Package: phyf (via r-universe)

October 24, 2024

Title Phylogenetic Flow Objects for Easy Manipulation and Modelling of Data on Phylogenetic Trees and Graphs

Version 0.0.0.9000

Description The {phyf} package implements a tibble and vctrs based object for storing phylogenetic trees along with data. It is fast and flexible and directly produces data structures useful for phylogenetic modelling in the {fibre} package.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports ape, assertthat, fastmatch, Matrix, phangorn, phytools, pillar, purrr, tibble, vctrs, rlang, magrittr, dplyr, methods, inlabru, scales, ggplot2, utils, ggtree (>= 3.6.1), ivs

Suggests cli, knitr, rmarkdown, testthat (>= 3.0.0), withr

VignetteBuilder knitr

URL `https://rdinnager.github.io/phyf/`

Depends R (>= 2.10)

LazyData true

LazyDataCompression bzip2

Config/testthat/edition 3

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

RemoteUrl https://github.com/rdinnager/phyf

RemoteRef HEAD

RemoteSha 39f1c42a86556aed8f6f3db6b299e040b429f347

Contents

`autoplot.pf` *Make an automatic* `ggplot2` *plot for a* `pf` *object*

Description

Make an automatic `ggplot2` plot for a `pf` object

Usage

```
## S3 method for class 'pf'
autoplot(
  object,
  columns = NULL,
  layout = "circular",
  suppress_tiplabels = FALSE,
  suppress_tippoints = FALSE,
  edge_traits = FALSE,
  continuous = "colour",
  size = 1.4,
  outline_size = 1.4 * size,
  ...
)
```

Arguments

`object`	A `pf` object to plot
`columns`	Columns to plot along with the phylogeny. Can use bare column names or any other `tidyselect` syntax
`layout`	`ggtree::ggtree()` layout to use.
`suppress_tiplabels`	If `TRUE`, don't draw tip labels.
`edge_traits`	A logical indicating whether the continuous avriable refers to edge traits, where the edge is determines by the terminal node. (default is FALSE, which means the variable refers to a node trait).

`continuous`	continuous transition for selected aesthethic ('size' or 'color'('colour')). It should be one of 'color' (or 'colour'), 'size', 'all' and 'none', default is 'colour'.
`...`	Other arguments passed to or from other methods.

Value

A `ggplot` object.

Examples

```
autoplot(pf(rpfc(100)) %>% dplyr::mutate(trait = rnorm(dplyr::n())), trait,
layout = "rectangular")
```

`avonet` *AVONET Bird Trait Data with Phylogeny*

Description

The AVONET Bird Trait Database joined to a `pf` object (for `{phyf}`).

Usage

```
avonet
```

Format

`avonet`**:**

A 'pf' data frame (subclasses `tibble`) with 13,338 rows and 38 columns:

label Node labels including species name for the tip labels

phlo The phylogenetic flow column which stores the phylogenetic information

Species3, Family3, Order3 Taxonomic names – Names of the species, family and order, respectively

Total.individuals Number of individuals used to measure the data

Beak.Length_Culmen:Species.Status Various traits of the bird species, see Source section to get more detailed information

The tree was from Jetz et al. (2012). I used the same tree as used in Maliet and Morlon (2021).

Source

https://figshare.com/articles/dataset/AVONET_morphological_ecological_and_geographical_data_for_all_birds_Tobias_et_al_2021_Ecology_Letters_/16586228

References

Tobias JA, Sheard C, Pigot AL, Devenish AJ, Yang J, Sayol F, Neate-Clegg MH, Alioravainen N, Weeks TL, Barber RA, Walkden PA. AVONET: morphological, ecological and geographical data for all birds. Ecology Letters. 2022 Mar 1;25(3):581-97.
Maliet, O., & Morlon, H. (2021). Fast and accurate estimation of species-specific diversification rates using data augmentation. Systematic biology.
Jetz, W., Thomas, G.H., Joy, J.B., Hartmann, K., & Mooers, A.O. (2012). The global diversity of birds in space and time. Nature, 491, 444-448.

`bird_beak_codes` *Dataset of latent codes representing bird beak 3D shapes*

Description

A deep learning model was trained on the signed distance field of 3d bird beak scans. This dataset contain the learned latent codes that produce the bird beak shapes when passed through the trained companion neural network. The trained neural network is available from `load_model()` #d scans used to train the model were retrieved from the MarkMyBird project dataset (https://www.markmybird.org/).

Usage

```
bird_beak_codes
```

Format

`bird_beak_codes`:

A 'pf' data frame (subclasses `tibble`) with 4,040 rows and 80 columns:

label Node labels including species name for the tip labels

is_tip Logical specifying whether the row represents a tip on the phylogeny

phlo The phylogenetic flow column which stores the phylogenetic information

Common_name The English common name for the bird species

Scientific The scientific name for the bird species

Clade Various traits of the bird species, see Source section to get more detailed information

BLFamilyLatin Taxonomic family latin name

BLFamilyEnglish Taxonomic family English common name

Order Taxonomic order

OscSubOsc Oscine or Sub-Oscine

X and Y Two dimensional UMAP dimension reduction of the 64 latent variables

X0, Y0, and Z0 Three dimensional UMAP dimension reduction of the 64 latent variables

latent_code_1:latent_code_63 64 latent codes representing bird bealk shapes, estimated using a autodecoder neural network architecture `ecoregion_ids` *Ecoregion IDs*

Description

A `data.frame` with ecoregion ids and their name. This can be matched to the ecoregions referred to in the dataset `mammal_biogeo`.

Usage

```
ecoregion_ids
```

Format

`ecoregion_ids`**:**
A ’pf’ data frame (subclasses `tibble`) with 847 rows and 2 columns:

ECO_ID ID numbers for 847 ecoregions across the world

ECO_NAME The name for the corresponding ecoregion ID

Source

`https://ecoregions.appspot.com`

`make_interp` *Make a phylogenetic interpolation* `pfc`

Description

Make a phylogenetic interpolation `pfc`

Usage

```
make_interp(x, ...)
```

Arguments

`x`	A `pfc` or `pf` object
`...`	Other arguments passed to or from other methods

`mammal_biogeo` *Terrestrial Mammal Bioegeography Data with Phylogeny*

Description

Data on terrestrial mammals biogeographic distributions across the world's ecoregions joined to a `pf` object (for `{phyf}`)

Usage

```
mammal_biogeo
```

Format

`mammal_biogeo`**:**

A 'pf' data frame (subclasses `tibble`) with 10,648 rows and 844 columns:

label Node labels including species name for the tip labels

phlo The phylogenetic flow column which stores the phylogenetic information

IUCN_binomial Species name used by the IUCN, matches with IUCN range polygons

body_mass_median Mammal species' body mass

litter_clutch_size Mammal species' average clutch size (# of offspring in a litter)

activity) Mammal species' primary time of activity

hab_breadth ...

volant ...

diet_5cat ...

range_size_km2 Mammal species' range size in kilometers squared

threat Mammal species' IUCN threat category

ecoregion:eco_id Each of the 832 columns starting with "ecoregion:" represents the proportion of the mammal species' range that fall in the ecoregion with id equal to "eco_id".

Source

`https://ecoregions.appspot.com`, `https://www.iucnredlist.org/resources/spatial-data-download`

References

None yet.

pf — pf *object constructor*

Description

pf object constructor

Usage

```
pf(x = pfc(), pf_column = "phlo", ...)
```

Arguments

x	a pfc object
pf_column	Name of the column to hold x. Default: 'phlo'
...	Reserved for future extensions. Not used.

Value

a pf object

Examples

```
pf(rpfc(100))
```

pfc — *Create a new phylogenetic flow collection object (*pfc*)*

Description

Create a new phylogenetic flow collection object (pfc)

Usage

```
pfc(
  pfn = character(),
  pfpp = pfp(),
  pfl = list(),
  is_tip = logical(),
  edge_names = character(),
  internal = logical(),
  node_ord = NULL,
  edge_ord = NULL,
  sparse_mat = NULL
)

pf_is_pfc(x)
```

Arguments

`pfn`	A character vector of names for each phylogenetic flow
`pfpp`	A vector of phylogenetic flow paths of class `pfp`
`pfl`	A list of phylogenetic flow feature. Should be numeric.
`is_tip`	A logical vector specifying whether the phylogenetic flow reaches the phylogeny's tips
`edge_names`	A character vector of names for the phylogeny's edges
`internal`	A logical vector specifying whether each edge is internal (not leading to a tip)
`node_ord`	An optional integer vector specifying an order for the nodes. Usually used to store ordering information from another tree format such as `phylo` so it is easier to match data up later.
`edge_ord`	An optional integer vector specifying an order for the edges. Usually used to store ordering information from another tree format such as `phylo` so it is easier to match data up later.
`sparse_mat`	A sparse matrix representation of the phylogenetic flow collection. Can be left `NULL`, in which case it is constructed from the other arguments.
`x`	An object to be tested

Value

A `pfc` object

`pfc_from_pftibble` *Convert pftibble into a pfc*

Description

Convert pftibble into a pfc

Usage

```
pfc_from_pftibble(pft)
```

Arguments

`x`	a pftibble `tibble`

Value

A `pfc`

Examples

```
pfc_from_pftibble(pf_as_pftibble(rpfc(100)))
```

pfp *Create a new phylogenetic flow path object (pfp)*

Description

Create a new phylogenetic flow path object (`pfp`)

Usage

```
pfp(x = list())

pf_is_pfp(x)
```

Arguments

`x`	• For `pfp()`: A list of nodes (integer) the path passes through • For `is_pfp()`: An object to test.

Value

A `pfp` object

Examples

```
pfp(ape::nodepath(ape::rtree(100)))
```

pf_anc *Return a `pfc` with ancestral features*

Description

Return a `pfc` with ancestral features

Usage

```
pf_anc(x, replace = 0)
```

Arguments

`x`	A `pfc` object
`replace`	Value to replace edge feature with no ancestral value (such as when the ancestor is the root)

Value

A new `pfc` with the same structure as `x`, but with the features of each edge's ancestor instead

Examples

```
pf_anc(rpfc(100))
```

`pf_as_pf` *Convert an object to a* `pf` *object*

Description

Convert an object to a `pf` object

Usage

```
pf_as_pf(x, ...)
```

Arguments

`x`	An object to convert
`...`	Other arguments pass to or from other methods.

Value

a `pf` object with branch lengths as features

Examples

```
pf_as_pf(ape::rtree(100))
```

`pf_as_pfc` *Make a* `pfc` *object from a* `phylo` *object*

Description

Make a `pfc` object from a `phylo` object

Usage

```
pf_as_pfc(x, ...)
```

Arguments

`x`	A phylogenetic tree in `ape::phylo` format
`...`	Arguments passed to or from other methods

Value

a `pfc`

Examples

```
pf_as_pfc(ape::rtree(100))
```

`pf_as_pftibble` *Convert pfc into a pftibble*

Description

Convert pfc into a pftibble

Usage

```
pf_as_pftibble(x)
```

Arguments

`x`	a `pfc` object

Value

A `tibble`

Examples

```
pf_as_pftibble(rpfc(100))
```

`pf_as_phylo` *Convert a* `pf` *or* `pfc` *object to a* `ape::phylo` *object*

Description

This function attempts to convert a `pf` or `pfc` object to a phylogenyin `phylo` format (from package `{ape}`). It will use the feature as branch lengths, and if edges have multiple feature values, they will be aggregated by averaging them. Note that this function can fail if the `pf` or `pfc` does not have a tree-like structure. An example of this would be an 'interaction' `pfc` (as generated by `pf_interaction`)

Usage

```
pf_as_phylo(x, ...)
```

Arguments

`x`	a `pf` or `pfc` object to be converted
`...`	Other arguments passed to or from other methods

Value

A `phylo` object

Examples

```
pf_as_phylo(rpfc(100))
```

`pf_as_sparse` *Convert a* `pf` *or* `pfc` *object to a sparse matrix representation.*

Description

Convert a `pf` or `pfc` object to a sparse matrix representation.

Usage

```
pf_as_sparse(x, ...)
```

Arguments

`x`	A `pfc` or `pf` object
`...`	Arguments passed to or from other methods

Value

A `dgCMatrix` object (from the `{Matrix} package`). See `Matrix::sparseMatrix()` for details.

Examples

```
pf_as_sparse(rpfc(100))
```

`pf_desc` *Return a* `pfc` *with descendent's features*

Description

Note that in the context of a phylogenetic flow, each element of the flow only has a single descendent, which is the next edge on the sequence between the root and the terminal node of the flow in question.

Usage

```
pf_desc(x, replace = 0)
```

Arguments

`x`	A `pfc` object
`replace`	Value to replace edge feature with no descendent values (such as when the edges are the tip edges).

Value

A new `pfc` with the same structure as `x`, but with the features of each edge's descendant instead

Examples

```
pf_desc(rpfc(100))
```

`pf_edge_apply` *Apply a function along edges within a* `pfc` *object*

Description

Features of the returned `pfc` will be the output of the function applied to the original features. Therefore the functionm provided must return a vector the same length of the input.

Usage

```
pf_edge_apply(x, fun, ...)
```

Arguments

`x`	A `pfc` object
`fun`	A function to apply to the edge vectors, which will be passed as the first argument. Must return a vector the same length and type as the input.
`...`	Any additional arguments to pass to `fun`

Value

A `pfc` object with the same structure as `x` but with features replaced with the output of `fun`

Examples

```
## average features of descendents of each edge as a pfc:
pf_edge_apply(pf_desc(pf_indexes(rpfc(100))), function(x) rep(mean(x), length(x)))
```

`pf_edge_names` *Extract edge names from `pfc` object*

Description

Extract edge names from `pfc` object

Usage

```
pf_edge_names(x)
```

Arguments

`x`	`pfc` object

Value

A character vector of edge names

Examples

```
pf_edge_names(rpfc(100))
```

`pf_edge_segmentize` *Segments the edges of a phylogeny by splitting them at particular positions*

Description

Segments the edges of a phylogeny by splitting them at particular positions

Usage

```
pf_edge_segmentize(x, edges, positions)
```

Arguments

`x`	A `pfc` object
`edges`	A vector of edge names
`positions`	A vector of position to cut the edges (must be same length as `edges`)

Value

A `pfc` object with edges segmented

Examples

```
pf_edge_segmentize(rpfc(100), "t1", 0.01)
```

pf_ends — *Returns the end edge of each phylogenetic flow as a two-column tibble with start and end columns*

Description

Returns the end edge of each phylogenetic flow as a two-column tibble with start and end columns

Usage

```
pf_ends(x, return_names = TRUE)
```

Arguments

x	A pfc object
return_names	If TRUE, return the start and end node names. If FALSE, return their indexes.

Value

A `tibble` with start and end nodes

Examples

```
pf_ends(rpfc(100))
```

pf_end_features — *Extract the features of the end edge of each phylogenetic flow*

Description

Extract the features of the end edge of each phylogenetic flow

Usage

```
pf_end_features(x)
```

Arguments

x	A `pfc` object

Value

A numeric vector of the features of the end edge of each phylogenetic flow

Examples

```
pf_end_features(rpfc(100))
```

pf_epoch_info — *Get edges and positions along edges where a set of epoch times intersect*

Description

Get edges and positions along edges where a set of epoch times intersect

Usage

```
pf_epoch_info(x, times)
```

Arguments

`x`	An object of class pfc
`times`	A vector of epoch times to slice the pfc along

Value

A tibble with edge labels in the first column and position in the second column

Examples

```
pf_epoch_info(rpfc(100), c(1, 2, 3))
```

pf_filter_with_mrca — *Filter descendents of a MRCA*

Description

Filter a `pf` or `pfc` object by taking all descendents of the most recent common ancestor (mrca) of `descendents`

Usage

```
pf_filter_with_mrca(
  x,
  descendents,
  drop_null_edges = TRUE,
  drop_root_edges = TRUE,
  ...
)
```

Arguments

`x`	A `pf` or `pfc` object
`descendents`	An index into a `pfc`, using any way that you can index a `pfc`. `filter_with_mrca` will find the most recent common ancestor of everything in `descendents` and then it will return a filtered `pf` or `pfc` with all descendents of the mrca. If of length one, it will instead be assumed to be the node name or number representing the mrca itself.
`drop_null_edges`	If `TRUE` (the default), drop edges with no descendants after the filtering.
`drop_root_edges`	If `TRUE` (the default), drop all edges leading up to the mrca (e.g. when combined together these would make a root edge leading to the desired clade)
`...`	Other arguments for future extensions.

Value

A `pf` or `pfc` object

Examples

```
avonet %>%
    pf_filter_with_mrca(label %in% c("Platalea_minor", "Pelecanus_occidentalis"))
```

`pf_flow_cumsum` *Calculate the cumulative sum of features for each flow*

Description

Cumulative sum the phylogenetic flow features for each flow in a `pfc`, returning a `pfc`.

Usage

```
pf_flow_cumsum(x, direction = c("from_root", "to_root"), ...)
```

Arguments

`x`	A `pfc` object
`direction`	Which direction to take the cumulative sum along? "from_root", the default, goes from the root to the terminal nodes. "to_root" goes in the opposite direction, from the terminal nodes to the root.
`...`	Other arguments to pass to or from other methods

Value

A `pfc` object with cumulative sums of features in place of the original features

Examples

```
pf_flow_cumsum(rpfc(100))
```

`pf_flow_sum` *Calculate the sum of features for each flow*

Description

Sums the phylogenetic flow features for each flow in a `pfc`, returning a vector of values.

Usage

```
pf_flow_sum(x, ...)
```

Arguments

`x`	A `pfc` object
`...`	Other arguments to pass to or from other methods

Value

A vector equal to the length of `x` with the sum of each flow's features

Examples

```
pf_flow_sum(rpfc(100))
```

`pf_indexes` *Replace feature with edge index in a* `pfc`

Description

This is similar to base R's `col()` function, it returns a `pfc` with features that are just the index of the edge. Useful for complex indexing procedures

Usage

```
pf_indexes(x)
```

Arguments

`x`	A `pfc` object

Value

A new `pfc` with the same structure as `x`, but with the features of each edge equal to the edge index (the column number in the sparse matrix representation)

Examples

```
pf_indexes(rpfc(100))
```

---

pf_internal *Extract or assign into the internal or terminal edges of a* pfc

---

Description

pf_internal extracts or assigns a pfc or dgCMatrix into the internal edges of a pfc.

Usage

```
pf_internal(x, ...)

pf_internal(x) <- value

pf_terminal(x, ...)

pf_terminal(x) <- value
```

Arguments

x	a pfc object
...	Arguments passed to or from other methods.
value	a pfc or dgCMatrix with the same dimensions as pf_internal(x) or pf_terminal(x)

Details

pf_terminal extracts or assigns a pfc or dgCMatrix into the terminal edges of a pfc.

Value

a pfc object with flows truncated to internal nodes.

Examples

```
pfc1 <- rpfc(100)
# Pagel's lambda transformation:
lambda <- 0.5
lambda_pfc <- pfc1
root2node_lens <- pf_flow_sum(lambda_pfc)
pf_internal(lambda_pfc) <- pf_internal(lambda_pfc) * lambda
pf_terminal(lambda_pfc) <- pf_terminal(lambda_pfc) * (root2node_lens / pf_flow_sum(lambda_pfc))
plot(lambda_pfc)
```

`pf_is_desc` *Return a logical vector determining if a flow's terminal node is a descendant of an edge.*

Description

Return a logical vector determining if a flow's terminal node is a descendant of an edge.

Usage

```
pf_is_desc(x, edge)
```

Arguments

`x`	A `pfc` object
`edge`	An edge number or name

Value

A logical vector

Examples

```
pf_is_desc(rpfc(100), "Node12")
```

`pf_is_empty` *Test which elements of a pfc are empty*

Description

Test which elements of a pfc are empty

Usage

```
pf_is_empty(x, ...)
```

Arguments

`x`	A `pfc` object
`...`	Further arguments for future extensions. Currently not used.

Value

A logical vector of the same length as `x` with `TRUE` where there are empty flows in `x`, `FALSE` otherwise

Examples

```
pf_is_empty(rpfc(100))
```

`pf_is_tips` *Return a logical vector which is* `TRUE` *for the elements of a* `pfc` *whci reprsent tips of a phylogeny*

Description

Return a logical vector which is `TRUE` for the elements of a `pfc` whci reprsent tips of a phylogeny

Usage

```
pf_is_tips(x, ...)
```

Arguments

`x`	A `pfc` object.
`...`	Other arguments passed to or from other methods

Examples

```
pf_is_tips(rpfc(100))
```

`pf_kronecker` *Calculate a kronecker product when a* `pfc` *is the multiplicand or the multiplier.*

Description

Calculate a kronecker product when a `pfc` is the multiplicand or the multiplier.

Usage

```
pf_kronecker(x, y, ...)
```

Arguments

`x`	The kronecker multiplicand.
`y`	The kronecker multiplier.
`...`	Other arguments passed to or from other methods.

Value

A `pfc` object

Examples

```
pf_kronecker(rpfc(20), rpfc(20))
```

`pf_labels` *Return a vector of labels for a* `pfc` *object*

Description

Return a vector of labels for a `pfc` object

Usage

```
pf_labels(x)
```

Arguments

`x`	A `pfc` object

Value

A character vector of labels

Examples

```
pf_labels(rpfc(100))
```

`pf_mean_edge_features` *Extract mean edge features from* `pfc` *object*

Description

Extract mean edge features from `pfc` object

Usage

```
pf_mean_edge_features(x)
```

Arguments

`x`	`pfc` object

Value

A numeric vector of mean edge features

Examples

```
pf_mean_edge_features(rpfc(100))
```

pf_mrca *Return the edge leading up to the most recent common ancestor of a set of phylogenetic flows*

Description

Return the edge leading up to the most recent common ancestor of a set of phylogenetic flows

Usage

```
pf_mrca(x, name = FALSE, ...)
```

Arguments

`x`	A `pfc` object
`name`	Should the edge name be returned? If `FALSE`, the default, the edge number is returned
`...`	Other arguments passed to or from other methods

Value

A string with the edge name is `name = TRUE`, or an integer index otherwise

Examples

```
require(dplyr)
avonet %>%
  filter(label %in% c("Platalea_minor", "Pelecanus_occidentalis")) %>%
  pull(phlo) %>%
  pf_mrca()
```

pf_nedges *Return the number of edges in a* `pfc`

Description

Return the number of edges in a `pfc`

Usage

```
pf_nedges(x)
```

Arguments

`x`	a `pfc` object

Value

The number of edges

Examples

```
pf_nedges(rpfc(100))
```

`pf_nodes` *Get only the (internal) node elements of a* `pfc`

Description

Get only the (internal) node elements of a `pfc`

Usage

```
pf_nodes(x)
```

Arguments

`x`	A `pfc` object

Value

A `pfc` object with only internal node elements

Examples

```
pf_nodes(rpfc(100))
```

`pf_ones` *Replace features with ones*

Description

Replace features with ones

Usage

```
pf_ones(x)
```

Arguments

`x`	A `pfc` object

Value

A `pfc` object with all feature values replaced with ones

Examples

```
pf_ones(rpfc(100))
```

`pf_path` *Extract `pfp` object from `pfc`, the paths of each flow from root to terminal node.*

Description

Extract `pfp` object from `pfc`, the paths of each flow from root to terminal node.

Usage

```
pf_path(x)
```

Arguments

`x`	A `pfc` object

Value

A `pfp` object

Examples

```
pf_path(rpfc(100))
```

`pf_phyloflow` *Extracts the phylogenetic flow column of an `pf` object*

Description

Extracts the phylogenetic flow column of an `pf` object

Usage

```
pf_phyloflow(x)
```

Arguments

`x`	A `pf` object

Value

A `pfc` object (phylogenetic flow collection)

Examples

```
pf_phyloflow(avonet)
```

`pf_position` *Replace feature with edge position in flow in a* `pfc`

Description

Replace feature with edge position in flow in a `pfc`

Usage

```
pf_position(x)
```

Arguments

`x`	A `pfc` object

Value

A new `pfc` with the same structure as `x`, but with the features of each edge equal to the position along the flow as an integer index

Examples

```
pf_position(rpfc(100))
```

`pf_row_kron` *Calculate a rowwise kronecker product when the multiplicand or multiplier is a* `pfc`*.*

Description

Calculate a rowwise kronecker product when the multiplicand or multiplier is a `pfc`.

Usage

```
pf_row_kron(x, y, ...)
```

Arguments

`x`	The rowwise kronecker multiplicand
`y`	The rowwise kronecker multiplier
`...`	Other arguments passed to or from other methods.

Value

A `pfc` object.

Examples

```
pf_row_kron(rpfc(20), rpfc(20))
```

`pf_scale_flow_sum` *Scale the phylogenetic flow features to a constant sum.*

Description

Can be used to standardise branch lengths to reasonable values

Usage

```
pf_scale_flow_sum(x, scale_to = 1)
```

Arguments

`x`	a `pfc` object
`scale_to`	The value to scale the sums to. Default is 1.

Value

a scaled `pfc` object

Examples

```
pf_scale_flow_sum(rpfc(100))
```

`pf_second_order`	*Calculate a 'second order'* `pfc`*, which represents summed branch length from each node to the terminal node along each flow. When used in modelling traits, a second order* `pfc` *implies that rates of evolution of the trait are themselves evolving according to Brownian motion.*

Description

Calculate a 'second order' `pfc`, which represents summed branch length from each node to the terminal node along each flow. When used in modelling traits, a second order `pfc` implies that rates of evolution of the trait are themselves evolving according to Brownian motion.

Usage

```
pf_second_order(x, ...)
```

Arguments

`x`	A `pfc` object
`...`	Additional arguments passed to or other methods.

Value

A `pfc` object

Examples

```
pf_second_order(rpfc(100))
```

`pf_standardise`	*Standardise the phylogenetic flow features to an implied typical variance of 1.*

Description

Can be used to make random effects based on `pfc`s comparable.

Usage

```
pf_standardise(x)
```

Arguments

`x`	a `pfc` object

Value

a scaled `pfc` object

Examples

```
pf_standardise(rpfc(100))
```

`pf_tips` *Get only the tip elements of a* `pfc`

Description

Get only the tip elements of a `pfc`

Usage

```
pf_tips(x)
```

Arguments

`x`	A `pfc` object

Value

A `pfc` object with only tip elements

Examples

```
pf_tips(rpfc(100))
```

`pf_vcv` *Calculate phylogenetic variance covariance matrix from* `pfc`

Description

Calculate phylogenetic variance covariance matrix from `pfc`

Usage

```
pf_vcv(x, ...)
```

Arguments

`x`	A `pfc` object
`...`	Other arguments for future extensions.

Value

A `sparseMatrix`

Examples

```
pf_vcv(rpfc(10))
```

`pf_zeros` *Replace features with zeros*

Description

Replace features with zeros

Usage

```
pf_zeros(x)
```

Arguments

`x`	A `pfc` object

Value

A `pfc` object with all feature values replaced with zeros

Examples

```
pf_zeros(rpfc(100))
```

`plant_fungus` *Dataset of latent codes representing bird beak 3D shapes*

Description

A deep learning model was trained on the signed distance field of 3d bird beak scans. This dataset contain the learned latent codes that produce the bird beak shapes when passed through the trained companion neural network. The trained neural network is available from `load_model()` #d scans used to train the model were retrieved from the MarkMyBird project dataset (https://www.markmybird.org/).

Usage

```
plant_fungus
```

Format

`plant_fungus`:

A 'pf' data frame (subclasses `tibble`) with 4,040 rows and 80 columns:

label Node labels including species name for the tip labels

is_tip Logical specifying whether the row represents a tip on the phylogeny

phlo The phylogenetic flow column which stores the phylogenetic information

Common_name The English common name for the bird species

Scientific The scientific name for the bird species

Clade Various traits of the bird species, see Source section to get more detailed information

BLFamilyLatin Taxonomic family latin name

BLFamilyEnglish Taxonomic family English common name

Order Taxonomic order

OscSubOsc Oscine or Sub-Oscine

X and Y Two dimensional UMAP dimension reduction of the 64 latent variables

X0, Y0, and Z0 Three dimensional UMAP dimension reduction of the 64 latent variables

latent_code_1:latent_code_63 64 latent codes representing bird bealk shapes, estimated using a autodecoder neural network architecture

Source

`https://datadryad.org/stash/dataset/doi:10.5061/dryad.723m1`

`plot.pf` *Make a plot for a* `pf` *object*

Description

Make a plot for a `pf` object

Usage

```
## S3 method for class 'pf'
plot(x, columns = NULL, layout = "fan", suppress_tiplabels = FALSE, ...)
```

Arguments

`x`	A `pf` obect to plot
`columns`	Bare column names of variables to plot with tree.
`layout`	Either 'phylogram' or 'fan'
`...`	Other arguments passed to `phytools::contMap()`

Value

A `phytools::contMap()` object.

`primate_diet` *Primate Diet Diversity and Threat Status Data with Phylogeny*

Description

Data on primates diets and their threat status joined to a `pf` object (for `{phyf}`)

Usage

```
primate_diet
```

Format

`primate_diet`**:**

A 'pf' data frame (subclasses `tibble`) with 504 rows and 55 columns:

label Node labels including species name for the tip labels

phlo The phylogenetic flow column which stores the phylogenetic information

Threat status IUCN threat category

Threat status source Threat status data source – see `primate_refs` dataset

Body mass (g) Primate species' mean body mass in grams

Body mass source) Body mass data source – see `primate_refs` dataset

Range size (km2) Primate species' range size in kilometers

Range size source) Range size data source – see `primate_refs` dataset

Diet disparity (PSV) Primate species' disparity of diet items as measured by Phylogenetic Species Variability (PSV) metric.

Diet breadth Primate species' breadth of diet items as measured by the number of different diet items (richness).

Diet diversity (DDI) Primate species' diversity of diet items as measured by the DDI metric.

Trophic guild Primate species' trophic guild. Possible values: "Omnivore", "Frugivore", "Gummivore", "Insectivore", or "Folivore-frugivore"

diet_item:item (40 columns) Each of the 40 columns starting with "diet_item:" represents a different type of item in primates' diets. These are binary integer columns with a 1 if the species feeds on that diet item or 0 if it does not.

Source

https://zslpublications.onlinelibrary.wiley.com/doi/full/10.1111/acv.12823

References

Machado, F. F., Jardim, L., Dinnage, R., Brito, D., & Cardillo, M. (2022). Diet disparity and diversity predict extinction risk in primates. Animal Conservation.

`primate_diet_hierarchy` *Primate Diet Items Hierarchy*

Description

A `data.frame` representing an hierarchical categorization of primate diet items

Usage

```
primate_diet_hierarchy
```

Format

`primate_hierarchy`**:**

A data frame with 40 rows and 6 columns:

FOOD ITEM Food item name. This matches the ’item’ ’diet_item:item’ columns in the `primate_diet` data set

level 2 A category categorizing the food items immediately above the items themselves

level 3:level 6 (4 columns) Additional categories arranged in an heirarchy

Source

https://zslpublications.onlinelibrary.wiley.com/doi/full/10.1111/acv.12823

References

Machado, F. F., Jardim, L., Dinnage, R., Brito, D., & Cardillo, M. (2022). Diet disparity and diversity predict extinction risk in primates. Animal Conservation.

`primate_diet_refs` *Primate Diet Data References*

Description

A `data.frame` with references for the Primate Diet Data. See the `primate_diet` data set.

Usage

```
primate_diet_refs
```

Format

`primate_diet_refs`**:**

A 'pf' data frame (subclasses `tibble`) with 4 rows and 3 columns:

FOOD ITEM Food item name. This matches the 'item' 'diet_item:item' columns in the `primate_diet` data set

level 2 A category categorizing the food items immediately above the items themselves

level 3:level 6 (4 columns) Additional categories arranged in an heirarchy

Source

https://zslpublications.onlinelibrary.wiley.com/doi/full/10.1111/acv.12823

References

Machado, F. F., Jardim, L., Dinnage, R., Brito, D., & Cardillo, M. (2022). Diet disparity and diversity predict extinction risk in primates. Animal Conservation.

`rpfc` *Generate a random tree and return it as a* `pfc` *object*

Description

Generate a random tree and return it as a `pfc` object

Usage

```
rpfc(n, method = ape::rcoal, ...)
```

Arguments

`n`	Number of tips in the generated tree.
`method`	A function to generate a tree. Default is `ape::rcoal`. See `ape::rtree()` for more options.
`...`	Additional arguments to pass to `method`

Value

a `pfc` object

Examples

```
plot(rpfc(100))
```

`vert_bmr` *Vertebrate Base Metabolic Rates with Phylogeny*

Description

Data on vertebrate base Metabolic rates joined to a `pf` object (for `{phyf}`)

Usage

```
vert_bmr
```

Format

`vert_bmr`:
A 'pf' data frame (subclasses `tibble`) with 1,712 rows and 8 columns:

label Node labels including species name for the tip labels
phlo The phylogenetic flow column which stores the phylogenetic information
lnBMR Natural log of the base metabolic rate
lnMass Natural log of body mass
lnMass2 Squared natural log of body mass
lnGS Natural log of genome size
endo Is the species endothermic? 1 for yes, 0 for no

Source

`https://datadryad.org/stash/dataset/doi:10.5061/dryad.3c6d2`

References

Uyeda JC, Pennell MW, Miller ET, Maia R, McClain CR. The evolution of energetic scaling across the vertebrate tree of life. The American Naturalist. 2017 Aug 1;190(2):185-99.

Index