstruct.tree(data)
```

Arguments

`data`	A morpho object which contains fossil data and a time-calibrated tree.

Value

A list of length 2:

`b.colours`	Vector of branch colors for plotting.
`rem`	Indices of branches with fossils.

`sim.missing.data` *Remove morphological character data*

Description

This function removes characters from a morphological matrix simulated using morphsim

Usage

```
sim.missing.data(
  data = NULL,
  seq = NULL,
  method = NULL,
  probability = NULL,
  traits = NULL,
  taxa = NULL
)
```

Arguments

`data`	A ‘morpho‘ object with sequence data.
`seq`	Character. Which sequence data to use: "tips", "nodes", or "SA".
`method`	Character. Method for removing data. Options: • `"random"`: removes characters randomly across the entire matrix. • `"partition"`: removes characters by partition (one probability per partition). • `"rate"`: removes characters by rate category (one probability per rate category, ordered from slowest to fastest). • `"trait"`: removes characters from specific traits across all taxa. • `"taxa"`: removes characters from specific taxa across all traits. • `"extinct"`: removes data from extinct taxa only. Requires `seq = "tips"` as extinct taxa are only present at the tips of the tree. • `"trait_taxa"`: removes characters at the intersection of specific traits and specific taxa, reflecting scenarios where certain anatomical regions are less likely to be preserved in particular lineages.
`probability`	Numeric. Probability of missing data (single value or vector depending on method).
`traits`	Integer vector. Indices of traits to remove. Required when `method = "trait"` or `method = "trait_taxa"`.
`taxa`	Character vector. Names of taxa to remove data from. Required when `method = "taxa"` or `method = "trait_taxa"`.

Value

An object of class morpho.

Examples

```
# simulate a phylogenetic tree
phy <- ape::rtree(10)

# simulate characters along the branches of the tree
morpho_data <-  sim.morpho(tree = phy,
                           k = c(2,3,4),
                           trait.num = 20,
                           ancestral = TRUE,
                           partition = c(10,5,5),
                           ACRV = "gamma",
                           variable = TRUE,
                           ACRV.ncats = 4,
                           define.Q = NULL)

# randomly remove data
missing.data <- sim.missing.data(data = morpho_data,
                                  method = "random",
                                  seq = "tips",
                                  probability = 0.5)


# remove data based on the partition
missing.data <- sim.missing.data(data = morpho_data,
                                 method = "partition",
                                 seq = "tips",
                                 probability = c(0.7, 0, 0.5))

# remove data based on the rate it was simulated under
missing.data <- sim.missing.data(data = morpho_data,
                                        method = "rate",
                                        seq = "tips",
                                        probability = c(0,0,0.2,1))

# remove characters from specific traits
missing.data <- sim.missing.data(data = morpho_data,
                                 method = "trait",
                                 seq = "tips",
                                 probability = 1,
                                 traits = c(1,2,5))

# remove characters from specific taxa
missing.data <- sim.missing.data(data = morpho_data,
                                 method = "taxa",
                                 seq = "tips",
                                 probability = 1,
                                 taxa = c("t1", "t2"))
```

```
# remove specific character traits from specific taxa
missing.data <- sim.missing.data(data = morpho_data,
                                 method = "trait_taxa",
                                 seq = "tips",
                                 probability = 0.8,
                                 traits = c(1,2,5),
                                 taxa = c("t1", "t2"))
```

`sim.morpho` *Simulate characters along branches in a tree*

Description

This function simulates discrete character data along the branches of a phylogenetic tree using a continuous-time Markov chain. It implements the Mk model and its extensions (Lewis 2001), including among-character rate variation (ACRV) using discretized gamma distributions (Yang 1994) or lognormal distributions (Wagner 2012), following the implementation described in Capobianco and Hohna (2025). It can be used with either a time tree or a tree with branch lengths in evolutionary distance. If using a time tree, branch rates can be specified either as a single value for all branches or as a vector with different rates per branch. If no branch rates are specified a default of 0.1 is applied to all branches.

Usage

```
sim.morpho(
  tree = NULL,
  time.tree = NULL,
  k = 2,
  trait.num,
  partition = NULL,
  br.rates = NULL,
  root.state = NULL,
  variable = FALSE,
  full.states = FALSE,
  parsimony = NULL,
  ACRV = NULL,
  alpha.gamma = 1,
  ACRV.ncats = 4,
  meanlog = NULL,
  sdlog = NULL,
  define.ACRV.rates = NULL,
  ancestral = TRUE,
  fossil = NULL,
  define.Q = NULL
)
```

Arguments

`tree`	A phylogenetic tree (class `"phylo"`) with branches representing genetic distance.
`time.tree`	A phylogenetic tree (class `"phylo"`) with branches representing time.
`k`	Integer (>= 2). Number of trait states. Can be a vector if using partitions.
`trait.num`	Integer (> 0). The total number of traits to simulate.
`partition`	Integer vector. Specifies the number of traits per partition. Must sum to `trait.num`.
`br.rates`	Numeric. Clock rates per branch. Can be a single value (strict clock) or a numeric vector of length equal to the number of branches in the tree, one rate per branch in the same order as `tree$edge`. Custom vectors of rates can be provided directly here, for example rates simulated using `simclock::relaxed.tree()`.
`root.state`	Integer. The root state at the start of the simulation. Must be within the range of states implied by `k`.
`variable`	Logical. If `TRUE`, simulate only varying characters (MkV model). Default is `FALSE`.
`full.states`	Logical. If `TRUE`, ensures that all character states specified by `k` are present in at least one tip for each trait. Default is `FALSE`.
`parsimony`	Character. If `"standard"`, retain only characters where at least two states each occur in two or more taxa; autapomorphic states are tolerated. If `"strict"`, every observed state must occur in two or more taxa, excluding characters with any autapomorphic states entirely. If `NULL`, no parsimony filter is applied. Default is `NULL`.
`ACRV`	Character. Among-character rate variation model. One of `"gamma"`, `"lgn"`, `"user"`, or `NULL`. `"gamma"` draws rates from a discretized gamma distribution (Yang 1994); requires `ACRV.ncats` and `alpha.gamma`. `"lgn"` draws rates from a discretized lognormal distribution (Wagner 2012); requires `ACRV.ncats`, `meanlog`, and `sdlog`. `"user"` uses user-supplied rates via `define.ACRV.rates`. `NULL` simulates all traits under the same rate. Default is `NULL`.
`alpha.gamma`	Numeric. Shape parameter α for the gamma distribution. Default is 1.
`ACRV.ncats`	Integer. Number of discrete rate categories for among-character rate variation.
`meanlog`	Numeric. Mean of the lognormal distribution on the log scale. Required when `ACRV = "lgn"`.
`sdlog`	Numeric. Standard deviation of the lognormal distribution on the log scale. Required when `ACRV = "lgn"`.
`define.ACRV.rates`	Numeric vector. User-supplied rate categories. Required when `ACRV = "user"`.
`ancestral`	Logical. If `TRUE`, return the states at all ancestral nodes. Default is `TRUE`.
`fossil`	A fossil object from `FossilSim` to simulate morphological characters for sampled ancestors.
`define.Q`	Numeric matrix. A custom transition rate matrix Q for simulation. Must be square and rows must sum to zero.

Value

An object of class ‘morpho‘, with the following components:

sequences A list containing up to 3 elements: morphological data for the ‘tips‘ of the tree, the ‘nodes‘, and, if provided, the sampled ancestors (‘SA‘). For ‘SA‘, the naming scheme differs from that of ‘FossilSim‘: the morphological data are named using the specimen number (‘data$fossil$specimen‘) and the branch number along which the fossil was sampled.

tree A list containing up to 3 elements: the ‘EvolTree‘ (branch lengths in genetic distance), the ‘TimeTree‘ (branch lengths in time units), and ‘BrRates‘ (the evolutionary rate per branch).

model Information about the model used to simulate the data. ‘Specified‘ states the exact model used per partition, as well as the number of traits and character states respectively. ‘RateVar‘ contains the relative rates used to simulate the data, and ‘RateVarTrait‘ contains information about which rate category was used to simulate each trait. These values are listed from lowest rate (1) to highest.

transition_history A list containing *n* data frames, where *n* is the number of simulated traits. Each data frame contains information about transitions that occurred for that trait, including the branch number (‘edge‘), the new state number (‘state‘), and the point along the branch where the transition occurred (‘hmin‘).

root.states A vector of root states for each trait.

fossil The fossil object provided to ‘morphsim‘ from ‘FossilSim‘. The naming scheme therefore matches that of ‘FossilSim‘.

References

Lewis, P.O. (2001) A likelihood approach to estimating phylogeny from discrete morphological character data. *Systematic Biology* 50:913-925.

Yang, Z. (1994) Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *Journal of Molecular Evolution* 39:306-314.

Wagner, P.J. (2012) Modelling rate distributions using character compatibility: implications for morphological evolution among fossil invertebrates. *Biology Letters* 8:143-146.

Capobianco, A. and Hohna, S. (2025) On the MkV model with among-character rate variation. *Systematic Biology*.

Examples

```
# simulated tree
phy <- ape::rtree(10)

# simulate characters along the branches of the tree
morpho_data <-  sim.morpho(tree = phy,
                           k = c(2,3,4),
                           trait.num = 20,
                           partition = c(10,5,5),
                           ACRV = "gamma",
                           ACRV.ncats = 4,
                           variable = TRUE,
                           ancestral = TRUE,
```

```
                          define.Q = NULL)

# To simulate ordered characters:
# First define a Q-matrix. The following is for ordered characters where transitions can only occur
# between states 0 and 1 and 1 and 2

ord_Q <- matrix(c(
-0.5, 0.5, 0.0,
0.3333333, -0.6666667, 0.3333333,
0.0, 0.5, -0.5
), nrow = 3, byrow = TRUE)

# This Q matrix can be then used to simulate character data.

morpho_data <- sim.morpho(tree = phy,
                          k = 3,
                          trait.num = 20,
                          ancestral = TRUE,
                          ACRV = "gamma",
                          variable = TRUE,
                          ACRV.ncats = 4,
                         define.Q = ord_Q)
```

`stats.morpho` *Calculates statistics for a morpho object*

Description

Computes three key pieces of information for a morpho object: 1. The Consistency Index (CI) and Retention Index (RI) based on the tip sequence data. 2. Convergent traits, identifying traits that have evolved independently multiple times. 3. Summary information about the size and structure of the tree.

Usage

```
stats.morpho(data)
```

Arguments

`data`	A morpho object

Value

A list with three elements: - `Statistics`: data.frame with CI and RI - `Convergent_Traits`: data.frame listing convergent traits - `Tree`: data.frame summarizing extant/extinct tips and sampled ancestors

See Also

`get.convergent` for detailed per-trait convergence information

Examples

```
data(morpho_data)

summary <- stats.morpho(data = morpho_data)
```

`symmetric.Q.matrix` *Generate a symmetric Q matrix*

Description

Creates a K x K symmetric rate matrix (Q matrix) with equal transition rates between states. The diagonal elements are set such that each row sums to zero.

Usage

```
symmetric.Q.matrix(K)
```

Arguments

`K`	Integer. The number of states.

Value

A K x K numeric matrix representing the symmetric Q matrix.

Examples

```
symmetric.Q.matrix(4)
```

`write.morpho` *Write morpho data to file*

Description

Export components of a morpho object to file. Can write trees, character matrices, or fossil ages in various formats.

Usage

```
write.morpho(data, file, type = "tree", reconstructed = FALSE, uncertainty = 0)
```

Arguments

`data`	A morpho object
`file`	File name
`type`	type to write: "tree", "matrix", or "ages"
`reconstructed`	If TRUE, write the reconstructed version. Default FALSE.
`uncertainty`	Numeric. Age uncertainty for fossil ages. Default 0.

Value

No return value, called for its side effect of writing data to a file.

Examples

```
data(morpho_data)
tmp <- tempfile(fileext = ".tre")
write.morpho(morpho_data, file = tmp, type = "tree")
```

`write.recon.matrix` *Write reconstructed character matrix to file*

Description

Write the character matrix for the reconstructed tree to a nexus file

Usage

```
write.recon.matrix(data, file = NULL)
```

Arguments

`data`	Morpho object
`file`	File name

Value

No return value, called for its side effect of writing data to a file.

Examples

```
data(morpho_data)
tmp <- tempfile(fileext = ".nex")
write.recon.matrix(data = morpho_data, file = tmp)
```

`write.recon.tree` *Write reconstructed tree to file*

Description

Write the reconstructed tree to Newick string

Usage

```
write.recon.tree(data = NULL, file = NULL)
```

Arguments

`data`	Morpho object
`file`	File name

Value

No return value, called for its side effect of writing data to a file.

Examples

```
data(morpho_data)
tmp <- tempfile(fileext = ".tre")
write.recon.tree(data = morpho_data, file = tmp)
```

`write.recon.tsv` *Write the taxa ages of reconstructed tree*

Description

Writes the ages of the specimen in the reconstructed tree to a file. The tsv format used here is directly compatible with RevBayes

Usage

```
write.recon.tsv(data, file, uncertainty = 0)
```

Arguments

`data`	Morpho object
`file`	File name
`uncertainty`	Numeric. Adds uncertainty to fossil ages in the morpho object. The ages in the object are point estimates by default; setting ‘uncertainty‘ will create an age range of ± this value (in millions of years).

Value

No return value, called for its side effect of writing data to a file.

Examples

```
data(morpho_data)
tmp <- tempfile(fileext = ".tsv")
write.recon.tsv(data = morpho_data, file = tmp)
```

`write.tsv` *Write the taxa ages*

Description

Writes the ages of the specimens in the true tree to a file. The tsv format used here is directly compatible with RevBayes

Usage

```
write.tsv(data, file, uncertainty = 0)
```

Arguments

`data`	Morpho object
`file`	File name
`uncertainty`	Numeric. Adds uncertainty to fossil ages in the morpho object. The ages in the object are point estimates by default; setting ‘uncertainty‘ will create an age range of ± this value (in millions of years).

Value

No return value, called for its side effect of writing data to a file.

Examples

```
data(morpho_data)
tmp <- tempfile(fileext = ".tsv")
write.tsv(data = morpho_data, file = tmp)
```

Index