Package ‘DAISIEprep’

January 20, 2025

Type Package

Title Extracts Phylogenetic Island Community Data from Phylogenetic Trees

Version 1.0.0

Maintainer Joshua W. Lambert `<j.w.l.lambert@rug.nl>`

Description Extracts colonisation and branching times of island species to be used for analysis in the R package 'DAISIE'. It uses phylogenetic and endemicity data to extract the separate island colonists and store them.

URL `https://github.com/joshwlambert/DAISIEprep`, `https://joshwlambert.github.io/DAISIEprep/`

BugReports `https://github.com/joshwlambert/DAISIEprep/issues`

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Depends R (>= 4.0)

Imports methods, ape, phylobase, ggplot2, scales, ggtree, DAISIE, castor, tibble, rlang

Suggests testthat (>= 3.0.0), knitr, rmarkdown, covr, diversitree, corHMM, tidyr, dplyr, ggimage

VignetteBuilder knitr

Config/testthat/edition 3

Collate 'DAISIEprep-package.R' 'add_asr_node_states.R' 'add_island_colonist.R' 'add_missing_species.R' 'add_multi_missing_species.R' 'add_outgroup.R' 'all_descendants_conspecific.R' 'any_back_colonisation.R' 'any_outgroup.R' 'any_polyphyly.R' 'as_daisie_datatable.R' 'benchmark.R' 'bind_colonist_to_tbl.R' 'check_phylo_data.R' 'count_missing_species.R' 'create_daisie_data.R'

'create_endemicity_status.R' 'create_test_phylod.R' 'data.R'
'default_params_doc.R' 'endemicity_to_sse_states.R'
'extract_asr_clade.R' 'extract_clade_name.R'
'extract_endemic_clade.R' 'extract_endemic_singleton.R'
'extract_island_species.R' 'extract_multi_tip_species.R'
'extract_nonendemic.R' 'extract_species_asr.R'
'extract_species_min.R' 'extract_stem_age.R'
'extract_stem_age_asr.R' 'extract_stem_age_genus.R'
'extract_stem_age_min.R' 'get_endemic_species.R'
'is_back_colonisation.R' 'is_duplicate_colonist.R'
'is_identical_island_tbl.R' 'is_multi_tip_species.R'
'island_colonist-class.R' 'island_colonist-accessors.R'
'island_tbl-class.R' 'island_tbl-accessors.R'
'island_tbl-methods.R' 'multi_extract_island_species.R'
'multi_island_tbl-class.R' 'multi_island_tbl-methods.R'
'plot_colonisation.R' 'plot_performance.R' 'plot_phylod.R'
'plot_sensitivity.R' 'print-methods.R' 'read_performance.R'
'read_sensitivity.R' 'rm_duplicate_island_species.R'
'rm_island_colonist.R' 'rm_multi_missing_species.R'
'rm_nonendemic_in_clade.R' 'sensitivity.R' 'translate_status.R'
'unique_island_genera.R' 'utils.R' 'write_biogeobears_input.R'

NeedsCompilation no

Author Joshua W. Lambert [aut, cre] (<`https://orcid.org/0000-0001-5218-3046`>),
Luis Valente [aut] (<`https://orcid.org/0000-0003-4247-8785`>),
Pedro Santos Neves [aut] (<`https://orcid.org/0000-0003-2561-4677`>),
Lizzie Roeble [aut] (<`https://orcid.org/0000-0003-3664-4222`>),
Theo Pannetier [aut] (<`https://orcid.org/0000-0002-8424-3573`>)

Repository CRAN

Date/Publication 2024-12-18 00:20:02 UTC

Contents

add_asr_node_states 4
add_island_colonist 5
add_missing_species 7
add_multi_missing_species 8
add_outgroup 9
all_descendants_conspecific 10
all_endemicity_status 10
any_back_colonisation 11
any_outgroup 12
any_polyphyly 13
as_daisie_datatable 13
benchmark 14
bind_colonist_to_tbl 16
check_island_colonist 17

check_island_tbl 18
check_multi_island_tbl 18
check_phylo_data 19
coccyzus_phylod 20
columbiformes_phylod 20
count_missing_species 21
create_daisie_data 22
create_endemicity_status 24
create_test_phylod 25
default_params_doc 26
endemicity_to_sse_states 32
extract_asr_clade 33
extract_biogeobears_ancestral_states_probs 33
extract_clade_name 34
extract_endemic_clade 34
extract_endemic_singleton 35
extract_island_species 36
extract_multi_tip_species 38
extract_nonendemic 39
extract_species_asr 40
extract_species_min 42
extract_stem_age 43
extract_stem_age_asr 45
extract_stem_age_genus 45
extract_stem_age_min 46
finches_phylod 47
GalapagosTrees 47
get_clade_name 48
get_island_tbl 51
get_sse_tip_states 52
island_colonist 53
Island_colonist-class 54
island_tbl 55
Island_tbl-class 55
is_back_colonisation 56
is_duplicate_colonist 57
is_identical_island_tbl 58
mimus_phylod 59
multi_extract_island_species 59
multi_island_tbl 61
Multi_island_tbl-class 61
myiarchus_phylod 61
plant_phylo 62
plot_colonisation 62
plot_performance 63
plot_phylod 64
progne_phylod 65
pyrocephalus_phylod 65

rm_island_colonist 66
rm_multi_missing_species 66
round_up 68
select_endemicity_status 68
sensitivity 69
setophaga_phylod 70
sse_states_to_endemicity 71
translate_status 71
unique_island_genera 72
write_biogeobears_input 73
write_newick_file 73
write_phylip_biogeo_file 74

Index **75**

`add_asr_node_states` *Fits a model of ancestral state reconstruction of island presence*

Description

Fits a model of ancestral state reconstruction of island presence

Usage

```
add_asr_node_states(
  phylod,
  asr_method,
  tie_preference = "island",
  earliest_col = FALSE,
  rate_model = NULL,
  ...
)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`asr_method`	A character string, either "parsimony" or "mk" determines whether a maximum parsimony or continuous-time markov model reconstructs the ancestral states at each node. See documentation in `castor::asr_max_parsimony()` or `castor::asr_mk_model()` in `castor` R package for details on the methods used.
`tie_preference`	Character string, either "island" or "mainland" to choose the most probable state at each node using the `max.col()` function. When a node has island presence and absence equally probable we need to decide whether that species should be considered on the island. To consider it on the island use `ties.method = "last"` in the `max.col()` function, if you consider it not on the island use `ties.method = "first"`. Default is "island".

`earliest_col`	A boolean to determine whether to take the colonisation time as the most probable time (FALSE) or the earliest possible colonisation time (TRUE), where the probability of a species being on the island is non-zero. Default is FALSE.
`rate_model`	Rate model to be used for fitting the transition rate matrix. Can be "ER" (all rates equal), "SYM" (transition rate i–>j is equal to transition rate j–>i), "ARD" (all rates can be different), "SUEDE" (only stepwise transitions i–>i+1 and i–>i-1 allowed, all 'up' transitions are equal, all 'down' transitions are equal) or "SRD" (only stepwise transitions i–>i+1 and i–>i-1 allowed, and each rate can be different). Can also be an index matrix that maps entries of the transition matrix to the corresponding independent rate parameter to be fitted. Diagonal entries should map to 0, since diagonal entries are not treated as independent rate parameters but are calculated from the remaining entries in the transition matrix. All other entries that map to 0 represent a transition rate of zero. The format of this index matrix is similar to the format used by the `ace` function in the `ape` package. `rate_model` is only relevant if `transition_matrix==NULL`.
`...`	dots Allows arguments to be passed to `castor::asr_mk_model()` and `castor::asr_max_parsimony()`. These arguments must match by name exactly, see `?castor::asr_mk_model()` and `?castor::asr_max_parsimony()` for information on arguments.

Details

The `rate_model` argument documentation is inherited from `castor::asr_mk_model()`, therefore, the last sentence about the `transition_matrix` argument does not apply to `add_asr_node_states()`.

Value

An object of `phylo4d` class with tip and node data

`add_island_colonist` *Adds an island colonists (can be either a singleton lineage or an island clade) to the island community (island_tbl).*

Description

Adds an island colonists (can be either a singleton lineage or an island clade) to the island community (island_tbl).

Usage

```
add_island_colonist(
  island_tbl,
  clade_name,
  status,
  missing_species,
  col_time,
  col_max_age,
  branching_times,
```

```
  min_age,
  species,
  clade_type
)
```

Arguments

`island_tbl`	An instance of the `Island_tbl` class.
`clade_name`	Character name of the colonising clade.
`status`	Character endemicity status of the colonising clade. Either `"endemic"` or `"nonendemic"`.
`missing_species`	Numeric number of missing species from the phylogeny that belong to the colonising clade. For a clade with missing species this is $n - 1$, where n is the number of missing species in the clade. If the clade is an island singleton, the number of missing species is `0` because by adding the colonist it already counts as one automatically. If the clade has more than one species, the `missing_species` is $n - 1$ because adding the lineage already counts as one.
`col_time`	Numeric with the colonisation time of the island colonist
`col_max_age`	Boolean determining whether colonisation time should be considered a precise time of colonisation or a maximum time of colonisation
`branching_times`	Numeric vector of one or more elements which are the branching times on the island.
`min_age`	Numeric minimum age (time before the present) that the species must have colonised the island by. This is known when there is a branching on the island, either in species or subspecies.
`species`	Character vector of one or more elements containing the name of the species included in the colonising clade.
`clade_type`	Numeric determining which type of clade the island colonist is, this determines which macroevolutionary regime (parameter set) the island colonist is in. After formatting the `island_tbl` to a DAISIE data list, the clade type can be used to conduct a 2-type analysis (see `https://CRAN.R-project.org/package=DAISIE/vignettes/demo_optimize.html` for more information)

Value

An object of `Island_tbl` class

Examples

```
# create an empty island_tbl to add to
island_tbl <- island_tbl()

# add a new island colonist
island_tbl <- add_island_colonist(
  island_tbl,
  clade_name = "new_clade",
```

```
  status = "endemic",
  missing_species = 0,
  col_time = 1,
  col_max_age = FALSE,
  branching_times = NA,
  min_age = NA,
  species = "new_clade",
  clade_type = 1
)
```

`add_missing_species` *Adds a specified number of missing species to an existing island_tbl at the colonist specified by the species_to_add_to argument given. The species given is located within the island_tbl data and missing species are assigned. This is to be used after* `extract_island_species()` *to input missing species.*

Description

Adds a specified number of missing species to an existing island_tbl at the colonist specified by the species_to_add_to argument given. The species given is located within the island_tbl data and missing species are assigned. This is to be used after `extract_island_species()` to input missing species.

Usage

```
add_missing_species(island_tbl, num_missing_species, species_to_add_to)
```

Arguments

`island_tbl`	An instance of the `Island_tbl` class.
`num_missing_species`	Numeric for the number of missing species in the clade.
`species_to_add_to`	Character string with the name of the species to identify which clade to assign missing species to.

Value

Object of Island_tbl class

Examples

```
set.seed(
  1,
  kind = "Mersenne-Twister",
  normal.kind = "Inversion",
  sample.kind = "Rejection"
```

```
)
phylo <- ape::rcoal(5)
phylo$tip.label <- c("bird_a", "bird_b", "bird_c", "bird_d", "bird_e")
phylo <- phylobase::phylo4(phylo)
endemicity_status <- c(
  "not_present", "not_present", "endemic", "not_present", "not_present"
)
phylod <- phylobase::phylo4d(phylo, as.data.frame(endemicity_status))
island_tbl <- extract_island_species(phylod, extraction_method = "min")
island_tbl <- add_missing_species(
  island_tbl = island_tbl,
  num_missing_species = 1,
  species_to_add_to = "bird_c"
)
```

`add_multi_missing_species` *Calculates the number of missing species to be assigned to each island clade in the island_tbl object and assigns the missing species to them. In the case that multiple genera are in an island clade and each have missing species the number of missing species is summed. Currently the missing species are assigned to the genus that first matches with the missing species table, however a more biologically or stochastic assignment is in development.*

Description

Calculates the number of missing species to be assigned to each island clade in the island_tbl object and assigns the missing species to them. In the case that multiple genera are in an island clade and each have missing species the number of missing species is summed. Currently the missing species are assigned to the genus that first matches with the missing species table, however a more biologically or stochastic assignment is in development.

Usage

```
add_multi_missing_species(missing_species, missing_genus, island_tbl)
```

Arguments

`missing_species`	Numeric number of missing species from the phylogeny that belong to the colonising clade. For a clade with missing species this is $n - 1$, where n is the number of missing species in the clade. If the clade is an island singleton, the number of missing species is `0` because by adding the colonist it already counts as one automatically. If the clade has more than one species, the `missing_species` is $n - 1$ because adding the lineage already counts as one.
`missing_genus`	A list of character vectors containing the genera in each island clade
`island_tbl`	An instance of the `Island_tbl` class.

Value

Object of Island_tbl class

Examples

```
phylod <- create_test_phylod(test_scenario = 6)
island_tbl <- suppressWarnings(extract_island_species(
  phylod = phylod,
  extraction_method = "asr",
))
phylod <- create_test_phylod(test_scenario = 7)
island_tbl <- suppressWarnings(extract_island_species(
  phylod = phylod,
  extraction_method = "asr",
  island_tbl = island_tbl
))

missing_species <- data.frame(
  clade_name = "bird",
  missing_species = 1,
  endemicity_status = "endemic"
)

missing_genus <- list("bird", character(0))

island_tbl <- add_multi_missing_species(
  missing_species = missing_species,
  missing_genus = missing_genus,
  island_tbl = island_tbl
)
```

`add_outgroup` *Add an outgroup species to a given phylogeny.*

Description

Add an outgroup species to a given phylogeny.

Usage

```
add_outgroup(phylo)
```

Arguments

`phylo`	A phylogeny either as a `phylo` (from the `ape` package) or `phylo4` (from the `phylobase` package) object.

Value

A `phylo` object

Examples

```
phylo <- ape::rcoal(10)
phylo_with_outgroup <- add_outgroup(phylo)
```

`all_descendants_conspecific` *Checks whether all species given in the descendants vector are the same species.*

Description

Checks whether all species given in the descendants vector are the same species.

Usage

```
all_descendants_conspecific(descendants)
```

Arguments

`descendants`	A vector character strings with the names of species to determine whether they are the same species.

Value

Boolean

Examples

```
# Example where species are not conspecific
descendants <- c("bird_a", "bird_b", "bird_c")
all_descendants_conspecific(descendants = descendants)

# Example where species are conspecific
descendants <- c("bird_a_1", "bird_a_2", "bird_a_3")
all_descendants_conspecific(descendants = descendants)
```

`all_endemicity_status` *All possible endemicity statuses*

Description

All possible endemicity statuses

Usage

```
all_endemicity_status()
```

Value

A vector of character strings with all the endemicity status options

---

`any_back_colonisation` *Detects any cases where a non-endemic species or species not present on the island has likely been on the island given its ancestral state reconstruction indicating ancestral presence on the island and so is likely a back colonisation from the island to the mainland (or potentially different island). This function is useful if using extraction_method = "min" in* `DAISIEprep::extract_island_species()` *as it may brake up a single colonist into multiple colonists because of back-colonisation.*

Description

Detects any cases where a non-endemic species or species not present on the island has likely been on the island given its ancestral state reconstruction indicating ancestral presence on the island and so is likely a back colonisation from the island to the mainland (or potentially different island). This function is useful if using extraction_method = "min" in `DAISIEprep::extract_island_species()` as it may brake up a single colonist into multiple colonists because of back-colonisation.

Usage

```
any_back_colonisation(phylod, only_tips = FALSE)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`only_tips`	A boolean determing whether only the tips (i.e. terminal branches) are searched for back colonisation events.

Value

A single or vector of character strings. Character string is in the format ancestral_node -> focal_node, where the ancestral node is not on mainland but the focal node is. In the case of no back colonisations a different message string is returned.

Examples

```
# Example with no back colonisation
phylod <- create_test_phylod(test_scenario = 15)
any_back_colonisation(phylod)

# Example with back colonisation
set.seed(
  3,
  kind = "Mersenne-Twister",
```

```
  normal.kind = "Inversion",
  sample.kind = "Rejection"
)
phylo <- ape::rcoal(5)
phylo$tip.label <- c("bird_a", "bird_b", "bird_c", "bird_d", "bird_e")
phylo <- phylobase::phylo4(phylo)
endemicity_status <- c("endemic", "endemic", "not_present",
                       "endemic", "not_present")
phylod <- phylobase::phylo4d(phylo, as.data.frame(endemicity_status))
phylod <- add_asr_node_states(phylod = phylod, asr_method = "parsimony")
# aritificially modify data to produce back-colonisation
phylobase::tdata(phylod)$island_status[8] <- "endemic"
any_back_colonisation(phylod = phylod)
```

`any_outgroup` *Checks whether the phylogeny has an outgroup that is not present on the island. This is critical when extracting data from the phylogeny so the stem age (colonisation time) is correct.*

Description

Checks whether the phylogeny has an outgroup that is not present on the island. This is critical when extracting data from the phylogeny so the stem age (colonisation time) is correct.

Usage

```
any_outgroup(phylod)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.

Value

Boolean

Examples

```
set.seed(
  1,
  kind = "Mersenne-Twister",
  normal.kind = "Inversion",
  sample.kind = "Rejection"
)
phylo <- ape::rcoal(10)
phylo$tip.label <- c("bird_a", "bird_b", "bird_c", "bird_d", "bird_e",
                     "bird_f", "bird_g", "bird_h", "bird_i", "bird_j")
phylo <- phylobase::phylo4(phylo)
```

```
endemicity_status <- sample(c("not_present", "endemic", "nonendemic"),
                            size = length(phylobase::tipLabels(phylo)),
                            replace = TRUE)
phylod <- phylobase::phylo4d(phylo, as.data.frame(endemicity_status))
any_outgroup(phylod)
```

`any_polyphyly` *Checks whether there are any species in the phylogeny that have multiple tips (i.e. multiple subspecies per species) and whether any of those tips are paraphyletic (i.e. are their subspecies more distantly related to each other than to other subspecies or species).*

Description

Checks whether there are any species in the phylogeny that have multiple tips (i.e. multiple subspecies per species) and whether any of those tips are paraphyletic (i.e. are their subspecies more distantly related to each other than to other subspecies or species).

Usage

```
any_polyphyly(phylod)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.

Value

Boolean

Examples

```
phylod <- create_test_phylod(test_scenario = 1)
any_polyphyly(phylod)
```

`as_daisie_datatable` *Converts the* `Island_tbl` *class to a data frame in the format of a DAISIE data table (see DAISIE R package for details). This can then be input into* `DAISIEprep::create_daisie_data()` *function which creates the list input into the DAISIE ML models.*

Description

Converts the `Island_tbl` class to a data frame in the format of a DAISIE data table (see DAISIE R package for details). This can then be input into `DAISIEprep::create_daisie_data()` function which creates the list input into the DAISIE ML models.

Usage

```
as_daisie_datatable(island_tbl, island_age, precise_col_time = TRUE)
```

Arguments

`island_tbl`	An instance of the `Island_tbl` class.
`island_age`	Age of the island in appropriate units.
`precise_col_time`	Boolean, TRUE uses the precise times of colonisation, FALSE makes every colonist a max age colonistion and uses minimum age of colonisation if available.

Value

A data frame in the format of a DAISIE data table

Author(s)

Joshua W. Lambert, Pedro Neves

Examples

```
phylod <- create_test_phylod(10)
island_tbl <- extract_island_species(
  phylod = phylod,
  extraction_method = "asr"
)

# Example where precise colonisation times are known
daisie_datatable <- as_daisie_datatable(
  island_tbl = island_tbl,
  island_age = 0.2,
  precise_col_time = TRUE
)

# Example where colonisation times are uncertain and set to max ages
daisie_datatable <- as_daisie_datatable(
  island_tbl = island_tbl,
  island_age = 0.2,
  precise_col_time = FALSE
)
```

`benchmark` *Performance analysis of the extract_island_species() function Uses system.time() for timing for reasons explained here: https://radfordneal.wordpress.com/2014/02/02/inaccurate-results-from-microbenchmark/ # nolint*

Description

Performance analysis of the extract_island_species() function Uses system.time() for timing for reasons explained here: https://radfordneal.wordpress.com/2014/02/02/inaccurate-results-from-microbenchmark/ # nolint

Usage

```
benchmark(
  phylod,
  tree_size_range,
  num_points,
  prob_on_island,
  prob_endemic,
  replicates,
  extraction_method,
  asr_method,
  tie_preference,
  log_scale = TRUE,
  parameter_index = NULL,
  verbose = FALSE
)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`tree_size_range`	Numeric vector of two elements, the first is the smallest tree size (number of tips) and the second is the largest tree size
`num_points`	Numeric determining how many points in the sequence of smallest tree size to largest tree size
`prob_on_island`	Numeric vector of each probability on island to use in the parameter space
`prob_endemic`	Numeric vector of each probability of an island species being endemic to use in the parameter space
`replicates`	Numeric determining the number of replicates to use to account for the stochasticity in sampling the species on the island and endemic species
`extraction_method`	A character string specifying whether the colonisation time extracted is the minimum time (`min`) (before the present), or the most probable time under ancestral state reconstruction (`asr`).
`asr_method`	A character string, either "parsimony" or "mk" determines whether a maximum parsimony or continuous-time markov model reconstructs the ancestral states at each node. See documentation in `castor::asr_max_parsimony()` or `castor::asr_mk_model()` in `castor` R package for details on the methods used.

`tie_preference`	Character string, either "island" or "mainland" to choose the most probable state at each node using the `max.col()` function. When a node has island presence and absence equally probable we need to decide whether that species should be considered on the island. To consider it on the island use `ties.method = "last"` in the `max.col()` function, if you consider it not on the island use `ties.method = "first"`. Default is "island".
`log_scale`	A boolean determining whether the sequence of tree sizes are on a linear (FALSE) or log (TRUE) scale
`parameter_index`	Numeric determining which parameter set to use (i.e which row in the parameter space data frame), if this is NULL all parameter sets will be looped over
`verbose`	Boolean. States if intermediate results should be printed to console. Defaults to FALSE

Value

Data frame

`bind_colonist_to_tbl` *Takes an existing instance of an* `Island_tbl` *class and bind the information from the instance of an* `Island_colonist` *class to it*

Description

Takes an existing instance of an `Island_tbl` class and bind the information from the instance of an `Island_colonist` class to it

Usage

```
bind_colonist_to_tbl(island_colonist, island_tbl)
```

Arguments

`island_colonist`	An instance of the `Island_colonist` class.
`island_tbl`	An instance of the `Island_tbl` class.

Value

An object of `Island_tbl` class

Examples

```
island_colonist <- DAISIEprep::island_colonist(
  clade_name = "bird",
  status = "endemic",
  missing_species = 0,
  col_time = 1,
  col_max_age = FALSE,
  branching_times = 0.5,
  species = "bird_a",
  clade_type = 1
)
island_tbl <- island_tbl()
bind_colonist_to_tbl(
  island_colonist = island_colonist,
  island_tbl = island_tbl
)
```

`check_island_colonist` *Checks the validity of the Island_colonist class*

Description

Checks the validity of the Island_colonist class

Usage

```
check_island_colonist(object)
```

Arguments

`object`	Instance of the island_colonist class

Value

Boolean or errors

Examples

```
island_colonist <- island_colonist()
check_island_colonist(island_colonist)
```

`check_island_tbl` *Checks the validity of the Island_tbl class*

Description

Checks the validity of the Island_tbl class

Usage

```
check_island_tbl(object)
```

Arguments

`object`	Instance of the Island_tbl class

Value

Boolean or errors

Examples

```
island_tbl <- island_tbl()
check_island_tbl(island_tbl)
```

`check_multi_island_tbl` *Checks the validity of the Multi_island_tbl class*

Description

Checks the validity of the Multi_island_tbl class

Usage

```
check_multi_island_tbl(object)
```

Arguments

`object`	Instance of the Multi_island_tbl class

Value

Boolean or errors

Examples

```
multi_island_tbl <- multi_island_tbl()
check_multi_island_tbl(multi_island_tbl)
```

`check_phylo_data` *Checks whether* `\linkS4class{phylo4d}` *object conforms to the requirements of the DAISIEprep package. If the function does not return anything the data is ready to be used, if an error is returned the data requires some pre-processing before DAISIEprep can be used*

Description

Checks whether `\linkS4class{phylo4d}` object conforms to the requirements of the DAISIEprep package. If the function does not return anything the data is ready to be used, if an error is returned the data requires some pre-processing before DAISIEprep can be used

Usage

```
check_phylo_data(phylod)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.

Value

Nothing or error message

Examples

```
set.seed(
  1,
  kind = "Mersenne-Twister",
  normal.kind = "Inversion",
  sample.kind = "Rejection"
)
phylo <- ape::rcoal(10)
phylo$tip.label <- c("bird_a", "bird_b", "bird_c", "bird_d", "bird_e",
                     "bird_f", "bird_g", "bird_h", "bird_i", "bird_j")
phylo <- phylobase::phylo4(phylo)
endemicity_status <- sample(
  c("not_present", "endemic", "nonendemic"),
  size = length(phylobase::tipLabels(phylo)),
  replace = TRUE,
  prob = c(0.6, 0.2, 0.2)
)
phylod <- phylobase::phylo4d(phylo, as.data.frame(endemicity_status))
check_phylo_data(phylod)
```

`coccyzus_phylod` *A phylogenetic tree of coccyzus species with endemicity status as tip states.*

Description

A phylogenetic tree of coccyzus species with endemicity status as tip states.

Usage

```
coccyzus_phylod
```

Format

`coccyzus_phylod`**:**
A `phylo4d` object (from the **phylobase** package) with 20 tips and 19 internal nodes.

Source

Valente, L.M., A.B. Phillimore and R.S. Etienne (2015). Equilibrium and non-equilibrium dynamics simultaneously operate in the Galapagos islands. Ecology Letters 18: 844-852.

`columbiformes_phylod` *A phylogenetic tree of columbiformes species with endemicity status as tip states.*

Description

A phylogenetic tree of columbiformes species with endemicity status as tip states.

Usage

```
columbiformes_phylod
```

Format

`columbiformes_phylod`**:**
A `phylo4d` object (from the **phylobase** package) with 242 tips and 241 internal nodes.

Source

Valente, L.M., A.B. Phillimore and R.S. Etienne (2015). Equilibrium and non-equilibrium dynamics simultaneously operate in the Galapagos islands. Ecology Letters 18: 844-852.

`count_missing_species` *Reads in the checklist of all species on an island, including those that are not in the phylogeny (represented by NA) and counts the number of species missing from the phylogeny each genus*

Description

Reads in the checklist of all species on an island, including those that are not in the phylogeny (represented by NA) and counts the number of species missing from the phylogeny each genus

Usage

```
count_missing_species(
  checklist,
  phylo_name_col,
  genus_name_col,
  in_phylo_col,
  endemicity_status_col,
  rm_species_col = NULL
)
```

Arguments

`checklist`	data frame with information on species on the island
`phylo_name_col`	A character string specifying the column name where the names in the phylogeny are in the checklist
`genus_name_col`	A character string specifying the column name where the genus names are in the checklist
`in_phylo_col`	A character string specifying the column name where the status of whether a species is in the phylogeny is in the checklist
`endemicity_status_col`	A character string specifying the column name where the endemicity status of the species are in the checklist
`rm_species_col`	A character string specifying the column name where the information on whether to remove species from the checklist before counting the number of missing species is in the checklist. This can be NULL if no species are to be removed from the checklist. This is useful when species are in the checklist because they are on the island but need to be removed as they are not in the group of interest, e.g. a migratory bird amongst terrestrial birds

Value

Data frame

Examples

```
mock_checklist <- data.frame(
  genus = c("bird", "bird", "bird", "bird", "bird", "bird", "bird",
            "bird", "bird", "bird"),
  species = c("a", "b", "c", "d", "e", "f", "g", "h", "i", "j"),
  species_names = c("bird_a", "bird_b", "bird_c", "bird_d", "bird_e",
                    "bird_f","bird_g", "bird_h", "bird_i", "bird_j"),
  sampled = c(TRUE, TRUE, TRUE, TRUE, TRUE, FALSE, TRUE, FALSE, TRUE, FALSE),
  endemicity_status = c("endemic", "endemic", "endemic", "nonendemic",
                        "endemic", "nonendemic", "endemic", "endemic",
                        "endemic", "endemic"),
  remove_species = (rep(FALSE, 10))
)

missing_species <- count_missing_species(
  checklist = mock_checklist,
  phylo_name_col = "species_names",
  genus_name_col = "genus",
  in_phylo_col = "sampled",
  endemicity_status_col = "endemicity_status",
  rm_species_col = NULL
)
```

`create_daisie_data` *This is a wrapper function for DAISIE::DAISIE_dataprep(). It allows the final DAISIE data structure to be produced from within DAISIEprep. For detailed documentation see the help documentation in the DAISIE package (?DAISIE::DAISIE_dataprep).*

Description

This is a wrapper function for DAISIE::DAISIE_dataprep(). It allows the final DAISIE data structure to be produced from within DAISIEprep. For detailed documentation see the help documentation in the DAISIE package (?DAISIE::DAISIE_dataprep).

Usage

```
create_daisie_data(
  data,
  island_age,
  num_mainland_species,
  num_clade_types = 1,
  list_type2_clades = NA,
  prop_type2_pool = "proportional",
  epss = 1e-05,
  verbose = FALSE,
  precise_col_time = TRUE
)
```

Arguments

`data`	Either an object of class `Island_tbl` or a DAISIE data table object (output from `as_daisie_datatable()`).
`island_age`	Age of the island in appropriate units.
`num_mainland_species`	The size of the mainland pool, i.e. the number of species that can potentially colonise the island.
`num_clade_types`	Number of clade types. Default num_clade_types = 1 all species are considered to belong to the same macroevolutionary process. If num_clade_types = 2, there are two types of clades with distinct macroevolutionary processes.
`list_type2_clades`	If num_clade_types = 2, list_type2_clades specifies the names of the clades that have a distinct macroevolutionary process. The names must match those in the "Clade_name" column of the source data table. If num_clade_types = 1, then list_type2_clades = NA should be specified (default).
`prop_type2_pool`	Specifies the fraction of potential mainland colonists that have a distinct macroevolutionary process. Applies only if number_clade_types = 2. Default "proportional" sets the fraction to be proportional to the number of clades of distinct macroevolutionary process that have colonised the island. Alternatively, the user can specify a value between 0 and 1 (e.g. if the mainland pool size is 1000 and prop_type2_pool = 0.02 then the number of type 2 species is 20).
`epss`	Default = 1e-5 should be appropriate in most cases. This value is used to set the maximum age of colonisation of "Non_endemic_MaxAge" and "Endemic_MaxAge" species to an age that is slightly younger than the island for cases when the age provided for that species is older than the island. The new maximum age is then used as an upper bound to integrate over all possible colonisation times.
`verbose`	Boolean. States if intermediate results should be printed to console. Defaults to FALSE
`precise_col_time`	Boolean, TRUE uses the precise times of colonisation, FALSE makes every colonist a max age colonistion and uses minimum age of colonisation if available.

Value

DAISIE data list

Examples

```
phylod <- create_test_phylod(3)
island_tbl <- extract_island_species(
  phylod = phylod,
  extraction_method = "min"
)
```

```
daisie_datatable <- as_daisie_datatable(island_tbl, island_age = 10)
daisie_data_list <- create_daisie_data(
  data = daisie_datatable,
  island_age = 10,
  num_mainland_species = 1000,
  num_clade_types = 1,
  list_type2_clades = NA,
  prop_type2_pool = NA,
  epss = 1e-5,
  verbose = FALSE
)
```

`create_endemicity_status` *Creates a data frame with the endemicity status (either 'endemic', 'nonendemic', 'not_present') of every species in the phylogeny using a phylogeny and a data frame of the island species and their endemicity (either 'endemic' or 'nonendemic') provided.*

Description

Creates a data frame with the endemicity status (either 'endemic', 'nonendemic', 'not_present') of every species in the phylogeny using a phylogeny and a data frame of the island species and their endemicity (either 'endemic' or 'nonendemic') provided.

Usage

```
create_endemicity_status(phylo, island_species)
```

Arguments

`phylo`	A phylogeny either as a `phylo` (from the `ape` package) or `phylo4` (from the `phylobase` package) object.
`island_species`	Data frame with two columns. The first is a character string of the tip_labels with the tip names of the species on the island. The second column a character string of the endemicity status of the species, either endemic or nonendemic.

Details

Species included in the `island_species` data frame but not included in the `phylo` will not be included in the output and warning will print all of the species that are in the `island_species` that are not found in the `phylo`.

Value

Data frame with single column of character strings and row names

Examples

```
set.seed(
  1,
  kind = "Mersenne-Twister",
  normal.kind = "Inversion",
  sample.kind = "Rejection"
)
phylo <- ape::rcoal(4)
phylo$tip.label <- c("species_a", "species_b", "species_c", "species_d")
phylo <- methods::as(phylo, "phylo4")
island_species <- data.frame(
  tip_labels = c("species_a", "species_b", "species_c", "species_d"),
  tip_endemicity_status = c("endemic", "endemic", "endemic", "nonendemic")
)
endemicity_status <- create_endemicity_status(
  phylo = phylo,
  island_species = island_species
)
```

`create_test_phylod` *Creates phylod objects.*

Description

A helper function that is useful in tests and examples to easily create `phylod` objects (i.e. phylogenetic trees with data).

Usage

```
create_test_phylod(test_scenario)
```

Arguments

`test_scenario`	Integer specifying which test phylod object to create.

Value

A `phylo4d` object

Examples

```
create_test_phylod(test_scenario = 1)
```

`default_params_doc` *Documentation for function in the DAISIEprep package*

Description

Documentation for function in the DAISIEprep package

Usage

```
default_params_doc(
  island_colonist,
  island_tbl,
  phylod,
  extraction_method,
  species_label,
  species_endemicity,
  x,
  value,
  clade_name,
  status,
  missing_species,
  col_time,
  col_max_age,
  branching_times,
  min_age,
  species,
  clade_type,
  endemic_clade,
  phylo,
  island_species,
  descendants,
  clade,
  asr_method,
  tie_preference,
  earliest_col,
  include_not_present,
  nested_asr_species,
  num_missing_species,
  species_to_add_to,
  node_pies,
  test_scenario,
  data,
  island_age,
  num_mainland_species,
  num_clade_types,
  list_type2_clades,
  prop_type2_pool,
```

```
  epss,
  verbose,
  precise_col_time,
  n,
  digits,
  include_crown_age,
  only_tips,
  node_label,
  multi_phylod,
  island_tbl_1,
  island_tbl_2,
  unique_clade_name,
  genus_name,
  stem,
  genus_in_tree,
  missing_genus,
  checklist,
  phylo_name_col,
  genus_name_col,
  in_phylo_col,
  endemicity_status_col,
  rm_species_col,
  tree_size_range,
  num_points,
  prob_on_island,
  prob_endemic,
  replicates,
  log_scale,
  parameter_index,
  sse_model,
  force_nonendemic_singleton,
  ...
)
```

Arguments

`island_colonist`	An instance of the `Island_colonist` class.
`island_tbl`	An instance of the `Island_tbl` class.
`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`extraction_method`	A character string specifying whether the colonisation time extracted is the minimum time (`min`) (before the present), or the most probable time under ancestral state reconstruction (`asr`).
`species_label`	The tip label of the species of interest.

`species_endemicity`	A character string with the endemicity, either "endemic" or "nonendemic" of an island species, or "not_present" if not on the island.
`x`	An object whose class is determined by the signature.
`value`	A value which can take several forms to be assigned to an object of a class.
`clade_name`	Character name of the colonising clade.
`status`	Character endemicity status of the colonising clade. Either `"endemic"` or `"nonendemic"`.
`missing_species`	Numeric number of missing species from the phylogeny that belong to the colonising clade. For a clade with missing species this is $n - 1$, where n is the number of missing species in the clade. If the clade is an island singleton, the number of missing species is `0` because by adding the colonist it already counts as one automatically. If the clade has more than one species, the `missing_species` is $n - 1$ because adding the lineage already counts as one.
`col_time`	Numeric with the colonisation time of the island colonist
`col_max_age`	Boolean determining whether colonisation time should be considered a precise time of colonisation or a maximum time of colonisation
`branching_times`	Numeric vector of one or more elements which are the branching times on the island.
`min_age`	Numeric minimum age (time before the present) that the species must have colonised the island by. This is known when there is a branching on the island, either in species or subspecies.
`species`	Character vector of one or more elements containing the name of the species included in the colonising clade.
`clade_type`	Numeric determining which type of clade the island colonist is, this determines which macroevolutionary regime (parameter set) the island colonist is in. After formatting the `island_tbl` to a DAISIE data list, the clade type can be used to conduct a 2-type analysis (see `https://CRAN.R-project.org/package=DAISIE/vignettes/demo_optimize.html` for more information)
`endemic_clade`	Named vector with all the species from a clade.
`phylo`	A phylogeny either as a `phylo` (from the `ape` package) or `phylo4` (from the `phylobase` package) object.
`island_species`	Data frame with two columns. The first is a character string of the tip_labels with the tip names of the species on the island. The second column a character string of the endemicity status of the species, either endemic or nonendemic.
`descendants`	A vector character strings with the names of species to determine whether they are the same species.
`clade`	A numeric vector which the indices of the species which are in the island clade.
`asr_method`	A character string, either "parsimony" or "mk" determines whether a maximum parsimony or continuous-time markov model reconstructs the ancestral states at each node. See documentation in `castor::asr_max_parsimony()` or `castor::asr_mk_model()` in `castor` R package for details on the methods used.

`tie_preference`	Character string, either "island" or "mainland" to choose the most probable state at each node using the `max.col()` function. When a node has island presence and absence equally probable we need to decide whether that species should be considered on the island. To consider it on the island use `ties.method = "last"` in the `max.col()` function, if you consider it not on the island use `ties.method = "first"`. Default is "island".
`earliest_col`	A boolean to determine whether to take the colonisation time as the most probable time (FALSE) or the earliest possible colonisation time (TRUE), where the probability of a species being on the island is non-zero. Default is FALSE.
`include_not_present`	A boolean determining whether species not present on the island should be included in island colonist when embedded within an island clade. Default is FALSE.
`nested_asr_species`	A `character` string which determines whether *nested island colonists* are split into separate colonists (`"split"`), or grouped into a single clade (`"group"`). Nested species are those whose tip state is on the island, and they have ancestral nodes on the island, but there are nodes in between these island state nodes that have the state `not_present` (i.e. not on the island). Therefore, the colonisation time can be extracted as the most recent node state on the island (this can be the branching time before the tip if the ancestor node of the tip is not on the island), or the older node state of the larger clade, for `"split"` or `"group"` respectively. **Note** This argument only applies when `extraction_method = "asr"`.
`num_missing_species`	Numeric for the number of missing species in the clade.
`species_to_add_to`	Character string with the name of the species to identify which clade to assign missing species to.
`node_pies`	Boolean determining if pie charts of the probabilities of a species being present on the island. If TRUE the correct data is required in the phylod object.
`test_scenario`	Integer specifying which test phylod object to create.
`data`	Either an object of class `Island_tbl` or a DAISIE data table object (output from `as_daisie_datatable()`).
`island_age`	Age of the island in appropriate units.
`num_mainland_species`	The size of the mainland pool, i.e. the number of species that can potentially colonise the island.
`num_clade_types`	Number of clade types. Default num_clade_types = 1 all species are considered to belong to the same macroevolutionary process. If num_clade_types = 2, there are two types of clades with distinct macroevolutionary processes.
`list_type2_clades`	If num_clade_types = 2, list_type2_clades specifies the names of the clades that have a distinct macroevolutionary process. The names must match those in the "Clade_name" column of the source data table. If num_clade_types = 1, then list_type2_clades = NA should be specified (default).

`prop_type2_pool`	Specifies the fraction of potential mainland colonists that have a distinct macroevolutionary process. Applies only if number_clade_types = 2. Default "proportional" sets the fraction to be proportional to the number of clades of distinct macroevolutionary process that have colonised the island. Alternatively, the user can specify a value between 0 and 1 (e.g. if the mainland pool size is 1000 and prop_type2_pool = 0.02 then the number of type 2 species is 20).
`epss`	Default = 1e-5 should be appropriate in most cases. This value is used to set the maximum age of colonisation of "Non_endemic_MaxAge" and "Endemic_MaxAge" species to an age that is slightly younger than the island for cases when the age provided for that species is older than the island. The new maximum age is then used as an upper bound to integrate over all possible colonisation times.
`verbose`	Boolean. States if intermediate results should be printed to console. Defaults to FALSE
`precise_col_time`	Boolean, TRUE uses the precise times of colonisation, FALSE makes every colonist a max age colonistion and uses minimum age of colonisation if available.
`n`	A numeric to be rounded.
`digits`	A numeric specifying which decimal places to round to
`include_crown_age`	A boolean determining whether the crown age gets plotted with the stem age.
`only_tips`	A boolean determing whether only the tips (i.e. terminal branches) are searched for back colonisation events.
`node_label`	A numeric label for a node within a phylogeny.
`multi_phylod`	A list of phylod objects.
`island_tbl_1`	An object of `Island_tbl` class to be comparedl
`island_tbl_2`	An object of `Island_tbl` class to be compared
`unique_clade_name`	Boolean determining whether a unique species identifier is used as the clade name in the Island_tbl object or a genus name which may not be unique if that genus has several independent island colonisations
`genus_name`	Character string of genus name to be matched with a genus name from the tip labels in the phylogeny
`stem`	Character string, either "genus" or "island_presence". The former will extract the stem age of the genussbased on the genus name provided, the latter will extract the stem age based on the ancestral presence on the island either based on the "min" or "asr" extraction algorithms.
`genus_in_tree`	A numeric vector that indicates which species in the genus are in the tree
`missing_genus`	A list of character vectors containing the genera in each island clade
`checklist`	data frame with information on species on the island
`phylo_name_col`	A character string specifying the column name where the names in the phylogeny are in the checklist

`genus_name_col`	A character string specifying the column name where the genus names are in the checklist
`in_phylo_col`	A character string specifying the column name where the status of whether a species is in the phylogeny is in the checklist
`endemicity_status_col`	A character string specifying the column name where the endemicity status of the species are in the checklist
`rm_species_col`	A character string specifying the column name where the information on whether to remove species from the checklist before counting the number of missing species is in the checklist. This can be NULL if no species are to be removed from the checklist. This is useful when species are in the checklist because they are on the island but need to be removed as they are not in the group of interest, e.g. a migratory bird amongst terrestrial birds
`tree_size_range`	Numeric vector of two elements, the first is the smallest tree size (number of tips) and the second is the largest tree size
`num_points`	Numeric determining how many points in the sequence of smallest tree size to largest tree size
`prob_on_island`	Numeric vector of each probability on island to use in the parameter space
`prob_endemic`	Numeric vector of each probability of an island species being endemic to use in the parameter space
`replicates`	Numeric determining the number of replicates to use to account for the stochasticity in sampling the species on the island and endemic species
`log_scale`	A boolean determining whether the sequence of tree sizes are on a linear (FALSE) or log (TRUE) scale
`parameter_index`	Numeric determining which parameter set to use (i.e which row in the parameter space data frame), if this is NULL all parameter sets will be looped over
`sse_model`	either "musse" (default) or "geosse". MuSSE expects state values 1, 2, 3, which here we encode as "not_present", "endemic", "nonendemic", respectively. GeoSSE expects trait values 0, 1, 2, with 0 the widespread state (here, "nonendemic"), and 1 and 2 are "not_present" and "endemic", respectively.
`force_nonendemic_singleton`	A boolean that determines whether all species that are classified as `"nonendemic"` are forced to be extracted as singletons (i.e single species lineages). By default it is `FALSE` so non-endemics can be extracted either as singletons or part of an endemic clade. When set to `TRUE` all non-endemic species in the tree will be single species colonists, with the colonisation time extracted as the stem age for the tip in the phylogeny. There are some exceptions to this, please see `vignette("Forcing_nonendemic_singleton", package = "DAISIEprep")` for more details. This argument is only active when `extraction_method = "asr"`, when `extraction_method = "min"` this argument will be ignored with a warning, as the `min` method always extracts non-endemic species as singletons.

`...`	dots Allows arguments to be passed to `castor::asr_mk_model()` and `castor::asr_max_parsimony()`. These arguments must match by name exactly, see `?castor::asr_mk_model()` and `?castor::asr_max_parsimony()` for information on arguments.

Value

Nothing

Author(s)

Joshua W. Lambert

`endemicity_to_sse_states`

Convert endemicity to SSE states

Description

Convert endemicity to SSE states

Usage

```
endemicity_to_sse_states(endemicity_status, sse_model = "musse")
```

Arguments

`endemicity_status`	character vector with values "endemic", "nonendemic" and/or "not_present"
`sse_model`	either "musse" (default) or "geosse". MuSSE expects state values 1, 2, 3, which here we encode as "not_present", "endemic", "nonendemic", respectively. GeoSSE expects trait values 0, 1, 2, with 0 the widespread state (here, "nonendemic"), and 1 and 2 are "not_present" and "endemic", respectively.

Value

an integer vector of tip states, following the encoding expected by the MuSSE/GeoSSE

`extract_asr_clade` *Extracts an island clade based on the ancestral state reconstruction of the species presence on the island, therefore this clade can contain non-endemic species as well as endemic species.*

Description

Extracts an island clade based on the ancestral state reconstruction of the species presence on the island, therefore this clade can contain non-endemic species as well as endemic species.

Usage

```
extract_asr_clade(phylod, species_label, clade, include_not_present)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`species_label`	The tip label of the species of interest.
`clade`	A numeric vector which the indices of the species which are in the island clade.
`include_not_present`	A boolean determining whether species not present on the island should be included in island colonist when embedded within an island clade. Default is FALSE.

Value

An object of `Island_colonist` class

`extract_biogeobears_ancestral_states_probs` *Extract ancestral state probabilities from BioGeoBEARS output*

Description

Extract the probabilities of each endemicity status for tip and internal node states from the output of an optimisation performed with BioGeoBEARS

Usage

```
extract_biogeobears_ancestral_states_probs(biogeobears_res)
```

Arguments

`biogeobears_res`	a list, the output of `BioGeoBEARS::bears_optim_run()`

Value

a data.frame with one row per node (tips and internals) and four columns: label | not_present | endemic | nonendemic, the last three columns containing the probability of each endemicity status (and summing to 1).

`extract_clade_name` *Creates a name for a clade depending on whether all the species of the clade have the same genus name or whether the clade is composed of multiple genera, in which case it will create a unique clade name by concatinating the genus names*

Description

Creates a name for a clade depending on whether all the species of the clade have the same genus name or whether the clade is composed of multiple genera, in which case it will create a unique clade name by concatinating the genus names

Usage

```
extract_clade_name(clade)
```

Arguments

`clade`	A numeric vector which the indices of the species which are in the island clade.

Value

Character

`extract_endemic_clade` *Extracts the information for an endemic clade (i.e. more than one species on the island more closely related to each other than other mainland species) from a phylogeny (specifically* `phylo4d` *object from* `phylobase` *package) and stores it in an* `Island_colonist` *class*

Description

Extracts the information for an endemic clade (i.e. more than one species on the island more closely related to each other than other mainland species) from a phylogeny (specifically `phylo4d` object from `phylobase` package) and stores it in an `Island_colonist` class

Usage

```
extract_endemic_clade(phylod, species_label, unique_clade_name)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`species_label`	The tip label of the species of interest.
`unique_clade_name`	Boolean determining whether a unique species identifier is used as the clade name in the Island_tbl object or a genus name which may not be unique if that genus has several independent island colonisations

Value

An object of `Island_colonist` class

Examples

```
set.seed(
  3,
  kind = "Mersenne-Twister",
  normal.kind = "Inversion",
  sample.kind = "Rejection"
)
phylo <- ape::rcoal(10)
phylo$tip.label <- c("bird_a", "bird_b", "bird_c", "bird_d", "bird_e",
                     "bird_f", "bird_g", "bird_h", "bird_i", "bird_j")
phylo <- methods::as(phylo, "phylo4")
endemicity_status <- sample(
  x = c("not_present", "endemic", "nonendemic"),
  size = length(phylobase::tipLabels(phylo)),
  replace = TRUE,
  prob = c(0.7, 0.3, 0)
)
phylod <- phylobase::phylo4d(phylo, as.data.frame(endemicity_status))
island_colonist <- extract_endemic_clade(
  phylod = phylod,
  species_label = "bird_i",
  unique_clade_name = TRUE
)
```

`extract_endemic_singleton` *Extracts the information for an endemic species from a phylogeny (specifically* `phylo4d` *object from* `phylobase` *package) and stores it in in an* `Island_colonist` *class*

Description

Extracts the information for an endemic species from a phylogeny (specifically `phylo4d` object from `phylobase` package) and stores it in in an `Island_colonist` class

Usage

```
extract_endemic_singleton(phylod, species_label)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`species_label`	The tip label of the species of interest.

Value

An object of `Island_colonist` class

Examples

```
set.seed(
  1,
  kind = "Mersenne-Twister",
  normal.kind = "Inversion",
  sample.kind = "Rejection"
)
phylo <- ape::rcoal(10)
phylo$tip.label <- c("bird_a", "bird_b", "bird_c", "bird_d", "bird_e",
                     "bird_f", "bird_g", "bird_h", "bird_i", "bird_j")
phylo <- phylobase::phylo4(phylo)
endemicity_status <- sample(
  x = c("not_present", "endemic", "nonendemic"),
  size = length(phylobase::tipLabels(phylo)),
  replace = TRUE,
  prob = c(0.6, 0.2, 0.2)
)
phylod <- phylobase::phylo4d(phylo, as.data.frame(endemicity_status))
extract_endemic_singleton(phylod = phylod, species_label = "bird_i")
```

`extract_island_species` *Extracts the colonisation, diversification, and endemicty data from phylogenetic and endemicity data and stores it in an* `Island_tbl` *object*

Description

Extracts the colonisation, diversification, and endemicty data from phylogenetic and endemicity data and stores it in an `Island_tbl` object

Usage

```
extract_island_species(
  phylod,
  extraction_method,
  island_tbl = NULL,
  include_not_present = FALSE,
  nested_asr_species = c("split", "group"),
  force_nonendemic_singleton = FALSE,
  unique_clade_name = TRUE
)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`extraction_method`	A character string specifying whether the colonisation time extracted is the minimum time (`min`) (before the present), or the most probable time under ancestral state reconstruction (`asr`).
`island_tbl`	An instance of the `Island_tbl` class.
`include_not_present`	A boolean determining whether species not present on the island should be included in island colonist when embedded within an island clade. Default is FALSE.
`nested_asr_species`	A `character` string which determines whether *nested island colonists* are split into separate colonists (`"split"`), or grouped into a single clade (`"group"`). Nested species are those whose tip state is on the island, and they have ancestral nodes on the island, but there are nodes in between these island state nodes that have the state `not_present` (i.e. not on the island). Therefore, the colonisation time can be extracted as the most recent node state on the island (this can be the branching time before the tip if the ancestor node of the tip is not on the island), or the older node state of the larger clade, for `"split"` or `"group"` respectively. **Note** This argument only applies when `extraction_method = "asr"`.
`force_nonendemic_singleton`	A boolean that determines whether all species that are classified as `"nonendemic"` are forced to be extracted as singletons (i.e single species lineages). By default it is `FALSE` so non-endemics can be extracted either as singletons or part of an endemic clade. When set to `TRUE` all non-endemic species in the tree will be single species colonists, with the colonisation time extracted as the stem age for the tip in the phylogeny. There are some exceptions to this, please see `vignette("Forcing_nonendemic_singleton", package = "DAISIEprep")` for more details. This argument is only active when `extraction_method = "asr"`, when `extraction_method = "min"` this argument will be ignored with a warning, as the `min` method always extracts non-endemic species as singletons.

`unique_clade_name`
: Boolean determining whether a unique species identifier is used as the clade name in the Island_tbl object or a genus name which may not be unique if that genus has several independent island colonisations

Value

An object of `Island_tbl` class

Examples

```
set.seed(
  1,
  kind = "Mersenne-Twister",
  normal.kind = "Inversion",
  sample.kind = "Rejection"
)
phylo <- ape::rcoal(10)
phylo$tip.label <- c("bird_a", "bird_b", "bird_c", "bird_d", "bird_e",
                     "bird_f", "bird_g", "bird_h", "bird_i", "bird_j")
phylo <- phylobase::phylo4(phylo)
endemicity_status <- sample(
  c("not_present", "endemic", "nonendemic"),
  size = length(phylobase::tipLabels(phylo)),
  replace = TRUE,
  prob = c(0.6, 0.2, 0.2)
)
phylod <- phylobase::phylo4d(phylo, as.data.frame(endemicity_status))
extract_island_species(phylod, extraction_method = "min")
```

`extract_multi_tip_species` *Extracts the information for a species (endemic or non-endemic) which has multiple tips in the phylogeny (i.e. more than one sample per species) from a phylogeny (specifically* `phylo4d` *object from* `phylobase` *package) and stores it in an* `Island_colonist` *class*

Description

Extracts the information for a species (endemic or non-endemic) which has multiple tips in the phylogeny (i.e. more than one sample per species) from a phylogeny (specifically `phylo4d` object from `phylobase` package) and stores it in an `Island_colonist` class

Usage

```
extract_multi_tip_species(phylod, species_label, species_endemicity)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`species_label`	The tip label of the species of interest.
`species_endemicity`	A character string with the endemicity, either "endemic" or "nonendemic" of an island species, or "not_present" if not on the island.

Value

An object of `Island_colonist` class

Examples

```
set.seed(
  1,
  kind = "Mersenne-Twister",
  normal.kind = "Inversion",
  sample.kind = "Rejection"
)
phylo <- ape::rcoal(10)
phylo$tip.label <- c("bird_a", "bird_b", "bird_c", "bird_d", "bird_e",
                     "bird_f", "bird_g", "bird_h_1", "bird_h_2", "bird_i")
phylo <- phylobase::phylo4(phylo)
endemicity_status <- c("not_present", "not_present", "not_present",
                       "not_present", "not_present", "not_present",
                       "not_present",  "endemic", "endemic", "not_present")
phylod <- phylobase::phylo4d(phylo, as.data.frame(endemicity_status))
extract_multi_tip_species(
  phylod = phylod,
  species_label = "bird_h_1",
  species_endemicity = "endemic"
)
```

`extract_nonendemic` *Extracts the information for a non-endemic species from a phylogeny (specifically* `phylo4d` *object from* `phylobase` *package) and stores it in in an* `island_colonist` *class*

Description

Extracts the information for a non-endemic species from a phylogeny (specifically `phylo4d` object from `phylobase` package) and stores it in in an `island_colonist` class

Usage

```
extract_nonendemic(phylod, species_label)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`species_label`	The tip label of the species of interest.

Value

An object of `island_colonist` class

Examples

```
set.seed(
  1,
  kind = "Mersenne-Twister",
  normal.kind = "Inversion",
  sample.kind = "Rejection"
)
phylo <- ape::rcoal(10)
phylo$tip.label <- c("bird_a", "bird_b", "bird_c", "bird_d", "bird_e",
                     "bird_f", "bird_g", "bird_h", "bird_i", "bird_j")
phylo <- phylobase::phylo4(phylo)
endemicity_status <- sample(
  x = c("not_present", "endemic", "nonendemic"),
  size = length(phylobase::tipLabels(phylo)),
  replace = TRUE,
  prob = c(0.6, 0.2, 0.2)
)
phylod <- phylobase::phylo4d(phylo, as.data.frame(endemicity_status))
extract_nonendemic(phylod = phylod, species_label = "bird_g")
```

`extract_species_asr` *Extracts the colonisation, diversification, and endemicty data from phylogenetic and endemicity data and stores it in an* `Island_tbl` *object using the "asr" algorithm that extract island species given their ancestral states of either island presence or absence.*

Description

Extracts the colonisation, diversification, and endemicty data from phylogenetic and endemicity data and stores it in an `Island_tbl` object using the "asr" algorithm that extract island species given their ancestral states of either island presence or absence.

Usage

```
extract_species_asr(
  phylod,
  species_label,
  species_endemicity,
```

```
  island_tbl,
  include_not_present
)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`species_label`	The tip label of the species of interest.
`species_endemicity`	A character string with the endemicity, either "endemic" or "nonendemic" of an island species, or "not_present" if not on the island.
`island_tbl`	An instance of the `Island_tbl` class.
`include_not_present`	A boolean determining whether species not present on the island should be included in island colonist when embedded within an island clade. Default is FALSE.

Value

An object of `island_tbl` class

Examples

```
set.seed(
  1,
  kind = "Mersenne-Twister",
  normal.kind = "Inversion",
  sample.kind = "Rejection"
)
phylo <- ape::rcoal(10)
phylo$tip.label <- c("bird_a", "bird_b", "bird_c", "bird_d", "bird_e",
                     "bird_f", "bird_g", "bird_h", "bird_i", "bird_j")
phylo <- phylobase::phylo4(phylo)
endemicity_status <- sample(c("not_present", "endemic", "nonendemic"),
                            size = length(phylobase::tipLabels(phylo)),
                            replace = TRUE, prob = c(0.8, 0.1, 0.1))
phylod <- phylobase::phylo4d(phylo, as.data.frame(endemicity_status))
phylod <- add_asr_node_states(
  phylod = phylod,
  asr_method = "parsimony"
)
island_tbl <- island_tbl()
extract_species_asr(
  phylod = phylod,
  species_label = "bird_i",
  species_endemicity = "endemic",
  island_tbl = island_tbl,
  include_not_present = FALSE
)
```

`extract_species_min` *Extracts the colonisation, diversification, and endemicty data from phylogenetic and endemicity data and stores it in an* `Island_tbl` *object using the "min" algorithm that extract island species as the shortest time to the present.*

Description

Extracts the colonisation, diversification, and endemicty data from phylogenetic and endemicity data and stores it in an `Island_tbl` object using the "min" algorithm that extract island species as the shortest time to the present.

Usage

```
extract_species_min(
  phylod,
  species_label,
  species_endemicity,
  island_tbl,
  unique_clade_name
)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`species_label`	The tip label of the species of interest.
`species_endemicity`	A character string with the endemicity, either "endemic" or "nonendemic" of an island species, or "not_present" if not on the island.
`island_tbl`	An instance of the `Island_tbl` class.
`unique_clade_name`	Boolean determining whether a unique species identifier is used as the clade name in the Island_tbl object or a genus name which may not be unique if that genus has several independent island colonisations

Value

An object of `island_tbl` class

Examples

```
set.seed(
  1,
  kind = "Mersenne-Twister",
  normal.kind = "Inversion",
  sample.kind = "Rejection"
```

```
)
phylo <- ape::rcoal(10)
phylo$tip.label <- c("bird_a", "bird_b", "bird_c", "bird_d", "bird_e",
                     "bird_f", "bird_g", "bird_h", "bird_i", "bird_j")
phylo <- phylobase::phylo4(phylo)
endemicity_status <- sample(
  c("not_present", "endemic", "nonendemic"),
  size = length(phylobase::tipLabels(phylo)),
  replace = TRUE,
  prob = c(0.6, 0.2, 0.2)
)
phylod <- phylobase::phylo4d(phylo, as.data.frame(endemicity_status))
island_tbl <- island_tbl()
extract_species_min(
  phylod = phylod,
  species_label = "bird_g",
  species_endemicity = "nonendemic",
  island_tbl = island_tbl,
  unique_clade_name = TRUE
)
```

`extract_stem_age` *Extracts the stem age from the phylogeny when the a species is known to belong to a genus but is not itself in the phylogeny and there are members of the same genus are in the phylogeny. The stem age can either be for the genus (or several genera) in the tree (`stem = "genus"`) or use an extraction algorithm to find the stem of when the species colonised the island (`stem = "island_presence`), either 'min' or 'asr' as in extract_island_species(). When `stem = "island_presence"` the reconstructed node states are used to determine the stem age.*

Description

Extracts the stem age from the phylogeny when the a species is known to belong to a genus but is not itself in the phylogeny and there are members of the same genus are in the phylogeny. The stem age can either be for the genus (or several genera) in the tree (`stem = "genus"`) or use an extraction algorithm to find the stem of when the species colonised the island (`stem = "island_presence`), either 'min' or 'asr' as in extract_island_species(). When `stem = "island_presence"` the reconstructed node states are used to determine the stem age.

Usage

```
extract_stem_age(genus_name, phylod, stem, extraction_method = NULL)
```

Arguments

`genus_name`	Character string of genus name to be matched with a genus name from the tip labels in the phylogeny

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`stem`	Character string, either "genus" or "island_presence". The former will extract the stem age of the genussbased on the genus name provided, the latter will extract the stem age based on the ancestral presence on the island either based on the "min" or "asr" extraction algorithms.
`extraction_method`	A character string specifying whether the colonisation time extracted is the minimum time (`min`) (before the present), or the most probable time under ancestral state reconstruction (`asr`).

Value

Numeric

Examples

```
# In this example the parrot clade is the genus of interest only the parrots
# are endemic to the island and all the passerines are not on the island
set.seed(1)
tree <- ape::rcoal(10)
tree$tip.label <- c(
  "passerine_a", "passerine_b", "passerine_c", "passerine_d", "passerine_e",
  "passerine_f", "parrot_a", "parrot_b", "parrot_c", "passerine_j")
tree <- phylobase::phylo4(tree)
endemicity_status <- c(
  "not_present", "not_present", "not_present", "not_present", "not_present",
  "not_present", "endemic", "endemic", "endemic", "not_present")
phylod <- phylobase::phylo4d(tree, as.data.frame(endemicity_status))
DAISIEprep::plot_phylod(phylod)
# the species 'parrot_a' is removed and becomes the missing species we want
# to the know the stem age for
phylod <- phylobase::subset(x = phylod, tips.exclude = "parrot_a")
DAISIEprep::plot_phylod(phylod)
extract_stem_age(
  genus_name = "parrot",
  phylod = phylod,
  stem = "island_presence",
  extraction_method = "min"
)
# here we use the extraction_method = "asr" which requires ancestral node
# states in the tree.
phylod <- add_asr_node_states(
  phylod = phylod,
  asr_method = "parsimony",
  tie_preference = "mainland"
)
DAISIEprep::plot_phylod(phylod)
extract_stem_age(
  genus_name = "parrot",
  phylod = phylod,
```

```
  stem = "island_presence",
  extraction_method = "asr"
)
# lastly we extract the stem age based on the genus name
extract_stem_age(
  genus_name = "parrot",
  phylod = phylod,
  stem = "genus",
  extraction_method = NULL
)
```

`extract_stem_age_asr` *Extracts the stem age from the phylogeny when the a species is known to belong to a genus but is not itself in the phylogeny and there are members of the same genus are in the phylogeny using the 'asr' extraction method*

Description

Extracts the stem age from the phylogeny when the a species is known to belong to a genus but is not itself in the phylogeny and there are members of the same genus are in the phylogeny using the 'asr' extraction method

Usage

```
extract_stem_age_asr(genus_in_tree, phylod)
```

Arguments

`genus_in_tree`	A numeric vector that indicates which species in the genus are in the tree
`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.

Value

Numeric

`extract_stem_age_genus` *Extracts the stem age from the phylogeny when the a species is known to belong to a genus but is not itself in the phylogeny and there are members of the same genus are in the phylogeny*

Description

Extracts the stem age from the phylogeny when the a species is known to belong to a genus but is not itself in the phylogeny and there are members of the same genus are in the phylogeny

Usage

```
extract_stem_age_genus(genus_in_tree, phylod)
```

Arguments

`genus_in_tree`	A numeric vector that indicates which species in the genus are in the tree
`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.

Value

Numeric

`extract_stem_age_min` *Extracts the stem age from the phylogeny when the a species is known to belong to a genus but is not itself in the phylogeny and there are members of the same genus are in the phylogeny using the 'min' extraction method*

Description

Extracts the stem age from the phylogeny when the a species is known to belong to a genus but is not itself in the phylogeny and there are members of the same genus are in the phylogeny using the 'min' extraction method

Usage

```
extract_stem_age_min(genus_in_tree, phylod)
```

Arguments

`genus_in_tree`	A numeric vector that indicates which species in the genus are in the tree
`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.

Value

Numeric

`finches_phylod` *A phylogenetic tree of finches species with endemicity status as tip states.*

Description

A phylogenetic tree of finches species with endemicity status as tip states.

Usage

```
finches_phylod
```

Format

`finches_phylod`**:**
A `phylo4d` object (from the **phylobase** package) with 16 tips and 15 internal nodes.

Source

Valente, L.M., A.B. Phillimore and R.S. Etienne (2015). Equilibrium and non-equilibrium dynamics simultaneously operate in the Galapagos islands. Ecology Letters 18: 844-852.

`GalapagosTrees` *Phylogenetic trees of the Galapagos bird lineages and sister species on the mainland.*

Description

Each dataset is a phylogenetic tree used to extract the Galapagos avifaunal lineages.

Usage

```
coccyzus_tree

columbiformes_tree

finches_tree

mimus_tree

myiarchus_tree

progne_tree

pyrocephalus_tree

setophaga_tree
```

Format

A `phylo` object (from the **ape** package)

An object of class `phylo` of length 5.

An object of class `phylo` of length 4.

An object of class `phylo` of length 5.

An object of class `phylo` of length 4.

An object of class `phylo` of length 5.

An object of class `phylo` of length 5.

An object of class `phylo` of length 5.

Source

Valente, L.M., A.B. Phillimore and R.S. Etienne (2015). Equilibrium and non-equilibrium dynamics simultaneously operate in the Galapagos islands. Ecology Letters 18: 844-852.

`get_clade_name` *Accessor functions for the data (slots) in objects of the `Island_colonist` class*

Description

Accessor functions for the data (slots) in objects of the `Island_colonist` class

Usage

```
get_clade_name(x)

## S4 method for signature 'Island_colonist'
get_clade_name(x)

set_clade_name(x) <- value

## S4 replacement method for signature 'Island_colonist'
set_clade_name(x) <- value

get_status(x)

## S4 method for signature 'Island_colonist'
get_status(x)

set_status(x) <- value

## S4 replacement method for signature 'Island_colonist'
set_status(x) <- value
```

```
get_missing_species(x)

## S4 method for signature 'Island_colonist'
get_missing_species(x)

set_missing_species(x) <- value

## S4 replacement method for signature 'Island_colonist'
set_missing_species(x) <- value

get_col_time(x)

## S4 method for signature 'Island_colonist'
get_col_time(x)

set_col_time(x) <- value

## S4 replacement method for signature 'Island_colonist'
set_col_time(x) <- value

get_col_max_age(x)

## S4 method for signature 'Island_colonist'
get_col_max_age(x)

set_col_max_age(x) <- value

## S4 replacement method for signature 'Island_colonist'
set_col_max_age(x) <- value

get_branching_times(x)

## S4 method for signature 'Island_colonist'
get_branching_times(x)

set_branching_times(x) <- value

## S4 replacement method for signature 'Island_colonist'
set_branching_times(x) <- value

get_min_age(x)

## S4 method for signature 'Island_colonist'
get_min_age(x)

set_min_age(x) <- value
```

```
## S4 replacement method for signature 'Island_colonist'
set_min_age(x) <- value

get_species(x)

## S4 method for signature 'Island_colonist'
get_species(x)

set_species(x) <- value

## S4 replacement method for signature 'Island_colonist'
set_species(x) <- value

get_clade_type(x)

## S4 method for signature 'Island_colonist'
get_clade_type(x)

set_clade_type(x) <- value

## S4 replacement method for signature 'Island_colonist'
set_clade_type(x) <- value
```

Arguments

`x`	An object whose class is determined by the signature.
`value`	A value which can take several forms to be assigned to an object of a class.

Value

Getter functions (get_*) return a variable from the Island_colonist class, the setter functions (set_*) return the modified Island_colonist class.

Author(s)

Joshua W. Lambert

Examples

```
colonist <- island_colonist()
get_clade_name(colonist)
set_clade_name(colonist) <- "abc"
get_status(colonist)
set_status(colonist) <- "abc"
get_missing_species(colonist)
set_missing_species(colonist) <- 0
get_col_time(colonist)
set_col_time(colonist) <- 1
get_col_max_age(colonist)
set_col_max_age(colonist) <- FALSE
```

```
get_branching_times(colonist)
set_branching_times(colonist) <- 0
get_min_age(colonist)
set_min_age(colonist) <- 0.1
get_species(colonist)
set_species(colonist) <- "abc_a"
get_clade_type(colonist)
set_clade_type(colonist) <- 1
```

`get_island_tbl` *Accessor functions for the data (slots) in objects of the* `Island_tbl` *class*

Description

Accessor functions for the data (slots) in objects of the `Island_tbl` class

Usage

```
get_island_tbl(x)

## S4 method for signature 'Island_tbl'
get_island_tbl(x)

set_island_tbl(x) <- value

## S4 replacement method for signature 'Island_tbl'
set_island_tbl(x) <- value

get_extracted_species(x)

## S4 method for signature 'Island_tbl'
get_extracted_species(x)

set_extracted_species(x) <- value

## S4 replacement method for signature 'Island_tbl'
set_extracted_species(x) <- value

get_num_phylo_used(x)

## S4 method for signature 'Island_tbl'
get_num_phylo_used(x)

set_num_phylo_used(x) <- value

## S4 replacement method for signature 'Island_tbl'
set_num_phylo_used(x) <- value
```

Arguments

`x`	An object whose class is determined by the signature.
`value`	A value which can take several forms to be assigned to an object of a class.

Value

Getter function (get_*) returns a data frame, the setter function (set_*) returns the modified Island_tbl class.

Author(s)

Joshua W. Lambert

Examples

```
island_tbl <- island_tbl()
get_island_tbl(island_tbl)
set_island_tbl(island_tbl) <- data.frame(
  clade_name = "birds",
  status = "endemic",
  missing_species = 0,
  branching_times = I(list(c(1.0, 0.5)))
)
```

`get_sse_tip_states` *Extract tip states from a phylod object*

Description

Extract tip states from a phylod object

Usage

```
get_sse_tip_states(phylod, sse_model = "musse")
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`sse_model`	either "musse" (default) or "geosse". MuSSE expects state values 1, 2, 3, which here we encode as "not_present", "endemic", "nonendemic", respectively. GeoSSE expects trait values 0, 1, 2, with 0 the widespread state (here, "nonendemic"), and 1 and 2 are "not_present" and "endemic", respectively.

Value

an integer vector of tip states, as expected by SSE models

`island_colonist` *Constructor for Island_colonist*

Description

Constructor for Island_colonist

Usage

```
island_colonist(
  clade_name = NA_character_,
  status = NA_character_,
  missing_species = NA_real_,
  col_time = NA_real_,
  col_max_age = NA,
  branching_times = NA_real_,
  min_age = NA_real_,
  species = NA_character_,
  clade_type = NA_integer_
)
```

Arguments

`clade_name`	Character name of the colonising clade.
`status`	Character endemicity status of the colonising clade. Either `"endemic"` or `"nonendemic"`.
`missing_species`	Numeric number of missing species from the phylogeny that belong to the colonising clade. For a clade with missing species this is $n - 1$, where n is the number of missing species in the clade. If the clade is an island singleton, the number of missing species is `0` because by adding the colonist it already counts as one automatically. If the clade has more than one species, the `missing_species` is $n - 1$ because adding the lineage already counts as one.
`col_time`	Numeric with the colonisation time of the island colonist
`col_max_age`	Boolean determining whether colonisation time should be considered a precise time of colonisation or a maximum time of colonisation
`branching_times`	Numeric vector of one or more elements which are the branching times on the island.
`min_age`	Numeric minimum age (time before the present) that the species must have colonised the island by. This is known when there is a branching on the island, either in species or subspecies.
`species`	Character vector of one or more elements containing the name of the species included in the colonising clade.

`clade_type`	Numeric determining which type of clade the island colonist is, this determines which macroevolutionary regime (parameter set) the island colonist is in. After formatting the `island_tbl` to a DAISIE data list, the clade type can be used to conduct a 2-type analysis (see `https://CRAN.R-project.org/package=DAISIE/vignettes/demo_optimize.html` for more information)

Value

Object of `Island_colonist` class.

Examples

```
# Without initial values
colonist <- island_colonist()

# With initial values
colonist <- island_colonist(
  clade_name = "bird",
  status = "endemic",
  missing_species = 0,
  col_time = 0.5,
  col_max_age = FALSE,
  branching_times = 0.5,
  min_age = NA_real_,
  species = "bird_a",
  clade_type = 1
)
```

`Island_colonist-class` *Defines the* `island_tbl` *class which is used when extracting information from the phylogenetic and island data to be used for constructing a* `daisie_data_tbl`

Description

Defines the `island_tbl` class which is used when extracting information from the phylogenetic and island data to be used for constructing a `daisie_data_tbl`

Slots

`clade_name` character.

`status` character.

`missing_species` character.

`col_time` numeric.

`col_max_age` logical.

`branching_times` numeric.

`min_age` numeric.

`species` character.

`clade_type` numeric.

`island_tbl` *Constructor function for* `Island_tbl` *class*

Description

Constructor function for `Island_tbl` class

Usage

```
island_tbl()
```

Value

An Island_tbl object.

`Island_tbl-class` *Defines the* `island_tbl` *class which is used when extracting information from the phylogenetic and island data to be used for constructing a* `daisie_data_tbl`

Description

Defines the `island_tbl` class which is used when extracting information from the phylogenetic and island data to be used for constructing a `daisie_data_tbl`

Slots

`island_tbl` data frame.

`metadata` list.

`is_back_colonisation` *Checks whether species has undergone back-colonisation from*

Description

Checks whether species has undergone back-colonisation from

Usage

```
is_back_colonisation(phylod, node_label)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`node_label`	A numeric label for a node within a phylogeny.

Value

A character string or FALSE. Character string is in the format ancestral_node -> focal_node, where the ancestral node is not on mainland but the focal node is.

Examples

```
set.seed(
3,
kind = "Mersenne-Twister",
normal.kind = "Inversion",
sample.kind = "Rejection"
)
phylo <- ape::rcoal(5)
phylo$tip.label <- c("bird_a", "bird_b", "bird_c", "bird_d", "bird_e")
phylo <- phylobase::phylo4(phylo)
endemicity_status <- c("endemic", "endemic", "not_present",
                       "endemic", "not_present")
phylod <- phylobase::phylo4d(phylo, as.data.frame(endemicity_status))
phylod <- add_asr_node_states(phylod = phylod, asr_method = "parsimony")
# aritificially modify data to produce back-colonisation
phylobase::tdata(phylod)$island_status[8] <- "endemic"
# Example without back colonisation
is_back_colonisation(phylod = phylod, node_label = 2)
# Example with back colonisation
is_back_colonisation(phylod = phylod, node_label = 3)
```

`is_duplicate_colonist` *Determines if colonist has already been stored in* `Island_tbl` *class. This is used to stop endemic clades from being stored multiple times in the island table by checking if the endemicity status and branching times are identical.*

Description

Determines if colonist has already been stored in `Island_tbl` class. This is used to stop endemic clades from being stored multiple times in the island table by checking if the endemicity status and branching times are identical.

Usage

```
is_duplicate_colonist(island_colonist, island_tbl)
```

Arguments

`island_colonist`	An instance of the `Island_colonist` class.
`island_tbl`	An instance of the `Island_tbl` class.

Value

Boolean

Examples

```
# with empty island_tbl
island_colonist <- island_colonist(
  clade_name = "bird",
  status = "endemic",
  missing_species = 0,
  col_time = 1.0,
  col_max_age = FALSE,
  branching_times = 0.5,
  species = "bird_a",
  clade_type = 1
)
island_tbl <- island_tbl()
is_duplicate_colonist(
  island_colonist = island_colonist,
  island_tbl = island_tbl
)

# with non-empty island_tbl
island_colonist <- island_colonist(
  clade_name = "bird",
  status = "endemic",
```

```
  missing_species = 0,
  col_time = 1.0,
  col_max_age = FALSE,
  branching_times = 0.5,
  species = c("bird_a", "bird_b"),
  clade_type = 1
)
island_tbl <- island_tbl()
island_tbl <- bind_colonist_to_tbl(
  island_colonist = island_colonist,
  island_tbl = island_tbl
)
island_colonist <- island_colonist(
  clade_name = "bird",
  status = "endemic",
  missing_species = 0,
  col_time = 1.0,
  col_max_age = FALSE,
  branching_times = 0.5,
  species = c("bird_a", "bird_b"),
  clade_type = 1
)
is_duplicate_colonist(
  island_colonist = island_colonist,
  island_tbl = island_tbl
)
```

`is_identical_island_tbl`

Checks whether two `Island_tbl` objects are identical. If they are different comparisons are made to report which components of the `Island_tbls` are different.

Description

Checks whether two `Island_tbl` objects are identical. If they are different comparisons are made to report which components of the `Island_tbls` are different.

Usage

```
is_identical_island_tbl(island_tbl_1, island_tbl_2)
```

Arguments

`island_tbl_1`	An object of `Island_tbl` class to be comparedl
`island_tbl_2`	An object of `Island_tbl` class to be compared

Value

Either TRUE or a character string with the differences

Examples

```
multi_island_tbl <- multi_extract_island_species(
  multi_phylod = list(
    create_test_phylod(test_scenario = 1),
    create_test_phylod(test_scenario = 1)),
 extraction_method = "min")
is_identical_island_tbl(multi_island_tbl[[1]], multi_island_tbl[[2]])
```

`mimus_phylod` *A phylogenetic tree of mimus species with endemicity status as tip states.*

Description

A phylogenetic tree of mimus species with endemicity status as tip states.

Usage

```
mimus_phylod
```

Format

`mimus_phylod`**:**

A `phylo4d` object (from the **phylobase** package) with 29 tips and 28 internal nodes.

Source

Valente, L.M., A.B. Phillimore and R.S. Etienne (2015). Equilibrium and non-equilibrium dynamics simultaneously operate in the Galapagos islands. Ecology Letters 18: 844-852.

`multi_extract_island_species` *Extracts the colonisation, diversification, and endemicty data from multiple* `phylod` *(*`phylo4d` *class from* `phylobase`*) objects (composed of phylogenetic and endemicity data) and stores each in an* `Island_tbl` *object which are stored in a* `Multi_island_tbl` *object.*

Description

Extracts the colonisation, diversification, and endemicty data from multiple `phylod` (`phylo4d` class from `phylobase`) objects (composed of phylogenetic and endemicity data) and stores each in an `Island_tbl` object which are stored in a `Multi_island_tbl` object.

Usage

```
multi_extract_island_species(
  multi_phylod,
  extraction_method,
  island_tbl = NULL,
  include_not_present = FALSE,
  verbose = FALSE,
  unique_clade_name = TRUE
)
```

Arguments

`multi_phylod`	A list of phylod objects.
`extraction_method`	A character string specifying whether the colonisation time extracted is the minimum time (`min`) (before the present), or the most probable time under ancestral state reconstruction (`asr`).
`island_tbl`	An instance of the `Island_tbl` class.
`include_not_present`	A boolean determining whether species not present on the island should be included in island colonist when embedded within an island clade. Default is FALSE.
`verbose`	Boolean. States if intermediate results should be printed to console. Defaults to FALSE
`unique_clade_name`	Boolean determining whether a unique species identifier is used as the clade name in the Island_tbl object or a genus name which may not be unique if that genus has several independent island colonisations

Value

An object of `Multi_island_tbl` class

Examples

```
multi_phylod <- list()
multi_phylod[[1]] <- create_test_phylod(test_scenario = 1)
multi_phylod[[2]] <- create_test_phylod(test_scenario = 2)
multi_island_tbl <- multi_extract_island_species(
  multi_phylod = multi_phylod,
  extraction_method = "min",
  island_tbl = NULL,
  include_not_present = FALSE
)
```

`multi_island_tbl` *Constructor function for* `Multi_island_tbl` *class*

Description

Constructor function for `Multi_island_tbl` class

Usage

```
multi_island_tbl()
```

Value

A Multi_island_tbl object.

`Multi_island_tbl-class` *Defines the* `Multi_island_tbl` *class which is multiple* `Island_tbl`*s.*

Description

Defines the `Multi_island_tbl` class which is multiple `Island_tbl`s.

Slots

`.Data` a list of `Island_tbl`.

`myiarchus_phylod` *A phylogenetic tree of myiarchus species with endemicity status as tip states.*

Description

A phylogenetic tree of myiarchus species with endemicity status as tip states.

Usage

```
myiarchus_phylod
```

Format

`myiarchus_phylod`**:**
A `phylo4d` object (from the **phylobase** package) with 13 tips and 12 internal nodes.

Source

Valente, L.M., A.B. Phillimore and R.S. Etienne (2015). Equilibrium and non-equilibrium dynamics simultaneously operate in the Galapagos islands. Ecology Letters 18: 844-852.

`plant_phylo` *A phylogenetic tree of plant species.*

Description

This is a mock dataset for demonstrating the **DAISIEprep** package.

Usage

```
plant_phylo
```

Format

`plant_phylo`**:**

A `phylo` object (from the **ape** package) with 10 tips and 9 internal nodes. Each tip label is formatted `"Plant_*"` with the letters `a`, `b`, etc. used as species names.

`plot_colonisation` *Plots a dot plot (cleveland dot plot when include_crown_age = TRUE) of the stem and potentially crown ages of a community of island colonists.*

Description

Plots a dot plot (cleveland dot plot when include_crown_age = TRUE) of the stem and potentially crown ages of a community of island colonists.

Usage

```
plot_colonisation(island_tbl, island_age, include_crown_age = TRUE)
```

Arguments

`island_tbl`	An instance of the `Island_tbl` class.
`island_age`	Age of the island in appropriate units.
`include_crown_age`	A boolean determining whether the crown age gets plotted with the stem age.

Value

`ggplot` object

Examples

```
set.seed(
  1,
  kind = "Mersenne-Twister",
  normal.kind = "Inversion",
  sample.kind = "Rejection"
)
phylo <- ape::rcoal(10)
phylo$tip.label <- c("bird_a", "bird_b", "bird_c", "bird_d", "bird_e",
                     "bird_f", "bird_g", "bird_h", "bird_i", "bird_j")
phylo <- phylobase::phylo4(phylo)
endemicity_status <- sample(
  c("not_present", "endemic", "nonendemic"),
  size = length(phylobase::tipLabels(phylo)),
  replace = TRUE,
  prob = c(0.6, 0.2, 0.2)
)
phylod <- phylobase::phylo4d(phylo, as.data.frame(endemicity_status))
island_tbl <- extract_island_species(phylod, extraction_method = "min")
plot_colonisation(island_tbl, island_age = 2)
```

`plot_performance` *Plots performance results for a grouping variable (prob_on_island or prob_endemic).*

Description

Plots performance results for a grouping variable (prob_on_island or prob_endemic).

Usage

```
plot_performance(performance_data, group_by)
```

Arguments

`performance_data`	Tibble of collated performance results
`group_by`	A variable to partition by for plotting. Uses data masking so variable does not need to be quoted.

Value

ggplot2 object

`plot_phylod` *Plots the phylogenetic tree and its associated tip and/or node data*

Description

Plots the phylogenetic tree and its associated tip and/or node data

Usage

```
plot_phylod(phylod, node_pies = FALSE)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`node_pies`	Boolean determining if pie charts of the probabilities of a species being present on the island. If TRUE the correct data is required in the phylod object.

Value

`ggplot` object

Examples

```
set.seed(
  1,
  kind = "Mersenne-Twister",
  normal.kind = "Inversion",
  sample.kind = "Rejection"
)
phylo <- ape::rcoal(10)
phylo$tip.label <- c("bird_a", "bird_b", "bird_c", "bird_d", "bird_e",
                     "bird_f", "bird_g", "bird_h", "bird_i", "bird_j")
phylo <- phylobase::phylo4(phylo)
endemicity_status <- sample(
  c("not_present", "endemic", "nonendemic"),
  size = length(phylobase::tipLabels(phylo)),
  replace = TRUE,
  prob = c(0.6, 0.2, 0.2)
)
phylod <- phylobase::phylo4d(phylo, as.data.frame(endemicity_status))
plot_phylod(phylod)
```

`progne_phylod` *A phylogenetic tree of progne species with endemicity status as tip states.*

Description

A phylogenetic tree of progne species with endemicity status as tip states.

Usage

```
progne_phylod
```

Format

`progne_phylod`**:**

A `phylo4d` object (from the **phylobase** package) with 19 tips and 18 internal nodes.

Source

Valente, L.M., A.B. Phillimore and R.S. Etienne (2015). Equilibrium and non-equilibrium dynamics simultaneously operate in the Galapagos islands. Ecology Letters 18: 844-852.

`pyrocephalus_phylod` *A phylogenetic tree of pyrocephalus species with endemicity status as tip states.*

Description

A phylogenetic tree of pyrocephalus species with endemicity status as tip states.

Usage

```
pyrocephalus_phylod
```

Format

`pyrocephalus_phylod`**:**

A `phylo4d` object (from the **phylobase** package) with 66 tips and 65 internal nodes.

Source

Valente, L.M., A.B. Phillimore and R.S. Etienne (2015). Equilibrium and non-equilibrium dynamics simultaneously operate in the Galapagos islands. Ecology Letters 18: 844-852.

`rm_island_colonist` *Removes an island colonist from an* `Island_tbl` *object*

Description

Removes an island colonist from an `Island_tbl` object

Usage

```
rm_island_colonist(island_tbl, clade_name)
```

Arguments

`island_tbl`	An instance of the `Island_tbl` class.
`clade_name`	Character name of the colonising clade.

Value

Object of `Island_tbl` class

Examples

```
phylod <- create_test_phylod(test_scenario = 1)
island_tbl <- extract_island_species(
  phylod = phylod,
  extraction_method = "min"
)
island_tbl <- rm_island_colonist(
  island_tbl = island_tbl,
  clade_name = "bird_b"
)
```

`rm_multi_missing_species` *Loops through the genera that have missing species and removes the ones that are found in the missing genus list which have phylogenetic data. This is useful when wanting to know which missing species have not been assigned to the island_tbl using* `add_multi_missing_species()`.

Description

Loops through the genera that have missing species and removes the ones that are found in the missing genus list which have phylogenetic data. This is useful when wanting to know which missing species have not been assigned to the island_tbl using `add_multi_missing_species()`.

Usage

```
rm_multi_missing_species(missing_species, missing_genus, island_tbl)
```

Arguments

`missing_species`	Numeric number of missing species from the phylogeny that belong to the colonising clade. For a clade with missing species this is $n - 1$, where n is the number of missing species in the clade. If the clade is an island singleton, the number of missing species is `0` because by adding the colonist it already counts as one automatically. If the clade has more than one species, the `missing_species` is $n - 1$ because adding the lineage already counts as one.
`missing_genus`	A list of character vectors containing the genera in each island clade
`island_tbl`	An instance of the `Island_tbl` class.

Value

Data frame

Examples

```
phylod <- create_test_phylod(test_scenario = 6)
island_tbl <- suppressWarnings(extract_island_species(
 phylod = phylod,
 extraction_method = "asr",
))
phylod <- create_test_phylod(test_scenario = 7)
island_tbl <- suppressWarnings(extract_island_species(
 phylod = phylod,
 extraction_method = "asr",
 island_tbl = island_tbl
))
missing_species <- data.frame(
  clade_name = "bird",
  missing_species = 1,
  endemicity_status = "endemic"
)
missing_genus <- list("bird", character(0))
rm_missing_species <- rm_multi_missing_species(
  missing_species = missing_species,
  missing_genus = missing_genus,
  island_tbl = island_tbl
)
```

`round_up` *Rounds numbers using the round up method, rather than the round to the nearest even number method used by the base function* `round`*.*

Description

Rounds numbers using the round up method, rather than the round to the nearest even number method used by the base function `round`.

Usage

```
round_up(n, digits = 0)
```

Arguments

`n`	A numeric to be rounded.
`digits`	A numeric specifying which decimal places to round to

Value

Numeric

`select_endemicity_status` *Select endemicity status from ancestral states probabilities*

Description

Selects a state for each node (both internal nodes, i.e. ancestral states, and tips, if included) from a table of probabilities.

Usage

```
select_endemicity_status(asr_df, method = "max")
```

Arguments

`asr_df`	a data frame containing at least these three columns: not_present_prob | endemic_prob | nonendemic_prob (in any order). Each column should contain the estimated probability of the state for each node (rows) and these columns should sum to 1.
`method`	"max" or "random". "max" will select the state with highest probability (selecting last state in event of a tie), while "random" will sample the states randomly with the probabilities as weight for each state.

Value

a character vector, with the selected endemicity status for each node.

`sensitivity` *Runs a sensitivity analysis to test the influences of changing the data on the parameter estimates for the DAISIE maximum likelihood inference model*

Description

Runs a sensitivity analysis to test the influences of changing the data on the parameter estimates for the DAISIE maximum likelihood inference model

Usage

```
sensitivity(
  phylo,
  island_species,
  extraction_method,
  asr_method,
  tie_preference,
  island_age,
  num_mainland_species,
  verbose = FALSE
)
```

Arguments

`phylo`	A phylogeny either as a `phylo` (from the `ape` package) or `phylo4` (from the `phylobase` package) object.
`island_species`	Data frame with two columns. The first is a character string of the tip_labels with the tip names of the species on the island. The second column a character string of the endemicity status of the species, either endemic or nonendemic.
`extraction_method`	A character string specifying whether the colonisation time extracted is the minimum time (`min`) (before the present), or the most probable time under ancestral state reconstruction (`asr`).
`asr_method`	A character string, either "parsimony" or "mk" determines whether a maximum parsimony or continuous-time markov model reconstructs the ancestral states at each node. See documentation in `castor::asr_max_parsimony()` or `castor::asr_mk_model()` in `castor` R package for details on the methods used.
`tie_preference`	Character string, either "island" or "mainland" to choose the most probable state at each node using the `max.col()` function. When a node has island presence and absence equally probable we need to decide whether that species should be considered on the island. To consider it on the island use `ties.method`

	= `"last"` in the `max.col()` function, if you consider it not on the island use `ties.method = "first"`. Default is "island".
`island_age`	Age of the island in appropriate units.
`num_mainland_species`	The size of the mainland pool, i.e. the number of species that can potentially colonise the island.
`verbose`	Boolean. States if intermediate results should be printed to console. Defaults to FALSE

Value

Data frame of parameter estimates and the parameter setting used when inferring them

---

`setophaga_phylod` *A phylogenetic tree of setophaga species with endemicity status as tip states.*

Description

A phylogenetic tree of setophaga species with endemicity status as tip states.

Usage

```
setophaga_phylod
```

Format

`setophaga_phylod`**:**

A `phylo4d` object (from the **phylobase** package) with 19 tips and 18 internal nodes.

Source

Valente, L.M., A.B. Phillimore and R.S. Etienne (2015). Equilibrium and non-equilibrium dynamics simultaneously operate in the Galapagos islands. Ecology Letters 18: 844-852.

`sse_states_to_endemicity` *Convert SSE states back to endemicity status*

Description

Convert SSE states back to endemicity status

Usage

```
sse_states_to_endemicity(states, sse_model = "musse")
```

Arguments

`states`	integer vector of tip states, as expected by SSE models
`sse_model`	either "musse" (default) or "geosse". MuSSE expects state values 1, 2, 3, which here we encode as "not_present", "endemic", "nonendemic", respectively. GeoSSE expects trait values 0, 1, 2, with 0 the widespread state (here, "nonendemic"), and 1 and 2 are "not_present" and "endemic", respectively.

Value

character vector with values "endemic", "nonendemic" and/or "not_present"

`translate_status` *Takes a string of the various ways the island species status can be and returns a uniform all lower-case string of the same status to make handling statuses easier in other function*

Description

Takes a string of the various ways the island species status can be and returns a uniform all lower-case string of the same status to make handling statuses easier in other function

Usage

```
translate_status(status)
```

Arguments

`status`	Character endemicity status of the colonising clade. Either `"endemic"` or `"nonendemic"`.

Value

Character string

Examples

```
translate_status("Endemic")
```

`unique_island_genera` *Determines the unique endemic genera that are included in the island clades contained within the island_tbl object and stores them as a list with each genus only occuring once in the first island clade it appears in*

Description

Determines the unique endemic genera that are included in the island clades contained within the island_tbl object and stores them as a list with each genus only occuring once in the first island clade it appears in

Usage

```
unique_island_genera(island_tbl)
```

Arguments

`island_tbl`	An instance of the `Island_tbl` class.

Value

list of character vectors

Examples

```
phylod <- create_test_phylod(test_scenario = 6)
island_tbl <- suppressWarnings(extract_island_species(
  phylod = phylod,
  extraction_method = "asr",
))
phylod <- create_test_phylod(test_scenario = 7)
island_tbl <- suppressWarnings(extract_island_species(
  phylod = phylod,
  extraction_method = "asr",
  island_tbl = island_tbl
))
unique_genera <- unique_island_genera(island_tbl = island_tbl)
```

`write_biogeobears_input` *Write input files for BioGeoBEARS*

Description

Write input files for a BioGeoBEARS analysis, i.e. a phlyogenetic tree in Newick format and occurrence data in PHYLIP format.

Usage

```
write_biogeobears_input(phylod, path_to_phylo, path_to_biogeo)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`path_to_phylo`	string specifying the path and name to write the phylogeny file to.
`path_to_biogeo`	string specifying the path and name to write the biogeography file to.

Value

Nothing, called for side-effects

`write_newick_file` *Write tree input file for BioGeoBEARS*

Description

Write a text file containing a phylogenetic tree in the Newick format expected by BioGeoBEARS

Usage

```
write_newick_file(phylod, path_to_phylo)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`path_to_phylo`	string specifying the path and name to write the file to.

Value

Nothing, called for side-effects.

`write_phylip_biogeo_file`

Write biogeography input file for BioGeoBEARS

Description

Write a text file containing occurrence data for all tips in the PHYLIP format expected by BioGeoBEARS

Usage

```
write_phylip_biogeo_file(phylod, path_to_biogeo)
```

Arguments

`phylod`	A `phylo4d` object from the package `phylobase` containing phylogenetic and endemicity data for each species.
`path_to_biogeo`	string specifying the path and name to write the file to.

Value

Nothing, called for side-effects.

Index

∗ **datasets**
  - coccyzus_phylod, 20
  - columbiformes_phylod, 20
  - finches_phylod, 47
  - GalapagosTrees, 47
  - mimus_phylod, 59
  - myiarchus_phylod, 61
  - plant_phylo, 62
  - progne_phylod, 65
  - pyrocephalus_phylod, 65
  - setophaga_phylod, 70

add_asr_node_states, 4
add_island_colonist, 5
add_missing_species, 7
add_multi_missing_species, 8
add_outgroup, 9
all_descendants_conspecific, 10
all_endemicity_status, 10
any_back_colonisation, 11
any_outgroup, 12
any_polyphyly, 13
as_daisie_datatable, 13

benchmark, 14
bind_colonist_to_tbl, 16

castor::asr_max_parsimony(), *4, 5*, *15*, *28*, *32*, *69*
castor::asr_mk_model(), *4, 5*, *15*, *28*, *32*, *69*
check_island_colonist, 17
check_island_tbl, 18
check_multi_island_tbl, 18
check_phylo_data, 19
coccyzus_phylod, 20
coccyzus_tree *(*GalapagosTrees*)*, 47
columbiformes_phylod, 20
columbiformes_tree *(*GalapagosTrees*)*, 47
count_missing_species, 21
create_daisie_data, 22
create_endemicity_status, 24
create_test_phylod, 25

default_params_doc, 26
dots, *5*, *32*

endemicity_to_sse_states, 32
extract_asr_clade, 33
extract_biogeobears_ancestral_states_probs, 33
extract_clade_name, 34
extract_endemic_clade, 34
extract_endemic_singleton, 35
extract_island_species, 36
extract_multi_tip_species, 38
extract_nonendemic, 39
extract_species_asr, 40
extract_species_min, 42
extract_stem_age, 43
extract_stem_age_asr, 45
extract_stem_age_genus, 45
extract_stem_age_min, 46

finches_phylod, 47
finches_tree *(*GalapagosTrees*)*, 47

GalapagosTrees, 47
get_branching_times *(*get_clade_name*)*, 48
get_branching_times,Island_colonist-method *(*get_clade_name*)*, 48
get_clade_name, 48
get_clade_name,Island_colonist-method *(*get_clade_name*)*, 48
get_clade_type *(*get_clade_name*)*, 48
get_clade_type,Island_colonist-method *(*get_clade_name*)*, 48
get_col_max_age *(*get_clade_name*)*, 48
get_col_max_age,Island_colonist-method *(*get_clade_name*)*, 48
get_col_time *(*get_clade_name*)*, 48

get_col_time,Island_colonist-method (get_clade_name), 48
get_extracted_species (get_island_tbl), 51
get_extracted_species,Island_tbl-method (get_island_tbl), 51
get_island_tbl, 51
get_island_tbl,Island_tbl-method (get_island_tbl), 51
get_min_age (get_clade_name), 48
get_min_age,Island_colonist-method (get_clade_name), 48
get_missing_species (get_clade_name), 48
get_missing_species,Island_colonist-method (get_clade_name), 48
get_num_phylo_used (get_island_tbl), 51
get_num_phylo_used,Island_tbl-method (get_island_tbl), 51
get_species (get_clade_name), 48
get_species,Island_colonist-method (get_clade_name), 48
get_sse_tip_states, 52
get_status (get_clade_name), 48
get_status,Island_colonist-method (get_clade_name), 48

is_back_colonisation, 56
is_duplicate_colonist, 57
is_identical_island_tbl, 58
Island_colonist, *48*
island_colonist, 53
Island_colonist-class, 54
Island_tbl, *51*
island_tbl, 55
Island_tbl-class, 55

mimus_phylod, 59
mimus_tree (GalapagosTrees), 47
multi_extract_island_species, 59
multi_island_tbl, 61
Multi_island_tbl-class, 61
myiarchus_phylod, 61
myiarchus_tree (GalapagosTrees), 47

plant_phylo, 62
plot_colonisation, 62
plot_performance, 63
plot_phylod, 64
progne_phylod, 65
progne_tree (GalapagosTrees), 47
pyrocephalus_phylod, 65
pyrocephalus_tree (GalapagosTrees), 47

rm_island_colonist, 66
rm_multi_missing_species, 66
round_up, 68

select_endemicity_status, 68
sensitivity, 69
set_branching_times<- (get_clade_name), 48
set_branching_times<-,Island_colonist-method (get_clade_name), 48
set_clade_name<- (get_clade_name), 48
set_clade_name<-,Island_colonist-method (get_clade_name), 48
set_clade_type<- (get_clade_name), 48
set_clade_type<-,Island_colonist-method (get_clade_name), 48
set_col_max_age<- (get_clade_name), 48
set_col_max_age<-,Island_colonist-method (get_clade_name), 48
set_col_time<- (get_clade_name), 48
set_col_time<-,Island_colonist-method (get_clade_name), 48
set_extracted_species<- (get_island_tbl), 51
set_extracted_species<-,Island_tbl-method (get_island_tbl), 51
set_island_tbl<- (get_island_tbl), 51
set_island_tbl<-,Island_tbl-method (get_island_tbl), 51
set_min_age<- (get_clade_name), 48
set_min_age<-,Island_colonist-method (get_clade_name), 48
set_missing_species<- (get_clade_name), 48
set_missing_species<-,Island_colonist-method (get_clade_name), 48
set_num_phylo_used<- (get_island_tbl), 51
set_num_phylo_used<-,Island_tbl-method (get_island_tbl), 51
set_species<- (get_clade_name), 48
set_species<-,Island_colonist-method (get_clade_name), 48
set_status<- (get_clade_name), 48

set_status<-,Island_colonist-method
    (get_clade_name), 48
setophaga_phylod, 70
setophaga_tree (GalapagosTrees), 47
sse_states_to_endemicity, 71

translate_status, 71

unique_island_genera, 72

write_biogeobears_input, 73
write_newick_file, 73
write_phylip_biogeo_file, 74