Package 'traitdataform'

Type Package

Title Formatting and Harmonizing Ecological Trait-Data

Version 0.5.7

Date 2020-04-04

Description Assistance for handling ecological trait data and applying the
Ecological Trait-Data Standard terminology (Schneider et al. 2019
<doi:10.1111/2041-210X.13288>). There are two major use cases: (1) preparation of
own trait datasets for upload into public data bases, and (2) harmonizing
trait datasets from different sources by re-formatting them into a unified
format. See 'traitdataform' website for full documentation.

License MIT + file LICENSE

LazyData TRUE

Depends R (>= 2.10)

Imports data.table, reshape2, taxize, getPass, units, XML, RCurl, plyr

URL https://EcologicalTraitData.github.io/traitdataform,
https://github.com/EcologicalTraitData/traitdataform

BugReports https://github.com/EcologicalTraitData/traitdataform/issues

VignetteBuilder knitr

RoxygenNote 7.1.0

Encoding UTF-8

Suggests knitr, rmarkdown, testthat

NeedsCompilation no

Author Florian D. Schneider [aut, cre]
(<https://orcid.org/0000-0002-1494-5684>),
Caterina Penone [ctb] (<https://orcid.org/0000-0002-8170-6659>),
Andreas Ostrowski [ctb] (<https://orcid.org/0000-0002-2033-779X>),
Dennis Heimann [ctb],
Felix Neff [ctb],
Markus J. Ankenbrand [ctb]

Maintainer Florian D. Schneider <florian.dirk.schneider@gmail.com>

R topics documented:

---

amniota   *Amniote life-history traits*

Description

An amniote life-history database to perform comparative analyses with birds, mammals, and reptiles, Ecological Archives E096-269

Details

Studying life-history traits within and across taxonomic classifications has revealed many interesting and important patterns, but this approach to life history requires access to large compilations of data containing many different life-history parameters. Currently, life-history data for amniotes (birds, mammals, and reptiles) is split among a variety of publicly available databases, data tables embedded in individual papers and books, and species-specific studies by experts. Using data from this wide range of sources is a challenge for conducting macroecological studies because of a lack of standardization in taxonomic classifications, parameter values, and even in which parameters are reported. In order to facilitate comparative analyses between amniote life-history data,

we created a database compiled from peer-reviewed studies on individual species, macroecological studies of multiple species, existing life-history databases, and other aggregated sources as well as published books and other compilations. First, we extracted and aggregated the raw data from the aforementioned sources. Next, we resolved spelling errors and other formatting inconsistencies in species names through a number of computational and manual methods. Once this was completed, subspecies-level data and species-level data were shared via a data-sharing algorithm to accommodate the variety of species transformations (taxonomic promotions, demotions, merges, divergences, etc.) that have occurred over time. Finally, in species where multiple raw data points were identified for a given parameter, we report the median value. Here, we report a normalized and consolidated database of up to 29 life-history parameters, containing at least one life-history parameter for 21 322 species of birds, mammals, and reptiles.

Author(s)

Nathan P. Myhrvold, Elita Baldridge, Benjamin Chan, Dhileep Sivam, Daniel L. Freeman, and S. K. Morgan Ernest

Source

Cite this dataset as

- P. Myhrvold, Nathan; Baldridge, Elita; Chan, Benjamin; Sivam, Dhileep; L. Freeman, Daniel; Ernest, S. K. Morgan (2016): An amniote life-history database to perform comparative analyses with birds, mammals, and reptiles. http://esapubs.org/archive/ecol/E096/269/

See Also

Other rawdata: `amphibio`, `arthropodtraits`, `carabids`, `heteroptera_raw`, `mammaldiet`, `pantheria`, `pulldata()`

| amphibio | *AmphiBIO, a global database for amphibian ecological traits* |

Description

A comprehensive database of natural history traits for amphibians worldwide.

Details

Current ecological and evolutionary research are increasingly moving from species- to trait-based approaches because traits provide a stronger link to organism's function and fitness. Trait databases covering a large number of species are becoming available, but such data remains scarce for certain groups. Amphibians are among the most diverse vertebrate groups on Earth, and constitute an abundant component of major terrestrial and freshwater ecosystems. They are also facing rapid population declines worldwide, which is likely to affect trait composition in local communities,

thereby impacting ecosystem processes and services. In this context, we introduce AmphiBIO, a comprehensive database of natural history traits for amphibians worldwide. The database releases information on 17 traits related to ecology, morphology and reproduction features of amphibians. We compiled data from more than 1,500 literature sources, and for more than 6,500 species of all orders (Anura, Caudata and Gymnophiona), 61 families and 531 genera. This database has the potential to allow unprecedented large-scale analyses in ecology, evolution and conservation of amphibians.

Author(s)

Brunno Freire Oliveira, Vinícius Avelar São-Pedro, Georgina Santos-Barrera, Caterina Penone, and Gabriel C. Costa

Source

Cite as:

- Oliveira, B.F., São-Pedro, V.A., Santos-Barrera, G., Penone, C. & Costa, G.C. (2017). AmphiBIO, a global database for amphibian ecological traits. Scientific Data, 4:170123. doi: 10.1038/sdata.2017.123

Please also cite the data repository on figshare:

- Oliveira, Brunno Freire; São-Pedro, Vinícius Avelar; Santos-Barrera, Georgina; Penone, Caterina; C. Costa, Gabriel (2017): AmphiBIO_v1. figshare. https://doi.org/10.6084/m9.figshare.4644424.v5

Creative Commons BY 4.0. You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

See Also

Other rawdata: amniota, arthropodtraits, carabids, heteroptera_raw, mammaldiet, pantheria, pulldata()

---

arthropodtraits	*Functional Arthropod Traits*

Description

Data from: A summary of eight traits of Coleoptera, Hemiptera, Orthoptera and Araneae, occurring in grasslands in Germany.

Details

When using this data, please cite the original publication:

- Gossner MM, Simons NK, Achtziger R, Blick T, Dorow WHO, Dziock F, Köhler F, Rabitsch W, Weisser WW (2015) A summary of eight traits of Coleoptera, Hemiptera, Orthoptera and Araneae, occurring in grasslands in Germany. Scientific Data 2: 150013. http://dx.doi.org/10.1038/sdata.2015.13

Additionally, please cite the Dryad data package:

- Gossner MM, Simons NK, Achtziger R, Blick T, Dorow WHO, Dziock F, Köhler F, Rabitsch W, Weisser WW (2015) Data from: A summary of eight traits of Coleoptera, Hemiptera, Orthoptera and Araneae, occurring in grasslands in Germany. Dryad Digital Repository. http://dx.doi.org/10.5061/dryad.53ds2

Author(s)

Gossner MM, Simons NK, Achtziger R, Blick T, Dorow WHO, Dziock F, Köhler F, Rabitsch W, Weisser WW

Source

http://dx.doi.org/10.5061/dryad.53ds2; Creative Commons 0. To the extent possible under law, the authors have waived all copyright and related or neighboring rights to this data.

See Also

Other rawdata: `amniota`, `amphibio`, `carabids`, `heteroptera_raw`, `mammaldiet`, `pantheria`, `pulldata()`

---

as.metadata   *Create metadata for trait dataset*

Description

Function to create a defined object containing metadata paramaters according to a pre-defined template.

Usage

```
as.metadata(
  ...,
  template = list(datasetID = NULL, datasetName = NULL, author = NULL, rightsHolder =
    NULL, bibliographicCitation = NULL, license = NULL, version = NULL, comments = NULL,
      description = NULL, region = NULL)
)
```

Arguments

`...`	named objects of any class. Names must be the same as given by template or they will be ignored when producing the metadata object.
`template`	The default template contains the elements as defined by the Ecological Trait-data Standard. (see Details)

Value

an object of class 'metadata', or - if metadata objects are provided as input - a list object of class 'metadatalist'.

Examples

```
# set metadata object
a <- as.metadata(author = "Martin Gossner", datasetName = "Heteroptera morphometric traits")

# update an existing metadata object
a1 <- as.metadata(datasetID = "heteroptera", template = a)
```

---

as.thesaurus	*Create a list of trait definitions (a thesaurus of traits)*

Description

Create a trait thesaurus object for use as a reference object/lookup table of traits within function `standardize.traits()`.

Usage

```
as.thesaurus(..., replace = NULL)
```

Arguments

`...`	multiple objects of class 'trait' (produced by function `as.trait()`) or a data.frame containing columns according to the terms provided by https://ecologicaltraitdata.github.io/ETS/#terms-for-thesauri.
`replace`	named character vector, with new names as values, and old names as names.

Details

the object class 'trait' comprises necessary information to map a trait name to a trait definition, a target unit and a globally unique identifier. The thesaurus will be used in function 'standardize.traits()' to apply unit conversion and factor level harmonization.

Value

a list of formalized objects of class 'trait', as returned by function as.trait().

Examples

```
# provide traitlist by defining individual traits using function `as.trait()`:

traitlist <- as.thesaurus(body_length = as.trait("body_length", expectedUnit = "mm",
    valueType = "numeric",
    identifier = "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Body_length"),
  antenna_length = as.trait("antenna_length", expectedUnit = "mm",
    valueType = "numeric",
    identifier = "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Antenna_length"),
  metafemur_length = as.trait("metafemur_length", expectedUnit = "mm",
    valueType = "numeric",
    identifier = "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Femur_length")
)

# provide traitlist from data frame object:

as.thesaurus(data.frame(
  trait = c("body_length", "antenna_length", "metafemur_length", "eyewidth_corr"),
  expectedUnit = "mm",
  valueType = "numeric",
  traitDescription = c("body length in mm", "length of antenna in mm",
                           "length of metafemur in mm", "eye width in mm"),
  identifier = c("http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Body_length",
    "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Antenna_length",
    "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Femur_length",
    "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Eye_diameter"
    )
)
)

# provide traitlist from remote archive with renaming
# (pulls from https://github.com/EcologicalTraitData/TraitDataList)

## Not run:
traits1 <- as.thesaurus(read.csv("https://git.io/fpsj5"),
                  replace = c(traitID = "identifier",
                                    traitName = "trait",
                                    traitUnit = "expectedUnit",
                                    Comments = "comments")
)

## End(Not run)
```

---

as.trait   *Create a trait definition*

Description

Creating an object containing a standardised trait definition according to the Ecological Traitdata Standard. Parameters correspond to the definition at http://ecologicaltraitdata.github.io/ETS/#terms-for-thesauri.

Usage

```
as.trait(
  trait,
  identifier = NA,
  broaderTerm = NA,
  narrowerTerm = NA,
  relatedTerm = NA,
  valueType = NA,
  expectedUnit = NA,
  factorLevels = NA,
  maxAllowedValue = NA,
  minAllowedValue = NA,
  traitDescription = NA,
  comments = NA,
  source = NA,
  version = NA,
  author = NA,
  ...
)
```

Arguments

trait	A character string, providing an intuitive, human-readable trait name.
identifier	Unique identifier for the trait, ideally unique and stable URI which identify the source of the trait definition.
broaderTerm	One or several terms that enclose the trait definition.
narrowerTerm	One or several terms that are enclosed by the trait definition.
relatedTerm	One or several terms that are related to this term (ideally given as URI).
valueType	the type of trait values. Possible entries are 'numeric', 'integer', 'categorical', 'logical', or 'character'.
expectedUnit	the unit expected for measurement entries.
factorLevels	A comma separated list of terms comprising the constrained vocabulary for categorical traits or ordinal binary traits.
maxAllowedValue	
	An upper boundary for accepted numerical values.
minAllowedValue	
	A lower boundary for accepted numerical values.
traitDescription	
	A short, unambiguous definition of the trait. May refer to a method of measurement.

comments	Details and Examples for clarification of the trait definition.
source	A character string providing a full bibliographic reference to the trait definition (giving title, author, year and publication).
version	A character string containing the version number of the referenced definition (e.g. "v1.2"), if applicable.
author	A character string or object of class 'person' (as created by as.person()) attributing the author(s) of the trait definition.
...	other arguments, passed on to print function.

Examples

```
body_length <- as.trait("body_length", expectedUnit = "mm", valueType = "numeric",
  identifier = "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Body_length",
  traitDescription = "The known longest dimension of the physical structure of organisms",
  relationSource = "Maggenti and Maggenti, 2005",
  broaderTerm = "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Body_dimension"
  )
```

---

as.traitdata	*Standardize format of traitdata*

Description

Turns wide-table formats (species-traits matrix and occurrence table) into long-table format. As input, the function requires information about which columns contain traits, given as a list of trait-names, and which column contains the taxon name. For tables containing repeated measurements of traits within the same taxon, an occurrenceID should be given or will be created.

Usage

```
as.traitdata(
  x,
  traits = attributes(x)$traits,
  taxa = attributes(x)$taxa,
  occurrences = attributes(x)$occurrences,
  datasetID = attributes(x)$datasetID,
  measurements = attributes(x)$measurements,
  units = attributes(x)$units,
  keep = attributes(x)$keep,
  drop = attributes(x)$drop,
  na.rm = TRUE,
  id.vars = names(x)[names(x) %in% keep & !names(x) %in% drop],
  thesaurus = attributes(x)$thesaurus,
  metadata = attributes(x)$metadata,
  longtable = TRUE,
```

```
   ...
 )
```

Arguments

x	data.frame object, containing at least a column of taxa, and one or more columns of trait measurements.
traits	a vector of column names containing traits.
taxa	the name of the column containing taxon names.
occurrences	either a column name containing identifiers for each individual specimen on which several traits were measured, i.e. an occurrence of this taxon, or a vector of occurrence identifiers which must be of the same length as the number of rows of the table. See 'Details'.
datasetID	a unique name for this dataset (optional). Will be prepended to the occurrence ID and measurement ID.
measurements	either a column name containing identifiers for each individual measurement, or a vector of measurement identifiers. This applies, if single trait measurements span across multiple columns of data, e.g. multivariate traits like quantitative measures of chemical compounds, wavelengths or x-y-z coordinates. In most cases, a measurementID will link the data across rows in the longtable format. Make sure that the traitnames given reflect the different dimensions of the trait measurement. If `measurement` remains blank, sequential identifiers will be auto-generated for each measured value.
units	a single character string or named vector giving the units that apply to the traits. If only one unit type is given, it will be applied to all traits.
keep	a vector or named vector containing the names of the input columns to be kept in the output. Vector names will be used to rename the columns. It is recommended to use accepted column names of the traitdata standard for renaming!
drop	a vector acting as the inverse of `keep`. All columns listed will be removed from the output dataset.
na.rm	logical defaults to `TRUE`. If `FALSE`, all measured Values containing NA will be kept in the output table. This is not reccomended for most data.
id.vars	a vector of column names to return. Autogenerated from input column names and 'keep' and 'drop'.
thesaurus	an object of class 'thesaurus' as created by function `as.thesaurus()`. If provided, this will superimpose trait names provided in argument `traits`. The thesaurus will be appended as an attribute and can be revisited by calling `attributes(x)$thesaurus`.
metadata	a list of class metadata, as created by function `as.metadata()`. Metadata will be added as attributes to the data table. Possible parameters to the function call are: `rightsHolder`, `bibliographicCitation`, `license`, `author`, `datasetID`, `datasetName`, `version`. (see 'Details')
longtable	logical, defaults to `TRUE`. If `FALSE`, data will not be converted into lontable format, but remain in widetable format as provided. Note that any columns not indicated in arguments `traits`, `keep`, `units`, `taxa`, `occurrences` will be dropped from the output.
...	other arguments, passed on to print function.

Details

If occurrences is left blank, the script will check for the structure of the input table. If several entries are given for the same taxon, it assumes that input is an occurrence table, i.e. with multiple observations of a single taxon, and assigns identifiers.

Metadata will be stored as attributes to the data frame and can be accessed via `attributes()`. It is not necessary but highly recommended to provide metadata when working with multiple trait data files. When appending datasets using `rbind()`, the metadata information will be added as additional columns and dataset attribution will be listed in attributes.

Examples

```
## Not run:
# species-trait matrix:

pulldata("carabids")

dataset1 <- as.traitdata(carabids,
  taxa = "name_correct",
  traits = c("body_length", "antenna_length", "metafemur_length"),
  units = "mm",
  keep = c(basisOfRecordDecription = "source_measurement", measurementRemark = "note")
  )

# occurrence table:

pulldata("heteroptera_raw")

dataset2 <- as.traitdata(heteroptera_raw,
  taxa = "SpeciesID",
  traits = c("Body_length", "Body_width", "Body_height", "Thorax_length",
    "Thorax_width", "Head_width", "Eye_width", "Antenna_Seg1", "Antenna_Seg2",
    "Antenna_Seg3", "Antenna_Seg4", "Antenna_Seg5", "Front.Tibia_length",
    "Mid.Tibia_length", "Hind.Tibia_length", "Front.Femur_length",
    "Hind.Femur_length", "Front.Femur_width", "Hind.Femur_width",
    "Rostrum_length", "Rostrum_width", "Wing_length", "Wing_widt"),
  units = "mm",
  keep = c(sex = "Sex", references = "Source", lifestage = "Wing_development"),
  metadata = as.metadata(
    author = "Gossner MM, Simons NK, HÃ¶ck L and Weisser WW",
    datasetName = "Morphometric traits Heteroptera",
    bibliographicCitation = attributes(heteroptera_raw)$citeAs,
    license = "http://creativecommons.org/publicdomain/zero/1.0/"
    )
)

## End(Not run)
```

---

carabids	*Carabid morphological traits*

Description

Average body measures of 120 Carabid species occuring in the Netherlands.

Format

A data frame containing following columns:

- name_correct = species name
- source_measurement = researcher who performed measurement;
- body_length = body length in mm;
- antenna_length = antenna length in mm;
- metafemur_length = length metafemur in mm;
- eyewidth_corr = eye width in mm;
- note = note;
- resid_femur = residual femur length in mm (i.e. residual from linear model in which femur length is explained by body length);
- resid_eye = residual eye length in mm (i.e. residual from linear model in which eye length is explained by body length)
- resid_antenna = residual antenna length in mm (i.e. residual from linear model in which antenna length is explained by body length)

Details

When using this data, please cite the original publication:

- van der Plas F, van Klink R, Manning P, Olff H, Fischer M (2017) Sensitivity of functional diversity metrics to sampling intensity. Methods in Ecology and Evolution 8(9): 1072-1080. https://doi.org/10.1111/2041-210x.12728

Additionally, please cite the Dryad data package:

- van der Plas F, van Klink R, Manning P, Olff H, Fischer M (2017) Data from: Sensitivity of functional diversity metrics to sampling intensity. Dryad Digital Repository. https://doi.org/10.5061/dryad.1fn46

Author(s)

Fons van der Plas, R. van Klink, P. Manning, H. Olff, M. Fischer

Source

http://dx.doi.org/10.5061/dryad.53ds2; Creative Commons 0. To the extent possible under law, the authors have waived all copyright and related or neighboring rights to this data.

See Also

Other rawdata: [amniota](), [amphibio](), [arthropodtraits](), [heteroptera_raw](), [mammaldiet](), [pantheria](), [pulldata]()

---

cast.traitdata	*Cast long-table trait data into wide-table format*

Description

Function to reformat trait data from the long-table into a matrix/wide-table or occurrence table format.

Usage

```
cast.traitdata(
  .data,
  values = "traitValue",
  traits = "traitName",
  units = "traitUnit",
  fun.aggregate = NULL
)
```

Arguments

`.data`	dataset of class 'traitdata' to be cast into wide-table format.
`values`	the column name containing the trait values to be used to fill the matrix (default is `traitValue`). Duplicate columns (e.g. `traitValueStd`) will be omitted. See notes.
`traits`	the column name to be kept for parsing into wide-table (default is `traitName`). Note that any duplicate column that contains trait names, e.g. `traitNameStd` will be omitted.
`units`	the column name containing the units of numerical values (default is `traitUnit`).
`fun.aggregate`	option for [reshape2::cast()]() to define method of aggregation.

Details

The wide-table will be composed while preserving the detail given in the dataset (occurrence level or taxa level). The cells will be filled with the values from 'traitValue'.

If taxa should be summarized, provide function for summarizing in parameter 'summarize'. This can be any function that takes a vector and returns a single value for **both numerical and factorial/character input vectors!**. The default for numerical values is to return an arithmetric mean while including any outliers. For factorial values, the value is accepted if it is equal in all of the input entries. Otherwise NA is returned. For heterogeneous factorial or character input, user action is required for homogenizing the data before calling `cast.traitdata()`.

Value

a wide-table data.frame object containing all taxa (and other differentiating parameters) in rows and all traits (extracted from column 'traitName') in columns.

Duplicate columns

The function is currently not able to handle multiple columns of trait data (incl. names and units). Those are currently omitted from the output and may be added manually. You can alter the columns to be used to construct the matrix by specifying those in parameters `traits`, `values`, and `units`. Automatic handling of the `Std` columns might be added at a later stage.

Examples

```
pulldata("arthropodtraits")
head(arthropodtraits)
dataset3 <- as.traitdata(arthropodtraits,
                         taxa = "SpeciesID",
                         traits = c("Body_Size", "Dispersal_ability",
                                "Feeding_guild","Feeding_guild_short",
                                "Feeding_mode", "Feeding_specialization",
                                "Feeding_tissue", "Feeding_plant_part",
                                "Endophagous_lifestyle", "Stratum_use",
                                "Stratum_use_short"),
                         units = c(Body_Size = "mm"),
                         keep = c(measurementRemark = "Remark"),
                         metadata = as.metadata(
                           license = "http://creativecommons.org/publicdomain/zero/1.0/"
                            )
)

head(dataset3)

dd3 <-cast.traitdata(dataset3)
head(dd3)
```

---

get_gbif_taxonomy   *Get accepted canonical names and taxonomy for a given species name*

Description

The function maps user provided names to accepted species names.

Usage

```
get_gbif_taxonomy(
  x,
  subspecies = TRUE,
  higherrank = FALSE,
  verbose = FALSE,
  fuzzy = TRUE,
  conf_threshold = 90,
  resolve_synonyms = TRUE
)
```

Arguments

x	a character string or vector of species names.
subspecies	logical. If TRUE (default), the given name is resolved to subspecies epithet, otherwise it will be mapped to species level.
higherrank	logical. If FALSE (default), it will not allow remapping of unknown species names to higher taxon ranks (e.g. genus).
verbose	logical. If FALSE (default), warnings and messages are suppressed.
fuzzy	logical. If FALSE (default), names are only resolved to exactly matching taxa on GBIF taxonomy service. Setting to TRUE may produce matches when failing otherwise.
conf_threshold	numerical, ranging from 0 to 100 (default value = 90). Defines the confidence level of the request to be accepted. To cover for misspellings and errors, could go as low as 50.
resolve_synonyms	
	logical. If TRUE (default), user provided synonyms are mapped to the accepted names on GBIF taxonomy service.

Details

The function relies on package 'taxize' by Scott Chamberlain. It uses the spell-checking and fuzzy matching algorithms provided by Global Names Resolver (`taxize::gnr_resolve()`) and forwards synonyms to the accepted names as provided by GBIF Backbone Taxonomy (`taxize::get_gbif_id_()`).

If 'synonym' is returned as TRUE, the user provided name has been identified as a synonym and was mapped to an accepted name.

The field confidence reports the confidence of the matching procedure performed by the function `get_gbifid_()` of the package 'taxize'. The taxonID is a globally valid URI that links to the taxon description of the GBIF backbone taxonomy.

Value

a data.frame mapping the user supplied names to the accepted taxon names and higher taxonomic information (kingdom, phylum, class, order, family, genus).

Examples

```
get_gbif_taxonomy(c("Chorthippus albomarginatus", "Chorthippus apricarius",
  "Chorthippus biguttulus", "Chorthippus dorsatus", "Chorthippus montanus",
  "Chorthippus parallelus", "Chrysochraon dispar", "Conocephalus dorsalis",
  "Conocephalus fuscus", "Decticus verrucivorus", "Euthystira brachyptera",
  "Gomphocerippus rufus", "Gryllus campestris", "Metrioptera roeselii",
  "Omocestus viridulus", "Phaneroptera falcata", "Platycleis albopunctata",
  "Spec", "Stenobothrus lineatus", "Stenobothrus stigmaticus",
  "Stethophyma grossum", "Tetrix kraussi", "Tetrix subulata",
  "Tetrix tenuicornis", "Tetrix undulata", "Tettigonia cantans",
  "Tettigonia viridissima")
  )

get_gbif_taxonomy("Vicia")
```

---

glossary   *Ecological Trait-data Standard vocabulary (ETS)*

Description

The terms and concepts as defined by the ETS (https://terminologies.gfbio.org/terms/ets/pages/)

Usage

```
glossary
```

Format

An object of class data.frame with 80 rows and 13 columns.

---

heteroptera_raw   *Heteroptera morphometry traits*

Description

Morphometric measures of Heteroptera sampled in grasslands across three regions of Germany.

Details

Trait-based approaches have increased significantly in community ecology during the last decade. This is not least because studies on biodiversity-ecosystem functioning relationships became a major topic in ecology. Species' functions in ecosystems are mediated by their traits. For a better understanding of the relationships between environmental drivers, the community composition of organisms and ecosystems functioning, it is crucial to understand how these relationships are mediated by the communities' trait composition. While there are world-wide efforts to set up trait

databases, most have so far focused on plants and species-poorer taxa such as birds or amphibians. In contrast, for insects, the large number of species makes the gathering of comparable trait data a challenging task. In addition, there is the danger that generic trait information, which is available from common textbooks, may not be sufficient to detect the response of insect communities to environmental change or the consequences of trait changes for ecosystem functioning. One method to overcome this is to take morphometric measurements of species. In this study we measured morphometric traits of a total of 179 Heteroptera species that were sampled by sweep-netting on a total of 150 managed grassland plots across three regions in Germany between 2008 and 2012. These plots represent the whole range of grassland management intensities from extensively used pastures to mown pastures to intensively managed and fertilized meadows. In this paper we provide a database of mean values of 23 morphometric measures across sex and morphotypes for each sampled Heteroptera species. Morphological traits are assumed to be related to their adaptation and function in the environment. Thus the relative morphometric traits can be used as proxies for ecological features of a species that may affect its performance or fitness. Our database can be used by future trait-based studies for developing and testing hypotheses of the functional significance of these traits. Examples include studying the functional responses of insect communities to environmental drivers or studying how the change in trait composition affects ecosystem processes.

Value

The dataset `heteropteraRaw` contains multiple observations of each species (occurence table). The dataset `heteroptera` is a compiled species-trait matrix.

Citation

Cite this dataset as

- Gossner, M. M., N. K. Simons, L. Höck, and W. W. Weisser. 2015. Morphometric measures of Heteroptera sampled in grasslands across three regions of Germany. Ecology 96:1154-1154.

- Data publication: Gossner, M.M, Simons, N.K., Höck, L., Weisser, W.W.,

1. Morphometric measures of Heteroptera sampled in grasslands across three regions of Germany. figshare. https://doi.org/10.6084/m9.figshare.c.3307611.v1

Author(s)

Martin M. Gossner , Nadja K. Simons, Leonhard Höck, Wolfgang W. Weisser

Source

https://figshare.com/articles/Data_Paper_Data_Paper/3561936; Creative Commons 0. To the extent possible under law, the authors have waived all copyright and related or neighboring rights to this data.

See Also

Other rawdata: amniota, amphibio, arthropodtraits, carabids, mammaldiet, pantheria, pulldata()

Other rawdata: amniota, amphibio, arthropodtraits, carabids, mammaldiet, pantheria, pulldata()

| mammaldiet | *Mammal diet database* |

Description

A comprehensive global dataset of diet preferences of mammals ('MammalDIET'). Diet information was digitized from the literature and extrapolated for species with missing information. The original and extrapolated data cover species-level diet information for >99% of all terrestrial mammals.

Details

Ecological trait data are essential for understanding the broad-scale distribution of biodiversity and its response to global change. For animals, diet represents a fundamental aspect of species' evolutionary adaptations, ecological and functional roles, and trophic interactions. However, the importance of diet for macroevolutionary and macroecological dynamics remains little explored, partly because of the lack of comprehensive trait datasets. We compiled and evaluated a comprehensive global dataset of diet preferences of mammals ("MammalDIET"). Diet information was digitized from two global and cladewide data sources and errors of data entry by multiple data recorders were assessed. We then developed a hierarchical extrapolation procedure to fill-in diet information for species with missing information. Missing data were extrapolated with information from other taxonomic levels (genus, other species within the same genus, or family) and this extrapolation was subsequently validated both internally (with a jack-knife approach applied to the compiled species-level diet data) and externally (using independent species-level diet information from a comprehensive continentwide data source). Finally, we grouped mammal species into trophic levels and dietary guilds, and their species richness as well as their proportion of total richness were mapped at a global scale for those diet categories with good validation results. The success rate of correctly digitizing data was 94%, indicating that the consistency in data entry among multiple recorders was high. Data sources provided species-level diet information for a total of 2033 species (38% of all 5364 terrestrial mammal species, based on the IUCN taxonomy). For the remaining 3331 species, diet information was mostly extrapolated from genus-level diet information (48% of all terrestrial mammal species), and only rarely from other species within the same genus (6%) or from family level (8%). Internal and external validation showed that: (1) extrapolations were most reliable for primary food items; (2) several diet categories ("Animal," "Mammal," "Invertebrate," "Plant," "Seed," "Fruit," and "Leaf") had high proportions of correctly predicted diet ranks; and (3) the potential of correctly extrapolating specific diet categories varied both within and among clades. Global maps of species richness and proportion showed congruence among trophic levels, but also substantial discrepancies between dietary guilds. MammalDIET provides a comprehensive, unique and freely available dataset on diet preferences for all terrestrial mammals worldwide. It enables broad-scale analyses for specific trophic levels and dietary guilds, and a first assessment of trait conservatism in mammalian diet preferences at a global scale. The digitalization, extrapolation and validation procedures could be transferable to other trait data and taxa.

Author(s)

Kissling, W.D., Dalby, L., Fløjgaard, C., Lenoir, J., Sandel, B., Sandom, C., Trøjelsgaard, K., Svenning, J.

Source

Cite this dataset as:

- Kissling, W.D., Dalby, L., Fløjgaard, C., Lenoir, J., Sandel, B., Sandom, C., Trøjelsgaard, K., Svenning, J. (2014). Establishing macroecological trait datasets: digitalization, extrapolation, and validation of diet preferences in terrestrial mammals worldwide. Ecol Evol, 4, 2913-2930. https://doi.org/10.1002/ece3.1136

Additionally, please cite the Dryad data package:

- Kissling WD, Dalby L, Fløjgaard C, Lenoir J, Sandel B, Sandom C, Trøjelsgaard K, Svenning J-C (2014) Data from: Establishing macroecological trait datasets: digitalization, extrapolation, and validation of diet preferences in terrestrial mammals worldwide. Dryad Digital Repository. https://doi.org/10.5061/dryad.6cd0v

Creative Commons 0. To the extent possible under law, the authors have waived all copyright and related or neighboring rights to this data.

See Also

Other rawdata: `amniota`, `amphibio`, `arthropodtraits`, `carabids`, `heteroptera_raw`, `pantheria`, `pulldata()`

| mutate.traitdata | *Mutate traits within a traitdata object.* |

Description

This function allows to transform, factorize, or combine trait measurements into compound measurements or update factor levels into binaries.

Usage

```
mutate.traitdata(
  .data,
  ...,
  values = "traitValue",
  traits = "traitName",
  units = "traitUnit"
)
```

Arguments

`.data`	the traitdata object to transform
`...`	named parameters giving definitions of new columns.
`values`	(NOT TESTED) the column name containing the trait values to be used to fill the matrix (default is `traitValue`). Duplicate columns (e.g. `traitValueStd`) will be omitted. See notes.

traits   (NOT TESTED) the column name to be kept for parsing into wide-table (default
                  is traitName). Note that any duplicate column that contains trait names, e.g.
                  traitNameStd will be omitted.

units   (NOT TESTED) the column name containing the units of numerical values (de-
                  fault is traitUnit).

Details

The function handles units for numerical traits and returns the new unit of the computed value
in column traitUnit, if units of input variables were specified according to the units package.
Handling of other columns than traitName and traitValue is not advised at present.

It is advised to mutate traits before applying standardize.traits()! If the mutate function is
applied to a standardised dataset, the new trait will not be mapped automatically to the provided
thesaurus. (automated re-mapping might be added in later versions of the package.)

Value

an updated traitdata object with the new trait measures or facts appended to the original table. If the
given trait name has been refined, it will be replaced.

Examples

```
## Not run:
pulldata("arthropodtraits")
dataset3 <- as.traitdata(arthropodtraits,
                         taxa = "SpeciesID",
                         traits = c("Body_Size", "Dispersal_ability",
                            "Feeding_guild","Feeding_guild_short",
                            "Feeding_mode", "Feeding_specialization",
                            "Feeding_tissue", "Feeding_plant_part",
                            "Endophagous_lifestyle", "Stratum_use",
                            "Stratum_use_short"),
                         units = c(Body_Size = "mm", Dispersal_ability = "unitless"),
                         keep = c(measurementRemark = "Remark"),
                         metadata = list(
                            license = "http://creativecommons.org/publicdomain/zero/1.0/"
                             )
)
head(dataset3)

updated <- mutate.traitdata(dataset3, predator = Feeding_guild == "c" )

head(updated[updated$traitName == "predator",])

levels(updated$traitName)

##

pulldata("heteroptera_raw")
dataset2 <- as.traitdata(heteroptera_raw,
```

```
                   traits = c("Body_length", "Body_width", "Body_height", "Thorax_length",
                              "Thorax_width", "Head_width", "Eye_width", "Antenna_Seg1",
                          "Antenna_Seg2", "Antenna_Seg3", "Antenna_Seg4", "Antenna_Seg5",
                          "Front.Tibia_length", "Mid.Tibia_length", "Hind.Tibia_length",
                          "Front.Femur_length", "Hind.Femur_length", "Front.Femur_width",
                              "Hind.Femur_width", "Rostrum_length", "Rostrum_width",
                                "Wing_length", "Wing_width"),
                   taxa = "SpeciesID",
                   occurrences = "ID"
)
updated <- mutate.traitdata(dataset2,
                        Body_shape = Body_length/Body_width,
                        Body_volume = Body_length*Body_width*Body_height,
                        Wingload = Wing_length*Wing_width/Body_volume)

head(updated[updated$traitName %in% c( "Body_volume"),])

## End(Not run)
```

---

pantheria   *PanTHERIA mammal traits*

Description

Here we describe a global species-level data set of key life-history, ecological and geographical traits of all known extant and recently extinct mammals (PanTHERIA) developed for a number of macroecological and macroevolutionary research projects.

Details

Data were gathered from the literature for 25 types of ecological and life history information for any extant or recently extinct species within class Mammalia (100740 data lines):

1. Activity Cycle; 2. Age at Eye Opening; 3. Age at First Birth; 4. Average Lifespan; 5. Body Mass; 6. Diet; 7. Dispersal Age; 8. Adult Limb Length; 9. Gestation Length; 10. Group Composition & Size; 11. Growth Data; 12. Habitat Layer; 13. Head-Body Length; 14. Interbirth Interval; 15. Litter size; 16. Litters Per Year; 17. Maximum Longevity; 18. Metabolic Rate; 19. Migratory Behaviour; 20. Mortality Data; 21. Population Density; 22. Ranging Behaviour; 23. Sexual Maturity Age; 24. Teat Number; and 25. Weaning Age.

30 specific variables (see Class IV, Table 1) were extracted from the above data types for PanTHERIA from a total of 94729 data lines (before error checking). Additionally, 4 variables were derived from extracted variables within PanTHERIA and 19 variables were calculated from other spatial data sources (see Class V, Section C).

see <http://esapubs.org/archive/ecol/E090/184/metadata.htm> for further information.

Author(s)

Kate E. Jones, Jon Bielby, Marcel Cardillo, Susanne A. Fritz, Justin O'Dell, C. David L. Orme, Kamran Safi, Wes Sechrest, Elizabeth H. Boakes, Chris Carbone, Christina Connolly, Michael J. Cutts, Janine K. Foster, Richard Grenyer, Michael Habib, Christopher A. Plaster, Samantha A. Price, Elizabeth A. Rigby, Janna Rist, Amber Teacher, Olaf R. P. Bininda-Emonds, John L. Gittleman, Georgina M. Mace, and Andy Purvis.

Source

Cite as:

E. Jones, Kate; Bielby, Jon; Cardillo, Marcel; A. Fritz, Susanne; O'Dell, Justin; David L. Orme, C.; Safi, Kamran; Sechrest, Wes; H. Boakes, Elizabeth; Carbone, Chris; Connolly, Christina; Cutts, Michael J.; Foster, Janine K.; Grenyer, Richard; Habib, Michael; Plaster, Christopher A.; Price, Samantha A.; Rigby, Elizabeth A.; Rist, Janna; Teacher, Amber; Bininda-Emonds, Olaf R. P.; Gittleman, John L.; M. Mace, Georgina; Purvis, Andy (2016): PanTHERIA: a species-level database of life history, ecology, and geography of extant and recently extinct mammals. http://esapubs.org/archive/ecol/E090/184/metadata.htm;

Creative Commons 0. To the extent possible under law, the authors have waived all copyright and related or neighboring rights to this data.

See Also

Other rawdata: amniota, amphibio, arthropodtraits, carabids, heteroptera_raw, mammaldiet, pulldata()

---

pulldata	*Pull trait data from the internet*

Description

This function sources a recipe for extracting public trait data.

Usage

```
pulldata(x = NULL)
```

Arguments

x   The name of the dataset to source.

Details

The package 'traitdataform' comes with a collection of recipes for public trait data. These R-scripts

1. define how to read the file from an online source, i.e. a URL to a txt, xlsx, or a ZIP archive
2. assigns metadata attributes about authorship, license and original publication
3. provide parameters for the standardize() function, i.e. a trait thesaurus, mappings and units.

New recipes can be suggested as a pull request via the package development page (https://github.com/EcologicalTraitData/tra

Author(s)

Florian D. Schneider

See Also

Other rawdata: amniota, amphibio, arthropodtraits, carabids, heteroptera_raw, mammaldiet, pantheria

Examples

```
# to get a list of all available data within the package
pulldata()

# to import a dataset
pulldata("carabids")
```

---

rbind.traitdata	*Combine trait datasets*

Description

Method for function `rbind()` to append objects of class 'traitdata' to each other.

Usage

```
## S3 method for class 'traitdata'
rbind(
  ...,
  metadata = NULL,
  datasetID = NULL,
  metadata_as_columns = FALSE,
  drop = NULL
)
```

Arguments

...	two or more objects of class traitdata.
metadata	a list of metadata entries which are to be added as dataset-level information.
datasetID	a vector of the same length as number of objects. If NULL (default), object names will be returned as ID.
metadata_as_columns	
	logical (defaults to FALSE) or vector of columns to return. If TRUE , the output will contain the "author", "license", "datasetName" and (autogenerated) "datasetID" name, if those are provided. If character vector, the output will contain the listed columns.
drop	FALSE by default. If true, columns that are not present in all datasets will be dropped.

Details

Metadata are ideally already included in the datasets as attributes (see ?as.traitdata). The function rbind.traitdata() takes a list of lists as its metadata argument. The outer list must have the same length as the provided objects to combine, with each entry containing objects according to the terms of the Ecological Traitdata Standard (http://ecologicaltraitdata.github.io/ETS/#metadata-vocabulary).

A lookup table for dataset details will be appended as an attribute to the output dataset, linked to each entry via the field datasetID. It can be accessed by calling attributes(<dataset>)$datasets.

Examples

```
pulldata("carabids")

dataset1 <- as.traitdata(carabids,
  taxa = "name_correct",
  traits = c("body_length", "antenna_length", "metafemur_length"),
  units = "mm",
  keep = c(datasetID = "source_measurement", measurementRemarks = "note"),
  metadata = as.metadata(
    bibliographicCitation = c(
        "van der Plas et al. (2017) Methods in Ecol. & Evol., doi: 10.1111/2041-210x.12728"
        ),
    author = "Fons van der Plas",
    license = "http://creativecommons.org/publicdomain/zero/1.0/"
    )
)


traits1 <- as.thesaurus(
 body_length = as.trait("body_length",
   expectedUnit = "mm", valueType = "numeric",
   identifier = "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Body_length"),
 antenna_length = as.trait("antenna_length",
   expectedUnit = "mm", valueType = "numeric",
   identifier = "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Antenna_length"),
 metafemur_length = as.trait("metafemur_length",
   expectedUnit = "mm", valueType = "numeric",
   identifier = "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Femur_length")
)

dataset1Std <- standardize.traits(dataset1, thesaurus = traits1)

# occurrence table:

pulldata("heteroptera_raw")

dataset2 <- as.traitdata(heteroptera_raw,
  taxa = "SpeciesID",
  traits = c("Body_length", "Antenna_Seg1", "Antenna_Seg2",
    "Antenna_Seg3", "Antenna_Seg4", "Antenna_Seg5", "Hind.Femur_length"),
```

```
    units = "mm",
    keep = c(sex = "Sex", references = "Source", lifeStage = "Wing_development"),
    metadata = as.metadata(
     bibliographicCitation = "Gossner et al. (2015) Ecology, 96:1154. doi: 10.1890/14-2159.1",
       author = "Martin Gossner",
       license = "http://creativecommons.org/publicdomain/zero/1.0/"
       )
)

dataset2 <- mutate.traitdata(dataset2,
  antenna_length = Antenna_Seg1 + Antenna_Seg2 + Antenna_Seg3 + Antenna_Seg4 + Antenna_Seg5
   )



traits2 <- as.thesaurus(
 Body_length = as.trait("body_length",
    expectedUnit = "mm", valueType = "numeric",
    identifier = "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Body_length"),
 antenna_length = as.trait("antenna_length",
    expectedUnit = "mm", valueType = "numeric",
    identifier = "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Antenna_length"),
 Hind.Femur_length = as.trait("metafemur_length",
    expectedUnit = "mm", valueType = "numeric",
    identifier = "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Femur_length")
)

dataset2Std <- standardize_traits(dataset2, thesaurus = traits2)

database <- rbind(dataset1Std, dataset2Std,
                  datasetID = c("vanderplas17", "gossner15"),
                  metadata_as_columns = c("author"))
head(database)
```

---

standardize	*Standardize trait datasets*

Description

wrapper that applies `standardize.taxonomy()` and `standardize.traits()` in one go.

Usage

```
standardize(x, ...)
```

Arguments

x	a traitdata object (as returned by `as.traitdata()`) or a data table containing at least the column 'scientificName.
...	parameters as described for `standardize.traits()` and `standardize.taxonomy()`.

See Also

Other standardize: [standardize_taxa](), [standardize_traits]()

---

standardize.exploratories
                          Standardize Georeference from Biodiversity Exploratories Plot ID

Description

Adds columns of georeference to trait-data table if measurements relate to specimens from the Biodiversity Exploratories plots or regions.

This function requires valid credentials for the Biodiversity Exploratories Information System (BExIS)!

Usage

```
standardize.exploratories(
  x,
  plots = "locationID",
  user = NULL,
  pswd = NULL,
  getdata = !is.null(user),
  fillall = TRUE,
  ...,
  verbose = NULL
)
```

Arguments

x	A traitdata table of class 'traitdata'.
plots	Name of column containing the plot IDs. Must match the Exploratories EP_PlotID scheme (e.g. AEG1, AEG12, HEW21, ...).
user	User name for Biodiversity Exploratories Information System (BExIS, https://www.bexis.uni-jena.de); required for download of exact geolocation.
pswd	password for above request. Will be prompted for if not provided.
getdata	logical; if FALSE it suppresses the extraction of location data from BExIS, and will not ask for user credentials. Instead, public data of less precision will be used.
fillall	if TRUE (default), the output will contain all terms suggested by the glossary and fill empty columns with NA. This is required for an upload of the data to BExIS.
...	If input is a rawdata table of type species-trait matrix or occurence table (wide table) then provide parameters according to as.traitdata().
verbose	logical; if FALSE all messages will be suppressed.

Value

std

Examples

```
## Not run:
moths <- read.service(21247, dec = ",")

dataset1 <- as.traitdata(moths, taxa = "species", traits = c(body_mass =
    "weight", wing_length = "wing_length", wing_width = "wing_width", wing_area =
    "wing_area", wing_loading = "wing_loading"),
  keep = c(locationID = "plot"))

dataset1Std <- standardize.exploratories(dataset1)

## End(Not run)
```

---

standardize_taxa   *Standardize scientific names of species*

Description

Adds columns to a traitdata object containing accepted species names and relates to globally unique taxon identifiers via URI.

Usage

```
standardize_taxa(
  x,
  method = get_gbif_taxonomy,
 method_options = c(subspecies = TRUE, higherrank = FALSE, verbose = FALSE, fuzzy =
    TRUE, conf_threshold = 90, resolve_synonyms = TRUE),
  return = c("kingdom", "phylum", "class", "order", "family"),
  ...
)
```

Arguments

x	a traitdata object (as returned by as.traitdata()) or a data table containing at least the column 'scientificName.
method	default option is get_gbif_taxonomy. In principle, takes any function that takes a vector of species names as input to produce a taxonomy lookup table (i.e. mapping user-provided scientificName to taxonID and other taxon-level information). Will allow to chose from different sources of taxonomic reference.
method_options	a name vector of arguments to be passed on to method. See get_gbif_taxonomy for options.

| return | a character vector containing the informatoin that should be extracted into the output. Valid entries are the column names returned by function `get_gbif_taxonomy()`. See 'Details'. |
| ... | parameters to be ignored, forwarded from wrapper function `standardize()`. |

Details

Taxonomic standardisation is an enormous challenge for biodiversity data management and research. Constant changes in species and higher taxa, refinements of phylogenetic trees and changing attribution to original authors, moving species into other genera or difficulties to place species into the Linean nomenclature results in highly fluctuent taxonomic definitions.

As a consequence, there is not one reference for accepted species names and depending on the field of resaerch and taxonomic focus other authorities will be employed.

For reasons of simplicity and because of its high coverage of taxa, the function `standardize.taxonomy()` uses the GBIF Backbone Taxonomy as its reference system and resolves all provided species names to the accepted name according to GBIF (resolving misspellings and synonyms in the process). We invite pull requests to make this function more general and enable a choice of a taxonomic reference.

See Also

Other standardize: `standardize_traits()`, `standardize()`

Examples

```
## Not run:

pulldata("carabids")

dataset1 <- as.traitdata(carabids,
  taxa = "name_correct",
  traits = c("body_length", "antenna_length", "metafemur_length"),
  units = "mm",
  keep = c(datasetID = "source_measurement", measurementRemark = "note"),
  metadata = list(
    bibliographicCitation = attributes(carabids)$citeAs,
    author = "Fons van der Plas",
    license = "http://creativecommons.org/publicdomain/zero/1.0/"
    )
)

dataset1Std <- standardize_taxa(dataset1)

## End(Not run)
```

---

standardize_traits   *Standardize trait names and harmonize measured values and reported facts*

Description

Adds columns to a traitdata table with standardized trait names and relates them to globally unique identifiers via URIs. Optionally converts units of values and renames factor levels into accepted terms.

Usage

```
standardize_traits(
  x,
  thesaurus = attributes(x)$thesaurus,
  rename = NULL,
  categories = c("No", "Yes"),
  output = "logical",
  ...
)
```

Arguments

x	a traitdata object (as returned by as.traitdata()) or a data table containing at least the column 'scientificName.
thesaurus	an object of class 'thesaurus' (as returned by as.thesaurus()).
rename	a named vector to map user-provided names to thesaurus object names (see Details).
categories	target categories for binary/logical traits harmonization.
output	behaviour of fixlogical(). see [fixlogical()]().
...	parameters to be ignored, forwarded from wrapper function standardize().

Details

The function matches the trait names provided in 'traitName' to the traits provided in the thesaurus (in field 'trait'). Matching must be exact (case sensitive). Fuzzy matching may be provided in a later version of the package.

The function parameter 'rename' should be provided to map trait names where user-provided names and thesaurus names are different. In this case, rename should be a named vector with the target names used in the thesaurus as names, and the original names as provided in 'traitName' as value. E.g. rename = c()

See Also

Other standardize: [standardize_taxa](), [standardize]()

Other standardize: [standardize_taxa](), [standardize]()

Examples

```
pulldata("carabids")

dataset1 <- as.traitdata(carabids,
  taxa = "name_correct",
  traits = c("body_length", "antenna_length", "metafemur_length"),
  units = "mm",
  keep = c(datasetID = "source_measurement", measurementRemark = "note"),
  metadata = list(
    bibliographicCitation = attributes(carabids)$citeAs,
    author = "Fons van der Plas",
    license = "http://creativecommons.org/publicdomain/zero/1.0/"
    )
)

traitlist <- as.thesaurus(
body_length = as.trait("body_length", expectedUnit = "mm", valueType = "numeric",
    identifier = "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Body_length"),
antenna_length = as.trait("antenna_length", expectedUnit = "mm", valueType = "numeric",
    identifier = "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Antenna_length"),
metafemur_length = as.trait("metafemur_length", expectedUnit = "mm", valueType = "numeric",
    identifier = "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Femur_length")
)

dataset1Std <- standardize_traits(dataset1, thesaurus = traitlist)



## Example: matching of original names to thesaurus

pulldata("heteroptera_raw")

dataset2 <- as.traitdata(heteroptera_raw,
  taxa = "SpeciesID",
  traits = c("Body_length", "Antenna_Seg1", "Antenna_Seg2",
    "Antenna_Seg3", "Antenna_Seg4", "Antenna_Seg5", "Hind.Femur_length"),
  units = "mm",
  keep = c(sex = "Sex", references = "Source", lifestage = "Wing_development"),
  metadata = list(
    bibliographicCitation = attributes(heteroptera_raw)$citeAs,
    license = "http://creativecommons.org/publicdomain/zero/1.0/"
    )
)



traits2 <- as.thesaurus(
    Body_length = as.trait("Body_length",
            expectedUnit = "mm", valueType = "numeric",
            traitDescription = "From the tip of the head to the end of the abdomen"),
    Antenna_Seg1 = as.trait("Antenna_Seg1",
```

```
                    expectedUnit = "mm", valueType = "numeric",
                    traitDescription = "Length of first antenna segment",
              broaderTerm = "http://ecologicaltraitdata.github.io/TraitDataList/Antenna_length"),
        Antenna_Seg2 = as.trait("Antenna_Seg2",
                    expectedUnit = "mm", valueType = "numeric",
                    traitDescription = "Length of second antenna segment",
              broaderTerm = "http://ecologicaltraitdata.github.io/TraitDataList/Antenna_length"),
        Antenna_Seg3 = as.trait("Antenna_Seg3",
                    expectedUnit = "mm", valueType = "numeric",
                    traitDescription = "Length of third antenna segment",
              broaderTerm = "http://ecologicaltraitdata.github.io/TraitDataList/Antenna_length"),
        Antenna_Seg4 = as.trait("Antenna_Seg4",
                    expectedUnit = "mm", valueType = "numeric",
                    traitDescription = "Length of fourth antenna segment",
              broaderTerm = "http://ecologicaltraitdata.github.io/TraitDataList/Antenna_length"),
        Antenna_Seg5 = as.trait("Antenna_Seg5",
                    expectedUnit = "mm", valueType = "numeric",
                    traitDescription = "Length of fifth antenna segment (only Pentatomoidea)",
              broaderTerm = "http://ecologicaltraitdata.github.io/TraitDataList/Antenna_length"),
        Hind.Femur_length = as.trait("Hind.Femur_length",
                    expectedUnit = "mm", valueType = "numeric",
                    traitDescription = "Length of the femur of the hind leg",
                  broaderTerm = "http://t-sita.cesab.org/BETSI_vizInfo.jsp?trait=Femur_length")
        )

    dataset2Std <- standardize_traits(dataset2,
        thesaurus = traits2
        )
```

Index