Package: reptiledbr (via r-universe)

May 16, 2026

Title Interface to the Reptile Database for Querying and Retrieving Taxonomic Data

Version 0.0.2

Description Provides tools to search, access, and format taxonomic information from the Reptile Database (<`http://reptile-database.org`>) directly within R. Users can retrieve species-level data, distribution, etymology, synonyms, common names, and other relevant information for reptiles. Designed for taxonomists, ecologists, and biodiversity researchers.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports xml2, stringr, purrr, tibble, tidyr, dplyr, utils, lifecycle, fuzzyjoin, cli

Suggests covr, knitr, mockery, reptiledb.data, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 4.1.0)

URL `https://github.com/PaulESantos/reptiledbr`, `https://paulesantos.github.io/reptiledbr/`

BugReports `https://github.com/PaulESantos/reptiledbr/issues`

VignetteBuilder knitr

Maintainer Paul Efren Santos Andrade <`paulefrens@gmail.com`>

Config/pak/sysreqs libicu-dev libxml2-dev

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

Date/Publication 2026-01-07 03:53:23 UTC

RemoteUrl https://github.com/PaulESantos/reptiledbr

RemoteRef HEAD

RemoteSha 73f02af44653ef2c33a4277ed19736badc160143

Contents

`check_reptiledb_update` *Manually check for reptiledb.data updates*

Description

This function checks if there's a newer version of the reptiledb.data package available. It's automatically called when reptiledbr loads, but can also be called manually by users.

Usage

```
check_reptiledb_update(silent = FALSE)
```

Arguments

`silent`	Logical. If TRUE, suppresses messages and returns only the result object. Default is FALSE.

Value

Invisibly returns the update check result object from reptiledb.data::check_data_update(), or NULL if the package is not installed.

Examples

```
# Check for updates with user-friendly messages
check_reptiledb_update()

# Silent check for programmatic use
result <- check_reptiledb_update(silent = TRUE)
if (!is.null(result) && result$needs_update) {
  message("Update available!")
}
```

`extract_attribute` *Extract and optionally format a specific attribute*

Description

[Experimental]

Usage

```
extract_attribute(reptile_data, attribute_name, format_function = NULL)
```

Arguments

`reptile_data`	A data frame obtained using the `get_reptile_species_data()` function.
`attribute_name`	Name of the attribute to extract (e.g., "Distribution", "Synonym").
`format_function`	Optional formatting function for custom processing.

Details

Extracts and optionally formats structured information for a given attribute.

Value

A tibble with formatted information for the specified attribute.

`format_all_attributes` *Format all available reptile attributes*

Description

[Experimental]

Usage

```
format_all_attributes(reptile_data, quiet = FALSE)
```

Arguments

`reptile_data`	A data frame obtained using the `get_reptile_species_data()` function.
`quiet`	Logical. If TRUE, suppresses informational messages. Default is FALSE.

Details

Applies formatting functions to all known attributes in the reptile dataset.

Value

A list containing all formatted attributes.

`format_comments` *Format reptile comment data* **[Experimental]** *Extracts and formats general comments associated with reptile species.*

Description

Format reptile comment data **[Experimental]** Extracts and formats general comments associated with reptile species.

Usage

```
format_comments(reptile_data)
```

Arguments

`reptile_data`	A data frame obtained using the `get_reptile_species_data()` function.

Value

A tibble containing formatted comments for each species.

`format_common_names` *Format common names for reptiles* **[Experimental]**

Description

Format common names for reptiles **[Experimental]**

Usage

```
format_common_names(reptile_data)
```

Arguments

`reptile_data`	A tibble returned by `get_reptile_species_data()`.

Value

A tibble with formatted common names by species.

`format_diagnosis` *Format diagnostic information for reptiles* **[Experimental]**

Description

Format diagnostic information for reptiles **[Experimental]**

Usage

```
format_diagnosis(reptile_data)
```

Arguments

`reptile_data`	A tibble returned by `get_reptile_species_data()`.

Value

A tibble with formatted diagnostic descriptions by species.

`format_distribution` *Format distribution data into a long table format*

Description

[Experimental]

Usage

```
format_distribution(reptile_data)
```

Arguments

`reptile_data`	A tibble returned by `get_reptile_species_data()`.

Value

A tibble with distribution data in long format.

`format_etymology` *Format etymological data of reptiles* **[Experimental]** *Extracts and formats the etymological information associated with reptile species.*

Description

Format etymological data of reptiles **[Experimental]** Extracts and formats the etymological information associated with reptile species.

Usage

```
format_etymology(reptile_data)
```

Arguments

`reptile_data`	A data frame obtained using the `get_reptile_species_data()` function.

Value

A tibble containing the formatted etymological details for each species.

`format_higher_taxa` *Format higher-level taxonomic data for reptiles* **[Experimental]**

Description

Format higher-level taxonomic data for reptiles **[Experimental]**

Usage

```
format_higher_taxa(reptile_data)
```

Arguments

`reptile_data`	A tibble returned by `get_reptile_species_data()`.

Value

A tibble with formatted higher taxonomic classification by species.

`format_references` *Format bibliographic reference data of reptiles* **[Experimental]** *Extracts and formats bibliographic references associated with reptile species.*

Description

Format bibliographic reference data of reptiles **[Experimental]** Extracts and formats bibliographic references associated with reptile species.

Usage

```
format_references(reptile_data)
```

Arguments

`reptile_data`	A data frame obtained using the `get_reptile_species_data()` function.

Value

A tibble containing formatted references for each species.

`format_reproduction` *Format reproductive data for reptiles* **[Experimental]**

Description

Format reproductive data for reptiles **[Experimental]**

Usage

```
format_reproduction(reptile_data)
```

Arguments

`reptile_data`	A tibble returned by `get_reptile_species_data()`.

Value

A tibble with formatted reproductive information by species.

`format_selected_attributes`
Format selected reptile attributes

Description

[Experimental]

Usage

```
format_selected_attributes(reptile_data, attributes, quiet = FALSE)
```

Arguments

`reptile_data`	A data frame obtained using the `get_reptile_species_data()` function.
`attributes`	A character vector specifying which attributes to extract.
`quiet`	Logical. If TRUE, suppresses informational messages. Default is FALSE.

Details

Extracts and formats only the specified attributes from the reptile dataset.

Value

A list containing the selected formatted attributes.

`format_subspecies` *Format subspecies data for reptiles* **[Experimental]**

Description

Format subspecies data for reptiles **[Experimental]**

Usage

```
format_subspecies(reptile_data)
```

Arguments

`reptile_data`	A tibble returned by `get_reptile_species_data()`.

Value

A tibble with formatted subspecies by species.

`format_synonyms` *Format synonym data for reptiles*

Description

[Experimental]

Usage

```
format_synonyms(reptile_data)
```

Arguments

`reptile_data`	A tibble returned by `get_reptile_species_data()`.

Value

A tibble with formatted synonym names by species.

`format_types` *Format nomenclatural type data for reptiles* **[Experimental]**

Description

Format nomenclatural type data for reptiles **[Experimental]**

Usage

```
format_types(reptile_data)
```

Arguments

`reptile_data`	A tibble returned by `get_reptile_species_data()`.

Value

A tibble with formatted type information by species.

`get_reptiledb_data` *Access Reptile Database Taxonomic Information* **[Experimental]**

Description

Retrieves taxonomic information on living reptile species from The Reptile Database. This function allows users to explore scientific names, synonyms, distributions, and taxonomic references for all known species of snakes, lizards, turtles, amphisbaenians, tuataras, and crocodiles.

Usage

```
get_reptiledb_data(species_names, timeout = 10, quiet = FALSE)
```

Arguments

`species_names`	A character string with the scientific name of the species (e.g., "Crocodylus acutus").
`timeout`	Maximum waiting time for each request (in seconds)
`quiet`	Logical value TRUE or FALSE.

Details

The Reptile Database currently includes more than 10,000 species and around 2,800 subspecies. It focuses on taxonomic and nomenclatural data, including valid names, synonyms, type localities, distribution, and original references. However, ecological and behavioral data are largely absent.

Data are compiled from peer-reviewed literature, expert contributions, and curated by an editorial team. Updates and corrections from users are welcome and help improve the resource.

The classification follows recent phylogenetic studies (e.g., Zheng & Wiens, 2016), although the database takes a conservative approach to rapidly changing taxonomic hypotheses. New genera or species proposals may first appear in the "synonyms" field pending wider scientific acceptance.

Note: The database does not support species identification by traits, but users can search by geographical distribution and higher taxonomic groups.

Value

A list or data frame containing available taxonomic information (e.g., accepted name, synonyms, family, distribution, literature references).

Author(s)

Data curated by P. Uetz and collaborators. Function implementation by Paul E. Santos Andrade.

Source

Uetz, P., Freed, P., & Hosek, J. (eds.) (2021). The Reptile Database. Available at: `https://reptile-database.reptarium.cz`

For more on phylogenetic background see: Zheng, Y., & Wiens, J. J. (2016). Combining phylogenomics and fossils in higher-level squamate reptile phylogeny. *BMC Evolutionary Biology*, 16, 1-20.

See Also

`https://reptile-database.reptarium.cz`

Examples

```
get_reptiledb_data("Anolis carolinensis",
                                  quiet = TRUE)
```

`list_subspecies_reptiledbr`

List Subspecies from ReptileDB

Description

This function processes results from a ReptileDB database search to extract subspecies information. It identifies species that have subspecies and returns a tibble with the species name, subspecies name, and author information.

Usage

```
list_subspecies_reptiledbr(df)
```

Arguments

`df`	A dataframe or tibble result from using reptiledbr_exact, reptiledbr_partial or search_reptiledbr functions.

Value

A tibble with three columns:

`species`	The name of the species
`subspecies_name`	The full name of the subspecies
`author`	The author and year of the subspecies description

Examples

```
## Not run:
# These examples require the 'reptiledb.data' package to be installed.
subspecies_names <- c("Lachesis muta",
  "Anilius scytale",
  "Anolis bahorucoensis")

search_reptiledbr(subspecies_names, use_fuzzy = FALSE) |>
  list_subspecies_reptiledbr()

## End(Not run)
```

`reptiledbr_exact` *Search Reptile Species by Exact Match and Subspecies Presence*

Description

This function searches for exact matches of scientific species names and indicates whether each matched species has associated subspecies in the dataset.

Usage

```
reptiledbr_exact(species_names)
```

Arguments

`species_names`	Character vector of full species names to search for.

Value

A tibble with taxonomic information and a message indicating subspecies presence.

The response variable may return different messages depending on the outcome of the query. Possible values include:

- `"Species not found"` – The specified species could not be matched in the database.
- `"Species has subspecies"` – The specified species exists and has one or more subspecies registered.
- `"No subspecies found"` – The species was found, but there are no subspecies associated with it in the database.

Examples

```
## Not run:
# These examples require the 'reptiledb.data' package to be installed.
# You can install it from its source if not on CRAN.
reptiledbr_exact(c("Ablepharus alaicus", "Anolis limon"))

## End(Not run)
```

`reptiledbr_partial` *Fuzzy Search for Species Names Using Approximate Matching*

Description

This function performs approximate (fuzzy) matching of species names from a given list of input terms against the species names in the reptile database, and indicates whether each matched species has subspecies.

Usage

```
reptiledbr_partial(species_names, max_dist = 2)
```

Arguments

`species_names`	Character vector. One or more scientific names or fragments to match approximately.
`max_dist`	Maximum string distance allowed for a match (default: 2).

Value

A tibble with matched species, taxonomic info, fuzzy match flag, and subspecies presence. The response variable may return different messages depending on the outcome of the query. Possible values include:

- `"Species not found"` – The specified species could not be matched in the database.
- `"Species has subspecies"` – The specified species exists and has one or more subspecies registered.
- `"No subspecies found"` – The species was found, but there are no subspecies associated with it in the database.

Examples

```
## Not run:
# These examples require the 'reptiledb.data' package to be installed.
reptiledbr_partial(c("Ablepharus alaicuss", "Anolis limom"))

## End(Not run)
```

`search_reptiledbr` *Comprehensive Search for Reptile Species with Exact and Fuzzy Matching*

Description

This function combines both exact and fuzzy matching approaches to search for reptile species names in the database. It first attempts exact matches and then uses fuzzy matching for any species names that weren't found exactly.

Usage

```
search_reptiledbr(species_names, max_dist = 2, use_fuzzy = TRUE)
```

Arguments

`species_names`	Character vector of scientific species names to search for.
`max_dist`	Maximum string distance allowed for fuzzy matching (default: 2).
`use_fuzzy`	Logical. If TRUE, performs fuzzy search for species not found exactly. If FALSE, only does exact matching (default: TRUE).

Value

A combined tibble with results from both exact and fuzzy matching approaches, with a flag indicating the match type. Results maintain the original order of species_names. The response variable may return different messages depending on the outcome of the query. Possible values include:

- `"Species not found"` – The specified species could not be matched in the database.
- `"Species has subspecies"` – The specified species exists and has one or more subspecies registered.
- `"No subspecies found"` – The species was found, but there are no subspecies associated with it in the database.

Examples

```
## Not run:
# These examples require the 'reptiledb.data' package to be installed.
search_reptiledbr(c("Ablepharus alaicus", "Anolis limom"))

## End(Not run)
```

Index