Package ‘letsRept’

October 21, 2025

Title An Interface to the Reptile Database

Version 1.1.0

URL https://joao-svalencar.github.io/letsRept/

BugReports https://github.com/joao-svalencar/letsRept/issues

Description Provides tools to retrieve and summarize taxonomic information and synonymy data for reptile species using data scraped from The Reptile Database website (<https://reptile-database.reptarium.cz/>). Outputs include clean and structured data frames useful for ecological, evolutionary, and conservation research.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports dplyr, httr, parallel, pbapply, pbmcapply, rvest, stringr, tidyr, utils, xml2

Depends R (>= 3.5)

LazyData true

LazyDataCompression xz

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author João Paulo dos Santos Vieira-Alencar [aut, cre] (ORCID: <https://orcid.org/0000-0001-6894-6773>), Christoph Liedtke [aut] (ORCID: <https://orcid.org/0000-0002-6221-8043>)

Maintainer João Paulo dos Santos Vieira-Alencar <joaopaulo.valencar@gmail.com>

Repository CRAN

Date/Publication 2025-10-21 09:10:40 UTC

Contents

`allReptiles` *Example dataset: allReptiles*

Description

This dataset contains the valid names and url addresses for all reptile species cataloged in The Reptile Database.

Usage

```
allReptiles
```

Format

A dataframe (download: September 15th, 2025) with 12500 rows and 8 variables:

order A species current order

suborder A species current suborder

family A species current family

genus A species current genus

species A character vector with known current valid name for all reptile species cataloged in The Reptile Database website

year A species description year

author The authors that described the species under the current valid name

url A character column with the respective url to access all reptile species cataloged in The Reptile Database website information

Source

The data was sampled from The Reptile Database website https://reptile-database.reptarium.cz using function letsRept::reptSpecies() with the url obtained from an 'Advanced search' set to exclude all reptile species described to the fictional planet Arrakis (-Arrakis).

`allSynonyms` *Example dataset: allSynonyms*

Description

This dataset contains the valid names and respective listed synonyms for all reptile species cataloged in The Reptile Database.

Usage

```
allSynonyms
```

Format

A dataframe with 53,159 rows and 2 variables:

species A character vector with known current valid name for all reptile species cataloged in The Reptile Database website (download: May 23rd, 2025)

synonym A character column with the respective synonyms for all reptile species cataloged in The Reptile Database website information (download: May 23rd, 2025)

Source

The data was sampled from The Reptile Database website https://reptile-database.reptarium.cz using function letsRept::reptSynonyms(letsRept::allReptiles)

`allSynonymsRef` *Example dataset: allSynonymsRef*

Description

This dataset contains the valid names and respective listed synonyms for all reptile species cataloged in The Reptile Database.

Usage

```
allSynonymsRef
```

Format

A dataframe with 111,549 rows and 3 variables:

species A character vector with known current valid name for all reptile species cataloged in The Reptile Database website (download: September 18tb, 2025)

synonym A character column with the respective synonyms for all reptile species cataloged in The Reptile Database website information (download: September 18tb, 2025)

ref A charater column with the synonyms and respective references that used it

Source

The data was sampled from The Reptile Database website https://reptile-database.reptarium.cz using function letsRept::reptSynonyms(letsRept::allReptiles)

`br_snakes_atlas` *Example dataset: br_snakes_atlas*

Description

This dataset contains Table S3, the georeferenced type localities, from the supplementary material provided by Nogueira et al., (2019)

Usage

```
br_snakes_atlas
```

Format

A dataframe with 411 rows and 3 colums:

species A character vector with known current valid name for all reptile species cataloged in The Reptile Database website

latitude Latitude coordinates in decimal degrees

longitude Longitude coordinates in decimal degrees

Source

Nogueira et al. (2019) supplementary material, Table S3

References

Nogueira, C. C., Argôlo, A. J. S., Arzamendia, V., Azevedo, J. A. R., Barbo, F. E., Bérnils, R. S., … & Martins, M. (2019). Atlas of Brazilian Snakes: Verified Point-Locality Maps to Mitigate the Wallacean Shortfall in a Megadiverse Snake Fauna. South American Journal of Herpetology, 14(sp1), 1–274. http://dx.doi.org/10.2994/sajh-d-19-00120.1

`letsRept_ReptTraits` *Example dataset: letsRept_ReptTraits*

Description

This dataset is a version of ReptTraits (Oskyrko et al. 2024) with two additional columns.

Usage

```
letsRept_ReptTraits
```

Format

A dataframe with 12,060 rows and 50 variables. The first three columns are:

species Species name as in the original ReptTraits database

RDB Current valid name according to the May 2025 version of the Reptile Database

nomenclature.status Status from reptSync and reptSplitCheck. Additional status are: "extinct" and "manual_fix"

Source

The original data source is from Oskyrko et al. (2024); The new nomenclature in the RDB column was collected from the Reptile Database website https://reptile-database.reptarium.cz using functions reptSync and reptSplitCheck

---

`letsRept_SquamBase` *Example dataset: letsRept_SquamBase*

Description

This dataset is a version of SquamBase (Meiri, 2024) with two additional columns.

Usage

```
letsRept_SquamBase
```

Format

A dataframe with 11,744 rows and 86 variables. The first three columns are:

species Species name as in the original SquamBase database

RDB Current valid name according to the May 2025 version of the Reptile Database

nomenclature.status Status from reptSync and reptSplitCheck. Additional status are: "extinct" and "manual_fix"

Source

The original data source is from Meiri (2024); The new nomenclature in the RDB column was collected from the Reptile Database website https://reptile-database.reptarium.cz using functions reptSync and reptSplitCheck

---

`reptAdvancedSearch` *Search The Reptile Database website (RDB): Advanced*

Description

Creates a search URL for retrieving species lists from RDB based on multiple filters. This URL is primarily used by `reptSpecies`, but can also be used manually for advanced queries.

If a synonym is provided and can be unambiguously matched to a valid species, the function also prints detailed information for that species.

Usage

```
reptAdvancedSearch(
  higher = NULL,
  genus = NULL,
  year = NULL,
  common_name = NULL,
  synonym = NULL,
  location = NULL,
  verbose = TRUE,
  exact = FALSE
)
```

Arguments

`higher`	Character string. A higher-level reptile taxon above genus (e.g., `"snakes"` or `"Boidae"`).
`genus`	Character string. The current valid name of a reptile genus (e.g., `"Apostolepis"`).
`year`	Character string. Filters the search by year of species description (e.g., `"2025"`).
`common_name`	Character string. A common name potentially linked to a species or genus (e.g., `"tree boa"`).
`synonym`	Character string. A name potentially regarded as a synonym of a valid taxon (e.g., `"Boa diviniloqua"`).
`location`	Character string. A country or region name used to list species expected to occur there.
`verbose`	Logical. To be passed to `reptSpecies()` in the case of a provided synonym corresponds unambiguously to a valid species. If `TRUE`, prints status messages and species information in the console. Default is `TRUE`.
`exact`	Logical. To return outputs that matches exactly the searched term (e.g., avoid returning genus "Boaedon" when searching for "Boa"). Default is `FALSE`.

Value

A character string containing the URL to be used in `reptSpecies`.

If a provided synonym corresponds unambiguously to a valid species, the function also prints species information retrieved from RDB to the console.

Note

The argument `exact` does not work properly for searches using logical arguments (e.g. `AND/OR`). If you want to force an exact match (e.g., `"Boa"` as a phrase) with multiple terms (e.g., `"Boa OR Apostolepis"`), you must manually include quotes in the input string, e.g., `"\"Boa\" OR Apostolepis"`.

Logical operators (e.g., `OR`, `AND`) are supported and will be properly formatted in the search. To exclude terms, use a leading minus sign (e.g., `higher = "-snakes"`) following RDB's query syntax, instead of using `NOT`.

When a synonym is matched to a single valid species, the function will also display the species' full information as a side effect.

Examples

```
reptAdvancedSearch(higher = "snakes", year = "2010", location = "Brazil")
reptAdvancedSearch(year = "2010 OR 2011 OR 2012")
reptAdvancedSearch(genus = "Apostolepis OR \"Boa\" OR Atractus") #quotes "Boa"
```

`reptCompare` *Compare species nomenclature between datasets or with Reptile Database Data*

Description

This function compares a list of species (`x`) with another list (`y`), typically from the Reptile Database (RDB). If `y` is not provided, it defaults to using the internal object `allReptiles`, extracted from RDB (version: May, 2025). The function returns species from xthat are either unmatched ("review") or matched with `y` or with `allReptiles`, if `y` is NULL. If `y` is provided and any species from `x` is a valid name according to RDB but is absent from `y`, it receives the status `absent`

Usage

```
reptCompare(x = NULL, y = NULL, filter = NULL, compareDataset = FALSE)
```

Arguments

`x`	A character vector or a data frame containing a column named `species` with species names to be compared.
`y`	Optional. A character vector or a data frame containing a column named `species` to be compared. Defaults to the internal object `allReptiles`.

`filter`	Optional. A character string or a vector of characters. If `"review"`, returns only unmatched species. If `"matched"`, returns only matched species. If `"absent"`, returns only species from `x` that are absent from `y`. If `NULL` (default), returns a data frame with all species and respective statuses. Multiple filters can be concatenated and the resulting data frame will return all species with both status.
`compareDataset`	Logical. If TRUE, assumes all input names are up-to-date and compares the input list with the reference database to identify any missing species.

Value

A character vector (if `filter` is `"review"`, `"matched"` or `"absent"`), or a data frame with columns:

species Species names from `x`

status Comparison result: `"review"`, `"matched"` or `"absent"`

If compareDataset = `TRUE`, then the function returns a vector of species from `y` that is absent from `x`.

Examples

```
my_species <- data.frame(species = c("Boa constrictor", "Pantherophis guttatus", "Fake species"))
reptCompare(my_species)
reptCompare(my_species, filter = "review")
reptCompare(my_species, filter = "matched")
```

`reptRefs` *Extract references from the Reptile Database*

Description

Extract references from the Reptile Database

Usage

```
reptRefs(x = NULL, getLink = TRUE)
```

Arguments

`x`	A binomial species name (e.g., "Boa constrictor").
`getLink`	Logical; if TRUE, also returns associated links to references.

Value

A character vector of references (if getLink = FALSE), or a data frame with columns `reference` and `link`.

Examples

```
df <- reptRefs("Apostolepis adhara")
```

`reptSearch` *Search for a Single Reptile Species in The Reptile Database (RDB)*

Description

Queries The Reptile Database (RDB) for information about a single reptile species using its binomial name.

Usage

```
reptSearch(binomial=NULL, getRef=FALSE, verbose=TRUE)
```

Arguments

`binomial`	Character string. The valid binomial name of a reptile species (e.g., "Boa constrictor").
`getRef`	Logical. If `TRUE`, returns the list of references from RDB associated with the species. Default is `FALSE`.
`verbose`	Logical. If `TRUE`, prints species information in the console. Default is `TRUE`.

Value

A list containing species information retrieved from The Reptile Database. If `getRef = TRUE`, returns references related to the species.

References

Uetz, P., Freed, P., & Hošek, J. (Eds.). (2025). The Reptile Database. Retrieved from `http://www.reptile-database.org`

See Also

`reptSynonyms`, `reptSpecies` for related species data functions.

Examples

```
reptSearch("Boa constrictor")
reptSearch("Boa constrictor", getRef = TRUE)
```

`reptSpecies` *Retrieve Reptile Species and Taxonomic Information from RDB*

Description

Retrieves a list of reptile species from The Reptile Database (RDB) based on a search URL, and optionally returns detailed taxonomic information for each species. This function can also save progress to disk during sampling and extract species-specific URLs for further use.

Usage

```
reptSpecies(
  url = NULL,
  showProgress = TRUE,
  dataList = NULL,
  taxonomicInfo = FALSE,
  fullHigher = FALSE,
  getLink = FALSE,
  cores = 1
)
```

Arguments

`url`	Character string. A search URL generated via an advanced search on the RDB website or with `reptAdvancedSearch`.
`showProgress`	Logical. If `TRUE`, prints sampling progress in the console. Default is `FALSE`.
`dataList`	Optional. A data frame with columns `species` and `url`, used to extract taxonomic information from previously sampled species links.
`taxonomicInfo`	Logical. If `TRUE`, returns taxonomic information for each species, including order, suborder, family, genus, author, and year. Default is `FALSE`.
`fullHigher`	Logical. If `TRUE`, includes the full higher taxonomic hierarchy as reported by RDB (e.g., including subfamilies). Requires `taxonomicInfo = TRUE`. Default is `FALSE`.
`getLink`	Logical. If `TRUE`, includes the RDB URL for each species (useful for follow-up functions like `reptSynonyms`). Default is `FALSE`.
`cores`	Integer. Number of CPU cores to use for parallel processing. Default is `cores = 1`.

Value

If `taxonomicInfo = FALSE` (default), returns a character vector of species names.

If `taxonomicInfo = TRUE`, returns a data frame with columns: `order`, `suborder` (if available), `family`, `genus`, `species`, `author`, and `year`.

If `fullHigher = TRUE`, includes an additional column with the full higher taxa classification.

If `getLink = TRUE`, includes a column with the URL for each species' page on RDB.

See Also

`reptAdvancedSearch`, `reptSynonyms`, `reptSearch`

Examples

```
boa <- reptSpecies(reptAdvancedSearch(genus = "Boa"),
                                  taxonomicInfo = TRUE,
                                  cores = 2)
```

`reptSplitCheck` *Check for potential taxonomic splits in a query*

Description

Check for potential taxonomic splits in a query

Usage

```
reptSplitCheck(
  x,
  pubDate = NULL,
  includeAll = FALSE,
  verbose = TRUE,
  cores = 1,
  showProgress = TRUE,
  exact = FALSE
)
```

Arguments

`x`	A character vector of species names to check. Usually from a database.
`pubDate`	Integer. An year (e.g., 2019) used as a reference date from when to check potential taxonomic split
`includeAll`	Logical; If `TRUE`, include all species described since `pubDate` regardless of if it is already included in the queried species list. Default is `FALSE`
`verbose`	Logical; If `TRUE`, prints progress messages. Default is `TRUE`.
`cores`	Integer. Number of CPU cores to use for parallel processing. Default is `cores = 1`.
`showProgress`	Logical. If `TRUE`, prints data sampling progress. Default is `TRUE`.
`exact`	Logical. Will search queried names for exact matches only (e.g., does not retrieve "Tantilla cf. melanocephala" when searching for "Tantilla melanocephala"). Default is `FALSE`.

Value

A data frame with the following columns:

- `query`: the original input names.
- `RDB`: the best-matching valid names according to The Reptile Database.
- `status`: a status label indicating the result of the match (`"check_split"`, `"up_to_date"`, `"not_found"`, or `"failed"`).

Examples

```
query <- c(
"Atractus dapsilis",
"Atractus trefauti",
"Atractus snethlageae",
"Tantilla melanocephala",
"Oxybelis aeneus",
"Oxybelis rutherfordi",
"Vieira-Alencar authoristicus",
"Oxybelis aeneus",
"Bothrops pauloensis")


result <- reptSplitCheck(x=query,
                        pubDate = 2019,
                        cores = 2,
                        showProgress = FALSE)

result <- reptSplitCheck(x=query,
                        pubDate = 2019,
                        cores = 2,
                        showProgress = FALSE,
                        includeAll = TRUE)
```

`reptStats` *Summarize Taxonomic Composition*

Description

This function summarizes the taxonomic content of a species list, typically an object created with `reptSpecies` with higher taxa information. If no object is provided it summarizes the internal dataset `allReptiles`.

Usage

```
reptStats(
  x = letsRept::allReptiles,
  verbose = FALSE,
```

```
  order = NULL,
  suborder = NULL,
  family = NULL,
  genus = NULL
)
```

Arguments

`x`	A data frame containing reptile taxonomy data. Defaults to the internal dataset `letsRept::allReptiles`.
`verbose`	Logical. If `TRUE`, returns a list of taxon names by rank. If `FALSE` (default), returns a summary table of counts.
`order`	Optional. A character string specifying a taxonomic order to filter by (e.g., `"Squamata"`).
`suborder`	Optional. A character string specifying a taxonomic suborder to filter by (e.g., `"Serpentes"`).
`family`	Optional. A character string specifying a family to filter by (e.g., `"Elapidae"`).
`genus`	Optional. A character string specifying a genus to filter by (e.g., `"Micrurus"`).

Details

The output can be either a compact table with taxonomic unit counts or a verbose list of names within each rank.

Optional arguments allow the user to filter the dataset by specific taxonomic levels (e.g., order, suborder, family, genus) before summarizing.

Value

Either a named list of taxonomic units (`verbose = TRUE`) or a data frame with taxonomic ranks and the number of units per rank (`verbose = FALSE`).

Examples

```
# Basic usage with default dataset
reptStats()

# Verbose summary listing elements in each rank
reptStats(verbose = TRUE)

# Filter by family and return summary table
reptStats(family = "Elapidae")

# Combine filters and return list
reptStats(suborder = "Serpentes", verbose = TRUE)
```

`reptSync` *Synchronize Species Names Using The Reptile Database*

Description

Queries a list of species names through `reptSearch()` and returns a data frame with the currently valid names and taxonomic status for each input.

Usage

```
reptSync(
  x,
  solveAmbiguity = TRUE,
  cores = 1,
  showProgress = TRUE,
  getLink = FALSE
)
```

Arguments

`x`	A character vector of taxon names to be matched (e.g., species lists, phylogenetic tip labels, or trait table entries).
`solveAmbiguity`	Logical. If `TRUE`, attempts to resolve ambiguous names by retrieving all possible valid species to which the query may refer. Default is `TRUE`.
`cores`	Integer. Number of CPU cores to use for parallel processing. Default is `cores = 1`.
`showProgress`	Logical. If `TRUE`, displays progress updates during processing. Default is `TRUE`.
`getLink`	Logical. If `TRUE`, retrieves searched species URLs. Defaults if `FALSE`.

Value

A data frame with the following columns:

- `query`: the original input names.
- `RDB`: the best-matching valid names according to The Reptile Database.
- `status`: a status label indicating the result of the match (`"up_to_date"`, `"updated"`, `"updated_typo"`, `"ambiguous"`, `"merge"`, or `"not_found"`).
- `url`: Optional, if getLink = TRUE returns the URL of the species page retrieved for each match, or a list of possible matches if ambiguous.

Note

`reptSync()` does not make authoritative taxonomic decisions. It matches input names against currently accepted names in The Reptile Database (RDB). A name marked as `"up_to_date"` may still refer to a taxon that has been split, and thus may not reflect the most recent population-level taxonomy.

For ambiguous names, the `url` field may contain multiple links corresponding to all valid species to which the queried name is considered a synonym.

See package vignettes for more details.

References

Liedtke, H. C. (2018). AmphiNom: an amphibian systematics tool. *Systematics and Biodiversity*, 17(1), 1–6. https://doi.org/10.1080/14772000.2018.1518935

Examples

```
query <- c("Vieira-Alencar authoristicus", "Boa atlantica", "Boa diviniloqua", "Boa imperator")


reptSync(x = query, cores = 2)
```

`reptSynonyms` *Retrieve Synonyms for Reptile Species from RDB*

Description

Retrieves a data frame containing the current valid names of reptile species along with all their recognized synonyms, as listed in The Reptile Database (RDB). Optionally, the references citing each synonym can also be included.

Usage

```
reptSynonyms(x, getRef = FALSE, showProgress = TRUE, cores = 1)
```

Arguments

`x`	A character string with a species binomial or a data frame with columns `species` and `url`, typically the output of `reptSpecies` with `getLink = TRUE`.
`getRef`	Logical. If `TRUE`, includes the reference(s) in which each synonym was mentioned. Default is `FALSE`.
`showProgress`	Logical. If `TRUE`, prints data sampling progress. Default is `TRUE`.
`cores`	Integer. Number of CPU cores to use for parallel processing. Default is `cores = 1`.

Value

A data frame with columns:

- `species`: The valid species name according to RDB.
- `synonym`: Recognized synonyms and chresonyms for the species. Chresonyms are usually separated from authors with an emdash.
- `reference` (optional): If `getRef = TRUE`, the citation where the synonym was reported.

References

Uetz, P., Freed, P., Aguilar, R., Reyes, F., Kudera, J., & Hošek, J. (eds.) (2025). The Reptile Database. Retrieved from http://www.reptile-database.org

See Also

`reptSpecies`, `reptAdvancedSearch`

Examples

```
# Filter species belonging to genus Boa
boa <- letsRept::allReptiles[grep("^Boa\\s", letsRept::allReptiles$species), ]


# Retrieve synonyms (without references)
boa_syn <- reptSynonyms(boa, getRef = FALSE, cores = 2)
Bconstrictor_syn <- reptSynonyms(x = "Boa constrictor")
```

`reptTidySyn` *Printing reptSync and reptSplitCheck outcomes in a tidy way*

Description

Prints the data frame derived from `reptSync` or `reptSplitCheck` in a tidy way. Optionally, it filters the data frame for species with unresolved nomenclature only.

Usage

```
reptTidySyn(df, filter = NULL)
```

Arguments

`df`	The data frame derivated from `reptSync`
`filter`	Logical. If `TRUE` will print only the species entries with unresolved nomenclature (e.g.: ambiguous or not_found). Default is `TRUE`

Value

Invisibly returns `NULL`. Used for side-effect printing only.

Examples

```
df <- data.frame(
  species = c("Genus epithet 1",
              "Genus epithet 2",
              "Genus epithet 3",
              "Genus epithet 4",
              "Genus epithet 5"),
  synonyms = c("Genus epithet 1.1; Genus epithet 1.2",
                "Genus epithet 2",
                "Genus epithet 3",
                "Not found",
                "Genus epithet 5.1; Genus epithet 5.2; Genus epithet 5.3"),
  status = c("ambiguous",
             "updated",
             "up_to_date",
             "not_found",
             "ambiguous"),
  stringsAsFactors = FALSE
)
reptTidySyn(df, filter = c("ambiguous", "not_found"))
```

Index