Package 'delimtools'

March 23, 2026

Type Package

Title Helper Functions for Species Delimitation Analysis

Version 0.2.2

Date 2026-03-20

Description Helpers functions to process, analyse, and visualize the output of single locus species delimitation methods.
For full functionality, please install suggested software at <https://legallab.github.io/delimtools/articles/install.html>.

License MIT + file LICENSE

Depends R (>= 4.2.0)

Imports cli, dplyr, glue, ggtree, methods, purrr, rlang, stringr, tidyr, tidyselect

Suggests ape, bGMYC (>= 1.0.2), forcats, furrr, future, ggfun, ggplot2, gtools, here, knitr, patchwork, randomcoloR, readr, RColorBrewer, scales, splits (>= 1.0.20), spider, tibble, tidytree, vctrs, withr, rmarkdown

Additional_repositories https://r-forge.r-project.org/, https://pedrosenna.github.io/drat/

Config/Needs/website bGMYC=github::https://github.com/pedrosenna/bGMYC, splits=url::https://download.r-forge.r-project.org/src/contrib/splits_1.0-20.tar.gz

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Maintainer Pedro Bittencourt <pedro.sennabittencourt@gmail.com>

URL https://github.com/legalLab/delimtools, https://legallab.github.io/delimtools/

BugReports https://github.com/legalLab/delimtools/issues

VignetteBuilder knitr

Author Pedro Bittencourt [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-4074-3541>),
Rupert Collins [aut, ctb, cph] (ORCID:
<https://orcid.org/0000-0002-9135-1169>),
Tomas Hrbek [aut, ctb] (ORCID: <https://orcid.org/0000-0003-3239-7068>)

Repository CRAN

Date/Publication 2026-03-23 08:30:02 UTC

Contents

`delimtools-package` *Helper Functions for Species Delimitation Analysis*

Description

Helpers functions to process, analyse, and visualize the output of single locus species delimitation methods. For full functionality, please install suggested software at https://legallab.github.io/delimtools/articles/install.html.

Author(s)

Maintainer: Pedro Bittencourt <pedro.sennabittencourt@gmail.com> (ORCID) [copyright holder]

Authors:

- Rupert Collins <rupertcollins@gmail.com> (ORCID) [contributor, copyright holder]
- Tomas Hrbek <hrbek@evoamazon.net> (ORCID) [contributor]

See Also

Useful links:

- https://github.com/legalLab/delimtools
- https://legallab.github.io/delimtools/
- Report bugs at https://github.com/legalLab/delimtools/issues

---

`abgd_tbl` *A Command-Line Interface for ABGD - Automatic Barcode Gap Discovery*

Description

`abgd_tbl()` returns species partition hypothesis estimated by ABGD software (https://bioinfo.mnhn.fr/abi/public/abgd/).

Usage

```
abgd_tbl(
  infile,
  exe = NULL,
  haps = NULL,
  slope = 1.5,
  model = 3,
  outfolder = NULL,
  webserver = NULL,
  delimname = "abgd"
)
```

Arguments

infile	Path to fasta file.
exe	Path to an ABGD executable.
haps	Optional. A vector of haplotypes to keep into the tbl_df.
slope	Numeric. Relative gap width (slope). Default to 1.5.
model	An integer specifying evolutionary model to be used. Available options are: • 0: Kimura-2P • 1: Jukes-Cantor (default) • 2: Tamura-Nei • 3: simple distance (p-distance)
outfolder	Path to output folder. Default to NULL. If not specified, a temporary location is used.
webserver	A .txt file containing ABGD results obtained from a webserver. Default to NULL.
delimname	Character. String to rename the delimitation method in the table. Default to 'abgd'.

Details

abgd_tbl() relies on system to invoke ABGD software through a command-line interface. Hence, you must have the software available as an executable file on your system in order to use this function properly. abgd_tbl() saves all output files in outfolder and imports the first recursive partition file generated to Environment. Alternatively, abgd_tbl() can parse a .txt file obtained from a webserver such as (https://bioinfo.mnhn.fr/abi/public/abgd/abgdweb.html).

Value

an object of class tbl_df

Author(s)

N. Puillandre, A. Lambert, S. Brouillet, G. Achaz

Source

Puillandre N., Lambert A., Brouillet S., Achaz G. 2012. ABGD, Automatic Barcode Gap Discovery for primary species delimitation. Molecular Ecology 21(8):1864-77.

Examples

```
#' # get path to fasta file
path_to_file <- system.file("extdata/geophagus.fasta", package = "delimtools")

# run ABGD
abgd_df <- try( abgd_tbl(
  infile = path_to_file,
```

```
  exe = "/usr/local/bin/abgd",
  model = 3,
  slope = 0.5,
  outfolder = NULL
)
)
# check
try(abgd_df)
```

---

asap_tbl *A Command-Line Interface for ASAP - Assemble Species by Automatic Partitioning*

Description

`asap_tbl()` returns species partition hypothesis estimated by ASAP software (https://bioinfo.mnhn.fr/abi/public/asap/).

Usage

```
asap_tbl(
  infile,
  exe = NULL,
  haps = NULL,
  model = 3,
  outfolder = NULL,
  webserver = NULL,
  delimname = "asap"
)
```

Arguments

`infile`	Path to fasta file.
`exe`	Path to an ASAP executable.
`haps`	Optional. A vector of haplotypes to keep into the tbl_df.
`model`	An integer specifying evolutionary model to be used. Available options are: • 0: Kimura-2P • 1: Jukes-Cantor (default) • 2: Tamura-Nei • 3: simple distance (p-distance)
`outfolder`	Path to output folder. Default to NULL. If not specified, a temporary location is used.
`webserver`	A .csv file containing ASAP results obtained from a webserver. Default to NULL.
`delimname`	Character. String to rename the delimitation method in the table. Default to 'asap'.

Details

`asap_tbl()` relies on system to invoke ASAP software through a command-line interface. Hence, you must have the software available as an executable file on your system in order to use this function properly. `asap_tbl()` saves all output files in `outfolder` and imports the first partition file generated to `Environment`. Alternatively, `asap_tbl()` can parse a .csv file obtained from webserver such as (https://bioinfo.mnhn.fr/abi/public/asap/asapweb.html).

Value

an object of class tbl_df

Author(s)

Nicolas Puillandre, Sophie Brouillet, Guillaume Achaz.

Source

Puillandre N., Brouillet S., Achaz G. 2021. ASAP: assemble species by automatic partitioning. Molecular Ecology Resources 21:609–620.

Examples

```
#' # get path to fasta file
path_to_file <- system.file("extdata/geophagus.fasta", package = "delimtools")

# run ASAP
asap_df <- try(asap_tbl(infile = path_to_file, exe= "/usr/local/bin/asap", model= 3))

# check
try(asap_df)
```

---

as_dwc *Rename Columns using Darwin Core Standard Terms*

Description

`as_dwc()` rename columns in a tbl_df using a vector of terms defined by Darwin Core Standard.

Usage

```
as_dwc(dwc, data, terms)
```

Arguments

dwc	a list of standard terms and definitions created using get_dwc().
data	a tbl_df.
terms	a vector or list of terms to be used as replacement.

Details

as_dwc() will replace current column names by the ones defined in terms. For each column in data, Darwin Core equivalent terms must be informed in the same order by the user. If terms and column names do not match in length or if terms used are not found in Darwin Core standard, an error will be printed on Console.

Value

an object of class tbl_df.

Author(s)

Pedro S. Bittencourt, Rupert A. Collins.

Examples

```
# get dwc terms and definitions
dwc <- get_dwc(type = "simple")

# create a data frame with sample metadata
my_df <- tibble::tibble(
  species = c("sp1", "sp2", "sp3"),
  location = c("loc1", "loc2", "loc3"),
  voucher = c("M01", "M02", "M03"),
  collector = c("John", "Robert", "David")
)

# rename columns
as_dwc(dwc, my_df, terms = c("scientificName", "locality", "catalogNumber", "recordedBy"))
```

---

bgmyc_tbl *Turns bGMYC Results Into a Tibble*

Description

bgmyc_tbl() processes output from bgmyc.singlephy into an object of class tbl_df.

Usage

```
bgmyc_tbl(bgmyc_res, ppcutoff = 0.05, delimname = "bgmyc")
```

Arguments

bgmyc_res	Output from bgmyc.singlephy.
ppcutoff	Posterior probability threshold for clustering samples into species partitions. See bgmyc.point for details. Default to 0.05.
delimname	Character. String to rename the delimitation method in the table. Default to 'bgmyc'.

Details

bGMYC package uses spec.probmat to create a matrix of probability of conspecificity and bgmyc.point to split samples into a list which individuals meets the threshold specified by ppcutoff. bgmyc_tbl() wraps up these two functions into a single one and turns these inputs into a tibble which matches the output from gmyc_tbl and locmin_tbl.

Value

an object of class tbl_df.

Author(s)

Noah M. Reid.

Source

Reid N.M., Carstens B.C. 2012. Phylogenetic estimation error can decrease the accuracy of species delimitation: a Bayesian implementation of the general mixed Yule-coalescent model. BMC Evolutionary Biology 12 (196).

Examples

```
# run bGMYC
bgmyc_res <- try( bGMYC::bgmyc.singlephy(ape::as.phylo(geophagus_beast),
  mcmc = 11000,
  burnin = 1000,
  thinning = 100,
  t1 = 2,
  t2 = ape::Ntip(geophagus_beast),
  start = c(1, 0.5, 50)
)
)
# create a tibble
bgmyc_df <- try( bgmyc_tbl(bgmyc_res, ppcutoff = 0.05) )

# check
try(bgmyc_df)
```

check_delim *Checks If Two or More Species Delimitation Outputs are (Nearly) Equal*

Description

`check_delim()` checks if two or more species delimitation outputs have differences in its dimensions, labels, and values.

Usage

```
check_delim(list)
```

Arguments

`list`	a list containing two or more species delimitation outputs to check.

Details

`check_delim()` will check if two or more species delimitation outputs have different dimensions (rows, columns), if labels are the same or if there are any duplicated or absent labels, and if there are any NA values or if partitions were set using non numeric values. If `TRUE` for any of the cases listed above, `check_delim()` will return an error.

Value

A single logical value, `TRUE` or `FALSE`.

Author(s)

Pedro S. Bittencourt, Rupert A. Collins.

Examples

```
# create dummy delimitation outputs
delim_1 <- tibble::tibble(
  labels = paste0("seq", 1:10),
  method_A = c(rep(1, 5), rep(2, 5))
)

delim_2 <- tibble::tibble(
  labels = paste0("seq", 1:10),
  method_B = c(rep(1, 3), rep(2, 2), rep(3, 5))
)

delim_3 <- tibble::tibble(
  labels = paste0("seq", 1:10),
  method_C = c(rep(1, 3), rep(2, 2), rep(3, 3), rep(4, 2))
)
```

```
# check outputs
check_delim(list(delim_1, delim_2, delim_3))
```

`check_identifiers` *Checks for Differences Between Identifiers in Metadata and DNA Sequence Files*

Description

`check_identifiers()` checks for differences between identifiers in metadata and DNA sequence files.

Usage

```
check_identifiers(data, identifier, dna)
```

Arguments

`data`	an object of class tbl_df containing sequence metadata.
`identifier`	column in `data` which contains sequence identifiers.
`dna`	a DNAbin object.

Details

`check_identifiers()` is a helper function to check for inconsistencies between identifiers in metadata and DNA sequences files, such as absence, mistyping, duplicated entries, or differences in size lengths. If any of these problems are found, warnings will appear in `Console` and corrections should be made to prevent unintended consequences later. A list containing erroneous identifiers is returned invisibly.

Value

A list containing erroneus identifiers between metadata and sequence file.

Author(s)

Pedro S. Bittencourt, Rupert A. Collins.

Examples

```
check_identifiers(geophagus_info, "gbAccession", geophagus)
```

clean_dna *Removes Gaps, Ambiguities and Missing Data from DNA Sequences*

Description

`clean_dna()` removes all character not a valid ACTG base from a DNAbin object.

Usage

```
clean_dna(dna, verbose = TRUE)
```

Arguments

`dna`	an object of class DNAbin.
`verbose`	logical. Returns a warning if any sequence contains non ACTG bases.

Details

`clean_dna()` detects and removes any non ACTG bases from alignment. This includes: "N", "-", "?", "R", "Y", etc. If `verbose = TRUE`, returns a warning if these characters are inside the sequences, i.e, are not alignment padding chars at the ends.

Value

an object of class DNAbin.

Author(s)

Rupert A. Collins

Examples

```
geo_clean <- clean_dna(geophagus)
```

collapse_others *Summarise Haplotype Metadata Down to One Row*

Description

`collapse_others()` returns a tbl_df summarising all unique haplotype frequencies, duplicates and selected metadata into a single row.

Usage

```
collapse_others(data, hap_tbl, labels, cols)
```

Arguments

data	An object of class tbl_df containing sequence metadata.
hap_tbl	Output from haplotype_tbl.
labels	Column name which contains sequence names.
cols	A character vector of variables to collapse.

Details

collapse_others() is a helper function to summarise metadata along with haplotype_tbl. For any given cols, collapse_others() flattens its content by unique haplotypes and its duplicates in hap_tbl.

Value

an object of class tbl_df.

Author(s)

Pedro S. Bittencourt, Rupert A. Collins.

Examples

```
# summarise haplotypes
hap_tbl <- haplotype_tbl(geophagus)

# summarise country
others_df <- collapse_others(geophagus_info, hap_tbl, "gbAccession", "country")
```

---

confidence_intervals *Confidence Intervals for Species Delimitations Methods*

Description

These functions compute confidence intervals for various species delimitation methods, including GMYC, bGMYC, Local Minima, and mPTP.

Usage

```
gmyc_ci(tr, posterior, method = "single", interval = c(0, 5))

bgmyc_ci(
  tr,
  posterior,
  ppcutoff = 0.05,
  mcmc,
  burnin,
```

```
  thinning,
  py1 = 0,
  py2 = 2,
  pc1 = 0,
  pc2 = 2,
  t1 = 2,
  t2 = 51,
  scale = c(20, 10, 5),
  start = c(1, 0.5, 50)
)

locmin_ci(dna, block = 1, reps = 100, threshold = 0.01, haps = NULL, ...)

mptp_ci(
  infile,
  bootstraps,
  exe = NULL,
  outfolder = NULL,
  method = c("multi", "single"),
  minbrlen = 1e-04,
  webserver = NULL
)
```

Arguments

tr	An ultrametric, dichotomous tree object in ape format.
posterior	Trees from posterior. An object of class multiphylo.
method	Method of analysis, either "single" for single-threshold version or "multiple" for multiple-threshold version.
interval	Upper and lower limit of estimation of scaling parameters, e.g. c(0,10)
ppcutoff	Posterior probability threshold for clustering samples into species partitions. See bgmyc.point for details. Default to 0.05.
mcmc	number of samples to take from the Markov Chain
burnin	the number of samples to discard as burn-in
thinning	the interval at which samples are retained from the Markov Chain
py1	governs the prior on the Yule (speciation) rate change parameter. using the default prior distribution, this is the lower bound of a uniform distribution. this can be the most influential prior of the three. rate change is parameterized as n^py where n is the number of lineages in a waiting interval (see Pons et al. 2006). if there are 50 sequences in an analysis and the Yule rate change parameter is 2, this allows for a potential 50-fold increase in speciation rate. this unrealistic parameter value can cause the threshold between Yule and Coalescent process to be difficult to distinguish. are more reasonable upper bound for the prior would probably be less than 1.5 (a potential 7-fold increase). Or you could modify the prior function to use a different distribution entirely.
py2	governs the prior on the Yule rate change parameter. using the default prior distribution, this is the upper bound of a uniform distribution.

pc1	governs the prior on the coalescent rate change parameter. using the default prior distribution, this is the lower bound of a uniform distribution. rate change is parameterized as (n(n-1))^pc where n is the number of lineages in a waiting interval (see Pons et al. 2006). In principle pc can be interpreted as change in effective population size (pc<1 decline, pc>1 growth) but because identical haplotypes must be excluded from this analysis an accurate biological interpretation is not possible.
pc2	governs the prior on the coalescent rate change parameter. using the default prior distribution, this is the upper bound of a uniform distribution.
t1	governs the prior on the threshold parameter. the lower bound of a uniform distribution. the bounds of this uniform distribution should not be below 1 or greater than the number of unique haplotypes in the analysis.
t2	governs the prior on the threshold parameter. the upper bound of a uniform distribution
scale	a vector of scale parameters governing the proposal distributions for the markov chain. the first to are the Yule and coalescent rate change parameters. increasing them makes the proposals more conservative. the third is the threshold parameter. increasing it makes the proposals more liberal.
start	a vector of starting parameters in the same order as the scale parameters, py, pc, t. t may need to be set so that it is not impossible given the dataset.
dna	an object of class DNAbin.
block	integer. Number of columns to be resampled together. Default to 1.
reps	Number of bootstrap replicates. Default to 100.
threshold	Distance cutoff for clustering. Default of 0.01. See localMinima for details.
haps	Optional. A vector of haplotypes to keep into the tbl_df.
...	Further arguments to be passed to dist.dna.
infile	Path to tree file in Newick format. Should be dichotomous and rooted.
bootstraps	Bootstrap trees. An object of class multiphylo.
exe	Path to an mPTP executable.
outfolder	Path to output folder. Default to NULL. If not specified, a temporary location is used.
minbrlen	Numeric. Branch lengths smaller or equal to the value provided are ignored from computations. Default to 0.0001. Use min_brlenfor fine tuning.
webserver	A .txt file containing mPTP results obtained from a webserver. Default to NULL.

Details

Both gmyc_ci and bgmyc_ci can take a very long time to proccess, depending on how many posterior trees are provided. As an alternative, these analyses can be sped up significantly by running in parallel using plan.

Value

A vector containing the number of species partitions in `tr`, `dna` or `infile` followed by the number of partitions in `posterior`, `reps` or `bootstraps`.

Author(s)

Pedro S. Bittencourt, Rupert A. Collins.

Examples

```
# gmyc confidence intervals

# compute values using multisession mode
{
  try( future::plan("multisession") )

  gmyc_res <- try( gmyc_ci(ape::as.phylo(geophagus_beast), geophagus_posterior) )

  # reset future parameters
  try( future::plan("sequential") )
}

# plot distribution
try(plot(density(gmyc_res)))

# tabulate
try( tibble::tibble(
  method = "gmyc",
  point_estimate = gmyc_res[1],
  CI_95 = as.integer(quantile(gmyc_res[-1], probs = c(0.025, 0.975))) |>
    stringr::str_flatten(collapse = "-"),
  CI_mean = as.integer(mean(gmyc_res[-1])),
  CI_median = as.integer(stats::median(gmyc_res[-1]))
)
)
```

`delim_autoplot` *Plot Phylogenetic Trees With Species Delimitation Partitions*

Description

`delim_autoplot()` returns a phylogenetic tree plotted using `ggtree` alongside with a customized tile plot using geom_tile combined by wrap_plots.

Usage

```
delim_autoplot(
  delim,
  tr,
  consensus = TRUE,
  n_match = NULL,
  delim_order = NULL,
  tbl_labs = NULL,
  col_vec = NULL,
  hexpand = 0.1,
  widths = c(0.5, 0.2)
)
```

Arguments

`delim`	Output from delim_join.
`tr`	A treedata object. Both phylogram and ultrametric trees are supported.
`consensus`	Logical. Should the majority-vote consensus to be estimated?
`n_match`	An Integer. If `consensus = TRUE`, threshold for majority-vote calculations. See delim_consensus for details.
`delim_order`	A character vector of species delimitation names ordered by user. Default to NULL.
`tbl_labs`	A tbl_df of customized labels for tree plotting. The first column must match tip labels of the `tr` object, while the second column should have customized labels.
`col_vec`	A color vector for species delimitation partitions. See delim_brewer for customized color palette options.
`hexpand`	Numeric. Expand xlim of tree by a ratio of x axis range. Useful if tiplabels become truncated when plotting. Default to `0.1`.
`widths`	A numeric vector containing the relative widths of the tree and species delimitation bars. See wrap_plots for details. Defaults to `c(0.5, 0.2)`.

Details

`delim_autoplot()` is a wrapper for tree plotting with associated data implemented using `ggtree`, `ggplot2`, and `patchwork`. If `consensus = TRUE`, a consensus bar will be plotted next to the species delimitation plot, summarizing partitions across samples. If no consensus is reached, an "X" will be plotted instead.

Value

A `patchwork` object.

Author(s)

Pedro S. Bittencourt, Rupert A. Collins.

Examples

```
# view partitions using an ultrametric tree
p <- delim_autoplot(geophagus_delims, geophagus_beast)
p

# view partitions using a phylogram
p1 <- delim_autoplot(geophagus_delims, geophagus_raxml)
```

`delim_autoplot2` *Plot Phylogenetic Trees With Species Delimitation Partitions*

Description

`delim_autoplot2()` returns a phylogenetic tree plotted using `ggtree` alongside with a customized tile plot using geom_tile combined by wrap_plots.

Usage

```
delim_autoplot2(
  delim,
  tr,
  consensus = TRUE,
  n_match = NULL,
  delim_order = NULL,
  tbl_labs,
  species,
  hexpand = 0.1,
  widths = c(0.5, 0.2)
)
```

Arguments

`delim`	Output from delim_join.
`tr`	A treedata object. Both phylogram and ultrametric trees are supported.
`consensus`	Logical. Should the majority-vote consensus to be estimated?
`n_match`	An Integer. If `consensus = TRUE`, threshold for majority-vote calculations. See delim_consensus for details.
`delim_order`	A character vector of species delimitation names ordered by user. Default to NULL.
`tbl_labs`	A tbl_df of customized labels for tree plotting. The first column must match tip labels of the `tr` object, while the second column should have customized labels.
`species`	column name in `tbl_labs` which contains species names for each tip of the tree.
`hexpand`	Numeric. Expand xlim of tree by a ratio of x axis range. Useful if tiplabels become truncated when plotting. Default to `0.1`.
`widths`	A numeric vector containing the relative widths of the tree and species delimitation bars. See wrap_plots for details. Defaults to `c(0.5, 0.2)`.

Details

`delim_autoplot2()` is a wrapper for tree plotting with associated data implemented using `ggtree`, `ggplot2`, and `patchwork`. If `consensus = TRUE`, a consensus bar will be plotted next to the species delimitation plot, summarizing partitions across samples. If no consensus is reached, an "X" will be plotted instead. This function is a modified version of delim_autoplot which plots species partitions using a black and grey color scheme.

Value

A `patchwork` object.

Author(s)

Pedro S. Bittencourt, Rupert A. Collins.

Examples

```
# create labels
labs <- geophagus_info |> dplyr::select(gbAccession, scientificName)

# view partitions using an ultrametric tree
p <- delim_autoplot2(geophagus_delims,
  geophagus_beast,
  tbl_labs = labs,
  species = "scientificName"
)
p

# view partitions using a phylogram
p1 <- delim_autoplot2(geophagus_delims,
  geophagus_raxml,
  tbl_labs = labs,
  species = "scientificName"
)
```

`delim_brewer` *Customize Delimitation Colors*

Description

`delim_brewer()` returns a set of colors created by interpolating or using color palettes from RColorBrewer, viridisLite or randomcoloR.

Usage

```
delim_brewer(delim, package = NULL, palette = NULL, seed = NULL)
```

Arguments

delim	Output from delim_join.
package	Package which contains color palettes. Available options are "RColorBrewer", "viridisLite" or "randomcoloR".
palette	A palette name. brewer.pal for RColorBrewer or viridis for viridisLite options.
seed	Integer. Number to initialize random number generator.

Details

delim_brewer() interpolates over a color palette and returns a vector of random colors whose length is equal to the sum of unique species delimitation partitions in delim. For reproducibility, make sure to provide a seed. If not provided, Sys.time will be used as seed instead. One should also try different seeds to get best color combinations for plotting.

Value

A character vector of hexadecimal color codes.

Author(s)

Rupert A. Collins, Pedro S. Bittencourt

Examples

```
# create a vector of colors
cols <- delim_brewer(geophagus_delims, package = "randomcoloR")
```

---

delim_consensus *Estimate a Majority-Vote Consensus*

Description

delim_consensus() estimates a majority-vote consensus over the output of delim_join in a row-wise manner.

Usage

```
delim_consensus(delim, n_match = NULL)
```

Arguments

delim	Output from delim_join.
n_match	An integer. Threshold for Majority-Vote calculations. If not specified, returns a warning and the threshold will be defined as ceiling(ncol(delim[, -1])/2).

Details

`delim_consensus()` iterates row-by-row, counting the number of matching species partition names across all species delimitations methods in delim_join output. If the sum of identical partition names is greater or equal `n_match`, the consensus column will be filled with its partition name. Otherwise, consensus column will be filled with NA.

Value

an object of class tbl_df.

Author(s)

Pedro S. Bittencourt

Examples

```
# estimate a majority vote consensus
delim_consensus <- delim_consensus(geophagus_delims, n_match= 5)

# check
delim_consensus
```

`delim_join` *Join Multiple Species Delimitation Methods Outputs*

Description

`delim_join()` returns a tbl_df of species delimitation outputs whose partitions are consistent across different methods.

Usage

```
delim_join(delim)
```

Arguments

`delim`	A list or data.frame of multiple species delimitation methods outputs.

Details

`delim_join()` is a helper function to join multiple lists or columns of species delimitation outputs into a single tbl_df while keeping consistent identifications across multiple methods. Species delimitation outputs are in general a list or data frame of sample labels and its species partitions (Species 1, Species 2, etc.). These partition names may be or not the same across two or more methods. `delim_join()` standardizes partition names across two or more species delimitation outputs while keeping its underlying structure intact.

Value

an object of class tbl_df.

Author(s)

Pedro S. Bittencourt, Rupert A. Collins.

Examples

```
## run GMYC
gmyc_res <- try( splits::gmyc(ape::as.phylo(geophagus_beast), method = "single") )

# create a tibble
gmyc_df <- try( gmyc_tbl(gmyc_res) )

## run bGMYC
bgmyc_res <- try( bGMYC::bgmyc.singlephy(ape::as.phylo(geophagus_beast),
  mcmc = 11000,
  burnin = 1000,
  thinning = 100,
  t1 = 2,
  t2 = ape::Ntip(ape::as.phylo(geophagus_beast)),
  start = c(1, 0.5, 50)
)
)
# create a tibble
bgmyc_df <- try( bgmyc_tbl(bgmyc_res, ppcutoff = 0.05) )

## LocMin

# create a distance matrix
mat <- try( ape::dist.dna(geophagus, model = "raw", pairwise.deletion = TRUE) )

# estimate local minima from `mat`
locmin_res <- try( spider::localMinima(mat) )

# create a tibble
locmin_df <- try( locmin_tbl(mat,
  threshold = locmin_res$localMinima[1],
  haps = ape::as.phylo(geophagus_beast)$tip.label
)
)
# join delimitations
all_delims <- try( delim_join(list(gmyc_df, bgmyc_df, locmin_df)) )

# check
try(all_delims)
```

drop_sequences *Remove Sequences of a DNAbin list object*

Description

`drop_sequences()` removes sequences of a FASTA file by its names.

Usage

```
drop_sequences(dna, identifier, drop = TRUE)
```

Arguments

`dna`	a DNAbin list object.
`identifier`	a character vector containing sequence names.
`drop`	Logical. If `TRUE`, sequence names in `identifier` will be dropped from `dna`. If `FALSE`, sequence names absent in `identifier` will be dropped instead.

Details

`drop_sequences()` relies on exact match between sequence names within a fasta file and `identifier` argument.

Value

an object of class DNAbin.

Author(s)

Pedro S. Bittencourt

Examples

```
# Create a vector of sequence names to drop or keep.
identifier <- names(geophagus)[1:3]

# Remove sequences listed in identifier
drop_sequences(geophagus, identifier, drop = TRUE)

# Remove sequences not listed in identifier
drop_sequences(geophagus, identifier, drop = FALSE)
```

dwc_terms *Print Darwin Core Terms, Definitions and Examples as Bullet Lists*

Description

`dwc_terms()` checks a vector or list of terms and return definitions and examples for each one of them.

Usage

```
dwc_terms(dwc, terms)
```

Arguments

`dwc`	a list of standard terms and definitions created using get_dwc.
`terms`	a vector or list of terms to check.

Details

For each term in a vector or list, `dwc_terms` will return a bullet list containing the term, followed by its definition and examples.

Value

a bullet list.

Author(s)

Pedro S. Bittencourt, Rupert A. Collins.

Examples

```
dwc <- get_dwc(type= "simple")
dwc_terms(dwc, c("genus", "scientificName"))
```

geophagus *Cytochrome C Oxidase Sequences of Geophagus Eartheaters*

Description

This is a set of 354 sequences of the mitochondrial gene cytochrome c oxidase subunit I (COI-5P) of the eartheaters of the *Geophagus sensu stricto* species group downloaded from GenBank. Most of these sequences are from the data analysed by Ximenes et al. (2021).

Usage

```
geophagus
```

Format

An object of class DNAbin

Source

Ximenes AM, Bittencourt PS, Machado VN, Hrbek T, Farias IP. 2021. Mapping the hidden diversity of the Geophagus sensu stricto species group (Cichlidae: Geophagini) from the Amazon basin. PeerJ 9:e12443.

---

geophagus_beast *Geophagus Eartheaters Ultrametric Tree*

Description

This is a Maximum Clade Credibility (MCC) tree containing unique haplotypes from geophagus estimated using BEAST2 v2.6.7. Unique haplotypes were select using hap_collapse.

Usage

```
geophagus_beast
```

Format

An object of class treedata.

---

geophagus_bootstraps *Geophagus Eartheaters Bootstrap Trees*

Description

This is a set of 100 Maximum Likelihood trees sampled from bootstrap trees used to estimate geophagus_raxml using RAxML-NG v. 1.1.0-master. Meant to be used for confidence_intervals estimation.

Usage

```
geophagus_bootstraps
```

Format

An object of class multiphylo

geophagus_delims *Geophagus Eartheaters Species Partitions*

Description

This is a data frame containing species delimitation partitions for all the 137 unique haplotypes of geophagus generated using functions contained in this package. Use report_delim to check number of lineages per method.

Usage

```
geophagus_delims
```

Format

A dataframe with 137 rows and 9 columns:

labels Unique haplotype labels

abgd species partitions for ABGD method

asap species partitions for ASAP method

bgmyc species partitions for bGMYC method

gmyc species partitions for GMYC method

locmin species partitions for locmin method

morph species partitions following NCBI taxonomy

mptp species partitions for mPTP method

ptp species partitions for PTP method

geophagus_info *Geophagus Earthearts Associated Metadata*

Description

This is the associated metadata for the 354 sequences of the mitochondrial gene cytochrome c oxidase subunit I (COI-5P) of the *Geophagus sensu stricto* species group downloaded from GenBank and stored in geophagus.

Usage

```
geophagus_info
```

Format

A data frame with 354 rows and 19 columns:

scientificName scientific name

scientificNameGenBank scientific name following NCBI taxonomy

class class

order order

family family

genus genus

dbid NCBI Nucleotide Database internal ID

gbAccession NCBI Nucleotide Database accession number

gene Gene acronym

length Sequence length in base pairs (bp)

organelle Organelle from which gene was sequenced

catalogNumber An identifier for the record within a data set or collection

country Name of the Country followed by sampling locality (when available)

publishedAs Title of the article which generated the sequences

publishedIn Journal which published the article

publishedBy A person, group, or organization responsible for depositing the sequence

date Date published

decimalLatitude Latitude in decimal degrees

decimalLongitude Longitude in decimal degrees

---

geophagus_posterior *Geophagus Eartheaters Posterior Trees*

Description

This is a set of 100 ultrametric trees sampled from the posterior trees used to estimate geophagus_beast using BEAST2 v2.6.7. Meant to be used for confidence_intervals estimation.

Usage

```
geophagus_posterior
```

Format

An object of class multiphylo

geophagus_raxml *Geophagus Eartheaters Phylogram*

Description

This is a Maximum Likelihood Estimation Tree containing unique haplotypes from geophagus estimated using RAxML-NG v. 1.1.0-master. Unique haplotypes were select using hap_collapse.

Usage

```
geophagus_raxml
```

Format

An object of class treedata.

get_delim_cols *Extract Labels and Colors from Species Delimitation Partitions*

Description

`get_delim_cols()` returns a tbl_df format containing extracted and processed data from delim_autoplot.

Usage

```
get_delim_cols(p, delimname = NULL, hap_tbl = NULL)
```

Arguments

`p`	Output from delim_autoplot.
`delimname`	A character vector of species delimitation names (optional). If provided, the function filters the data to only include rows matching such terms. Default to NULL.
`hap_tbl`	output from haplotype_tbl (optional). If provided, the function will annotate color and fill data for collapsed haplotypes. Default to NULL.

Details

`get_delim_cols()` is a convenience function to extract labels, species partitions, color and fill data from the output of delim_autoplot in a tbl_df format. It is best used when combined with haplotype information from haplotype_tbl or when combined with other metadata, such as GPS coordinates for map plotting.

Value

an object of class tbl_df.

Author(s)

Pedro S. Bittencourt.

Examples

```
# plot using autoplot
p <- delim_autoplot(geophagus_delims, geophagus_beast)

# view
p

# get haplotypes
hap_tbl <- haplotype_tbl(geophagus)

# extract colors for consensus
get_delim_cols(p, delimname= "consensus", hap_tbl= hap_tbl)
```

---

get_dwc *Get Darwin Core Terms and Definitions*

Description

`get_dwc()` returns a list of standardized terms and definitions used by the Darwin Core Maintenance Interest Group https://dwc.tdwg.org/.

Usage

```
get_dwc(type)
```

Arguments

`type`	Which type of distribution files to download. Available options are: • simple Simple Darwin Core Terms. • all All Darwin Core Terms.

Details

`get_dwc()` reads Darwin Core distribution documents and terms from Github repository https://github.com/tdwg/dwc directly into `Environment`. This function will return a list containing the most recent accepted terms as a vector and a tbl_df containing terms, definitions, examples and details about each one of them.

Value

a list.

Author(s)

Pedro S. Bittencourt, Rupert A. Collins

Examples

```
dwc <- get_dwc(type= "simple")
```

`gmyc_tbl` *Turns GMYC Results Into a Tibble*

Description

`gmyc_tbl()` processes output from gmyc into an object of class `tbl_df`.

Usage

```
gmyc_tbl(gmyc_res, delimname = "gmyc")
```

Arguments

`gmyc_res`	Output from gmyc.
`delimname`	Character. String to rename the delimitation method in the table. Default to 'gmyc'.

Details

`splits` package uses gmyc to optimize genetic clusters and spec.list to cluster samples into species partitions. `gmyc_tbl()` turns these results into a tibble which matches the output from bgmyc_tbl and locmin_tbl.

Value

An object of class tbl_df.

Author(s)

Thomas Ezard, Tomochika Fujisawa, Tim Barraclough.

Source

Pons J., Barraclough T. G., Gomez-Zurita J., Cardoso A., Duran D. P., Hazell S., Kamoun S., Sumlin W. D., Vogler A. P. 2006. Sequence-based species delimitation for the DNA taxonomy of undescribed insects. Systematic Biology. 55:595-609.

Monaghan M. T., Wild R., Elliot M., Fujisawa T., Balke M., Inward D. J. G., Lees D. C., Ranaivosolo R., Eggleton P., Barraclough T. G., Vogler A. P. 2009. Accelerated species inventory on Madagascar using coalescent-based models of species delineation. Systematic Biology. 58:298-311.

Fujisawa T., Barraclough T. G. 2013. Delimiting Species Using Single-Locus Data and the Generalized Mixed Yule Coalescent Approach: A Revised Method and Evaluation on Simulated Data Sets. Systematic Biology. 62(5):707–724.

Examples

```
# run GMYC
gmyc_res <- try( splits::gmyc(ape::as.phylo(geophagus_beast)) )

# create a tibble
gmyc_df <- try( gmyc_tbl(gmyc_res) )

# check
try(gmyc_df)
```

`haplotype_tbl` *Summarise Haplotypes Down to One Row*

Description

`haplotype_tbl()` returns a tbl_df summarising all unique haplotype frequencies and duplicates into a single row.

Usage

```
haplotype_tbl(dna, clean = TRUE, collapseSubstrings = TRUE, verbose = TRUE)
```

Arguments

`dna`	an object of class DNAbin.
`clean`	logical. Whether to remove or not remove non ACTG bases from alignment.
`collapseSubstrings`	logical. Whether to collapse or not collapse shorter but identical sequences.
`verbose`	logical. Returns a warning if any sequence contains non ACTG bases. See clean_dna for details.

Details

`haplotype_tbl()` uses a combination of clean_dna and hap_collapse to summarise haplotypes into a tibble. Each row of the tibble has an unique haplotype, its frequency and all its collapsed duplicates in a flattened string.

Value

an object of class tbl_df.

Author(s)

Rupert A. Collins, Pedro S. Bittencourt.

Examples

```
# get haplotype table
haplotype_tbl(geophagus)
```

---

hap_collapse *Removes Duplicated Sequences from Alignment*

Description

`hap_collapse()` collapses haplotypes from a DNAbin object, keeping unique haplotypes only.

Usage

```
hap_collapse(dna, clean = TRUE, collapseSubstrings = TRUE, verbose = TRUE)
```

Arguments

`dna`	A DNAbin object.
`clean`	logical. Whether to remove or not remove non ACTG bases from alignment.
`collapseSubstrings`	logical. Whether to collapse or not collapse shorter but identical sequences.
`verbose`	logical. Returns a warning if any sequence contains non ACTG bases. See clean_dna for details.

Details

`hap_collapse()` collapses a DNAbin object, keeping unique haplotypes only. If `clean = TRUE`, the function will call clean_dna to remove any non ACTG bases from alignment prior to collapsing haplotypes. If `clean = FALSE`, the function will treat data as it is, and will not remove any bases. If `collapseSubstrings = TRUE`, the function will consider shorter but identical sequences as the same haplotype and collapse them, returning the longest sequence. If `collapseSubstrings = FALSE`, the function will consider shorter but identical sequences as different haplotypes and will keep them.

Value

A DNAbin object.

Author(s)

Rupert A. Collins

Examples

```
# collapse into unique haplotypes, including shorter sequences
hap_collapse(geophagus, clean = TRUE, collapseSubstrings = TRUE)

# collapse into unique haplotypes keeping shorter sequences
hap_collapse(geophagus, clean = TRUE, collapseSubstrings = FALSE)
```

`hap_unite` *Unite Haplotype Summaries with Species Delimitation Outputs*

Description

`hap_unite()` returns a single tbl_df combining all results from haplotype_tbl or collapse_others with results from delim_join or delim_consensus.

Usage

```
hap_unite(hap_tbl, delim)
```

Arguments

`hap_tbl`	output from haplotype_tbl or collapse_others.
`delim`	output from delim_join or delim_consensus.

Details

Many functions in this package relies on the usage of unique haplotypes due to known issues when using identical or duplicated sequences for species delimitation analysis. Thus, these outputs will very often refer only to unique haplotypes within a given dataset, which can be determined by using functions like hap_collapse. Assuming that a duplicated or identical sequence should share the same properties as the first sequence of the group has, `hap_unite()` combines the output of haplotype_tbl with the output of delim_join. Alternativelly, one may use collapse_others and delim_consensus as well. This output may be used for downstream analysis or to determine in which cluster a given sequence belongs.

Value

an object of class tbl_df.

Author(s)

Pedro S. Bittencourt

Examples

```
# get haplotype table
hap_tbl <- haplotype_tbl(geophagus)

# unite
hap_unite(hap_tbl, geophagus_delims)
```

`locmin_tbl` *Turns Local Minima Results into a Tibble*

Description

`locmin_tbl()` processes output from tclust into an object of class tbl_df.

Usage

```
locmin_tbl(distobj, threshold = 0.01, haps = NULL, delimname = "locmin")
```

Arguments

`distobj`	A distance object (usually from dist.dna).
`threshold`	Distance cutoff for clustering. Default of 0.01. See localMinima for details.
`haps`	Optional. A vector of haplotypes to keep into the tbl_df.
`delimname`	Character. String to rename the delimitation method in the table. Default to 'locmin'.

Details

`spider` package uses localMinima to determine possible thresholds for any distance matrix and tclust to cluster samples within a given `threshold` into species partitions. `locmin_tbl()` turns these inputs into a tibble which matches the output from gmyc_tbl and bgmyc_tbl.

Value

An object of class tbl_df.

Author(s)

Samuel Brown.

Source

Brown S.D.J., Collins R.A., Boyer S., Lefort M.-C., Malumbres-Olarte J., Vink C.J., Cruickshank, R.H. 2012. Spider: An R package for the analysis of species identity and evolution, with particular reference to DNA barcoding. Molecular Ecology Resources, 12: 562-565.

Examples

```
# create a distance matrix
mat <- ape::dist.dna(geophagus, model = "raw", pairwise.deletion = TRUE)

# run Local Minima
locmin_res <- spider::localMinima(mat)

# create a tibble
locmin_df <- locmin_tbl(mat,
                        threshold = locmin_res$localMinima[1],
                        haps = ape::as.phylo(geophagus_beast)$tip.label)

# check
locmin_df
```

`match_ratio` *Compute Agreement Between Alternative Species Delimitation Partitions*

Description

`match_ratio()` uses the Match Ratio statistic of Ahrens et al. (2014) to compute agreement between all pairs of species delimitation partitions in delim_join output.

Usage

```
match_ratio(delim)
```

Arguments

`delim`	Output from delim_join.

Details

`match_ratio()` iterates between all species delimitation partitions in delim_join output and returns a tbl_df containing the following columns:

- `pairs` pairs of species delimitation methods analyzed.
- `delim_1` number of species partitions in method 1.
- `delim_2` number of species partitions in method 2.
- `n_match` number of identical species partitions in methods 1 and 2.
- `match_ratio` match ratio statistic, where 0 indicates no agreement between pairs of species delimitation partitions and 1 indicates complete agreement between them.

Value

an object of class tbl_df.

Author(s)

Pedro S. Bittencourt

Source

Ahrens D., Fujisawa T., Krammer H. J., Eberle J., Fabrizi S., Vogler A. P. 2016. Rarity and Incomplete Sampling in DNA-Based Species Delimitation. Systematic Biology 65 (3): 478-494.

Examples

```
# estimate match ratio statistics
match_ratio(geophagus_delims)
```

`min_brlen` *A function to report the smallest tip-to-tip distances in a phylogenetic tree*

Description

`min_brlen()` returns a table of smallest tip-to-tip distances in a phylogenetic tree.

Usage

```
min_brlen(tree, n = 5, verbose = TRUE)
```

Arguments

`tree`	A path to tree file in Newick format, or a phylogenetic tree object of class phylo.
`n`	Number of distances to report (default = 5).
`verbose`	Logical of whether to print the result to screen (default = TRUE).

Details

`min_brlen()` tabulates the smallest tip-to-tip distances in a phylogenetic tree using cophenetic.phylo and prints a table to screen. This is useful when excluding identical or near-identical haplotypes using the '–minbr' parameter in mPTP.

Value

an object of class tbl_df

Author(s)

Rupert A. Collins

Examples

```
# estimate minimum branch length from raxml tree
min_brlen(ape::as.phylo(geophagus_raxml), n = 5)
```

`morph_tbl` *Generating a Morphological Delimitation Table*

Description

`morph_tbl()` returns species partition hypothesis estimated from a prior taxonomic identifications supplied by the user.

Usage

```
morph_tbl(labels, sppVector, delimname = "morph")
```

Arguments

`labels`	Vector of unique sequence ID labels.
`sppVector`	Vector of corresponding morphological species delimitation groups.
`delimname`	Character. String to rename the delimitation method in the table. Default to 'morph'.

Details

`morph_tbl()` uses information in a species name vector to label each unique sample with a number corresponding to this name.

Value

an object of class tbl_df.

Author(s)

Rupert A. Collins

Examples

```
# create a tibble
morph_df <- morph_tbl(
  labels = geophagus_info$gbAccession,
  sppVector = geophagus_info$scientificName
)

# check
morph_df
```

mptp_tbl *A Command-Line Interface for mPTP - multi-rate Poisson Tree Processes*

Description

`mptp_tbl()` returns species partition hypothesis estimated by mPTP software https://github.com/Pas-Kapli/mptp.

Usage

```
mptp_tbl(
  infile,
  exe = NULL,
  outfolder = NULL,
  method = c("multi", "single"),
  minbrlen = 1e-04,
  webserver = NULL,
  delimname = "mptp"
)
```

Arguments

`infile`	Path to tree file in Newick format. Should be dichotomous and rooted.
`exe`	Path to an mPTP executable.
`outfolder`	Path to output folder. Default to NULL. If not specified, a temporary location is used.
`method`	Which algorithm for Maximum Likelihood point-estimate to be used. Available options are: • single Single-rate PTP model. It assumes that every species evolved with the same rate. • multi Multi-rate mPTP model. It assumes that all species have different evolutionary rates.
`minbrlen`	Numeric. Branch lengths smaller or equal to the value provided are ignored from computations. Default to 0.0001. Use min_brlenfor fine tuning.
`webserver`	A .txt file containing mPTP results obtained from a webserver. Default to NULL.
`delimname`	Character. String to rename the delimitation method in the table. Default to 'mptp'.

Details

`mptp_tbl()` relies on system to invoke mPTP software through a command-line interface. Hence, you must have the software available as an executable file on your system in order to use this function properly. `mptp_tbl()` saves all output files in `outfolder` and imports the results generated

to Environment. If an outfolder is not provided by the user, then a temporary location is used. Alternatively, mptp_tbl() can parse a file obtained from webserver such as https://mptp.h-its.org/.

Value

an object of class tbl_df

Author(s)

Paschalia Kapli, Sarah Lutteropp, Jiajie Zhang, Kassian Kobert, Pavlos Pavlides, Alexandros Stamatakis, Tomáš Flouri.

Source

Kapli T., Lutteropp S., Zhang J., Kobert K., Pavlidis P., Stamatakis A., Flouri T. 2016. Multi-rate Poisson tree processes for single-locus species delimitation under maximum likelihood and Markov chain Monte Carlo. Bioinformatics 33(11):1630-1638.

Examples

```
# get path to phylogram
path_to_file <- system.file("extdata/geophagus_raxml.nwk", package = "delimtools")

# run mPTP in single threshold mode (PTP)
ptp_df <- try( mptp_tbl(
  infile = path_to_file,
  exe = "/usr/local/bin/mptp",
  method = "single",
  minbrlen = 0.0001,
  delimname = "ptp",
  outfolder = NULL
)
)
# check
ptp_df

# run mPTP in multi threshold mode (mPTP)

mptp_df <- try( mptp_tbl(
  infile = path_to_file,
  exe = "/usr/local/bin/mptp",
  method = "single",
  minbrlen = 0.0001,
  delimname = "mptp",
  outfolder = NULL
)
)
# check
try(mptp_df)
```

`report_delim` *Report Unique Species Partitions*

Description

`report_delim()` reports the number of unique species partitions in `delim`.

Usage

```
report_delim(delim, verbose = TRUE)
```

Arguments

`delim`	Output from any `*_tbl()` (e.g. gmyc_tbl), delim_join or delim_consensus.
`verbose`	Logical. If TRUE, returns a message and a tabulated summary of `delim`.

Details

For each column in `delim`, `report_delim()` will calculate the number of unique partitions and print them to `Console`. If `delim` is an output from `*_tbl()`, `report_delim()` will get unique species partitions using vec_unique_count. If `delim` is an output from delim_join or delim_consensus, values are summarized by using n_distinct with `na.rm = TRUE`. This is to prevent any columns with NA values to be interpreted as species partitions.

Value

an object of class tbl_df].

Author(s)

Rupert A. Collins, Pedro S. Bittencourt

Examples

```
# report geophagus delimitations
report_delim(geophagus_delims)
```

Index