

# Package ‘taxify’

June 30, 2026

**Title** Offline Taxonomic Name Matching Against Local Darwin Core Snapshots

**Version** 0.2.12

**Description** Match taxonomic names against locally stored Darwin Core backbone databases ('WFO', 'COL', 'GBIF', 'ITIS', 'NCBI Taxonomy', 'Open Tree of Life', 'WoRMS', 'Species Fungorum', 'AlgaeBase'). Provides offline fuzzy and exact matching with synonym resolution, hybrid name detection, and a unified output schema across all sources. All heavy computation runs in the 'vectra' C11 columnar engine.

**License** MIT + file LICENSE

**URL** <https://github.com/gcol33/taxify>

**BugReports** <https://github.com/gcol33/taxify/issues>

**Additional\_repositories** <https://gcol33.r-universe.dev>

**Depends** R (>= 4.1.0)

**Imports** curl, jsonlite, rlang, vectra

**Suggests** DBI, knitr, openxlsx2, rmarkdown, RSQLite, taxifydb, testthat (>= 3.0.0), TR8

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**Config/roxygen2/version** 8.0.0

**NeedsCompilation** no

**Author** Gilles Colling [aut, cre, cph] (ORCID:  
<<https://orcid.org/0000-0003-3070-6066>>)

**Maintainer** Gilles Colling <[gilles.colling051@gmail.com](mailto:gilles.colling051@gmail.com)>

**Repository** CRAN

**Date/Publication** 2026-06-30 11:30:07 UTC

## Contents

add_algae_traits . . . . .	3
add_alien_first_records . . . . .	4
add_amphibio . . . . .	5
add_anage . . . . .	6
add_animaltraits . . . . .	7
add_arthropod_traits . . . . .	8
add_avonet . . . . .	9
add_baseflor . . . . .	10
add_col_info . . . . .	12
add_common_names . . . . .	13
add_conservation_status . . . . .	14
add_data . . . . .	15
add_diaz_traits . . . . .	17
add_ecoflora . . . . .	18
add_eive . . . . .	19
add_elton_traits . . . . .	20
add_fishbase . . . . .	22
add_fish_traits . . . . .	23
add_floraweb . . . . .	24
add_fungal_traits . . . . .	25
add_funguild . . . . .	27
add_gbif_info . . . . .	28
add_glonaf . . . . .	29
add_hybrid_info . . . . .	30
add_invasive_status . . . . .	31
add_leda . . . . .	32
add_leptraits . . . . .	33
add_lizard_traits . . . . .	34
add_pantheria . . . . .	35
add_pignatti . . . . .	36
add_qualifier_info . . . . .	37
add_wcvp . . . . .	38
add_wfo_info . . . . .	39
add_woodiness . . . . .	40
cite . . . . .	41
export_data . . . . .	42
list_enrichments . . . . .	42
lookup_genus . . . . .	43
print.taxify_result . . . . .	43
summary.taxify_result . . . . .	44
taxify . . . . .	44
taxify_clear_cache . . . . .	46
taxify_data_dir . . . . .	47
taxify_download . . . . .	47
taxify_download_enrichment . . . . .	48
taxify_download_vtr . . . . .	49

<code>add_algae_traits</code>	3
<code>taxify_example_data</code>	50
<code>taxify_load_register</code>	50
<code>taxify_long</code>	51
<code>taxify_refresh_manifest</code>	52
<code>taxify_register_coverage</code>	53
<b>Index</b>	<b>54</b>

---

<code>add_algae_traits</code>	<i>Add macroalgal functional traits (AlgaeTraits)</i>
-------------------------------	---

---

### Description

Joins AlgaeTraits (Vranken et al. 2023) macroalgal functional traits to a `taxify()` result by looking up `accepted_name`. AlgaeTraits provides morphological, ecological, and life-history traits for European seaweeds.

### Usage

```
add_algae_traits(x, verbose = TRUE)
```

### Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>verbose</code>	Logical. Default TRUE.

### Details

Source: AlgaeTraits (Vranken et al. 2023, VLIZ Marine Data Archive, CC BY 4.0). Coverage: ~1,745 European macroalgae species.

### Value

The same data.frame with additional columns:

- `algae_body_size_cm`** Maximum body size in centimetres.
- `algae_growth_form`** Growth form / body shape (e.g., filamentous, foliose, crustose).
- `algae_calcification`** Calcification type (e.g., uncalcified, articulated, encrusting).
- `algae_life_span`** Life span category (annual, perennial, etc.).
- `algae_tidal_zone`** Tidal zonation (e.g., supralittoral, eulittoral, sublittoral).
- `algae_wave_exposure`** Wave exposure tolerance (sheltered, moderately exposed, exposed).
- `algae_environment`** Habitat environment (marine, brackish, freshwater).
- `algae_substrate`** Environmental position / substrate type.

### References

Vranken S et al. (2023) AlgaeTraits: a trait database for (European) seaweeds. Earth System Science Data 15:2711-2754. doi:10.5194/essd-15-2711-2023

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Fucus vesiculosus", backend = "gbif") |>
  add_algae_traits()

options(old)
```

---

```
add_alien_first_records
```

*Add alien species first record years*

---

**Description**

Joins alien species first record data to a `taxify()` result, filtered by country. Data from the Global Alien Species First Record Database (Seebens et al. 2017).

**Usage**

```
add_alien_first_records(x, country, verbose = TRUE)
```

**Arguments**

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>country</code>	Character. ISO 3166-1 alpha-2 country code(s), or "all". <ul style="list-style-type: none"> <li>• Single code (e.g., "AT"): adds columns without suffix.</li> <li>• Multiple codes (e.g., c("AT", "DE")): adds columns with country suffix (e.g., <code>alien_first_record_AT</code>).</li> <li>• "all": adds one column set per country in the dataset.</li> </ul>
<code>verbose</code>	Logical. Default TRUE.

**Details**

Source: Global Alien Species First Record Database v3.1 (Seebens et al. 2017, Nature Communications 8, 14435). CC BY 4.0. Coverage: ~77k species x country combinations across all taxa.

**Value**

The same data.frame with additional column(s):

**alien\_first\_record** Year of the first record (integer), or NA if not recorded for that country.

**alien\_first\_record\_source** Database that contributed the record (e.g., "GAVIA", "CABI ISC").

**alien\_first\_record\_reference** Original citation or reference for the record.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Robinia pseudoacacia") |>
  add_alien_first_records(country = "AT")

taxify(c("Robinia pseudoacacia", "Ailanthus altissima")) |>
  add_alien_first_records(country = c("AT", "DE"))

options(old)
```

---

add_amphibio	<i>Add amphibian life-history traits (AmphiBIO)</i>
--------------	---

---

**Description**

Joins AmphiBIO amphibian life-history and ecological traits to a `taxify()` result by looking up `accepted_name`.

**Usage**

```
add_amphibio(x, verbose = TRUE)
```

**Arguments**

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>verbose</code>	Logical. Default TRUE.

**Details**

Source: AmphiBIO (Oliveira et al. 2017, CC BY 4.0). Coverage: ~6,800 amphibian species. Amphibians only.

**Value**

The same data.frame with additional columns:

**body\_size\_mm** Maximum body size in mm (snout-vent length).

**age\_maturity\_d** Age at maturity in days.

**longevity\_d** Maximum longevity in days.

**litter\_size** Clutch/litter size.

**reproductive\_output** Reproductive output per year.

**offspring\_size\_mm** Offspring size in mm.

**direct\_development** Direct development (0/1).

**larval** Has larval stage (0/1).  
**aquatic** Aquatic habitat (0/1).  
**fossorial** Fossorial habitat (0/1).  
**arboreal** Arboreal habitat (0/1).  
**diurnal** Diurnal activity (0/1).  
**nocturnal\_amphibio** Nocturnal activity (0/1). Named nocturnal\_amphibio to avoid collision with EltonTraits' nocturnal column.

## References

Oliveira BF, Sao-Pedro VA, Santos-Barrera G, Penone C, Costa GC (2017) AmphiBIO, a global database for amphibian ecological traits. *Scientific Data* 4:170123.

## Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Bufo bufo", backend = "gbif") |>
  add_amphibio()

options(old)
```

---

add\_anage

*Add longevity and life-history traits (AnAge)*

---

## Description

Joins AnAge (Animal Ageing and Longevity Database) traits to a `taxify()` result by looking up `accepted_name`.

## Usage

```
add_anage(x, verbose = TRUE)
```

## Arguments

`x` A data.frame returned by `taxify()`.  
`verbose` Logical. Default TRUE.

## Details

Source: AnAge (de Magalhaes & Costa 2009, CC BY). Coverage: ~4.7k vertebrate species (mammals, birds, reptiles, amphibians, fish).

## Value

The same data.frame with additional columns:

**max\_longevity\_yr** Maximum longevity in years.  
**anage\_body\_mass\_g** Adult body mass in grams.  
**metabolic\_rate\_w** Basal metabolic rate in watts.  
**female\_maturity\_d** Female age at sexual maturity in days.  
**male\_maturity\_d** Male age at sexual maturity in days.  
**gestation\_incubation\_d** Gestation or incubation length in days.  
**anage\_litter\_size** Litter or clutch size.  
**birth\_mass\_g** Mass at birth in grams.  
**growth\_rate** Growth rate (1/days).  
**temperature\_k** Body temperature in Kelvin.

## References

de Magalhaes JP, Costa J (2009) A database of vertebrate longevity records and their relation to other life-history traits. *Journal of Evolutionary Biology* 22:1770-1774.

## Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Vulpes vulpes", backend = "gbif") |>
  add_anage()

options(old)
```

---

add_animaltraits	<i>Add cross-taxon body mass and metabolic rate (AnimalTraits)</i>
------------------	--

---

## Description

Joins AnimalTraits body mass and metabolic rate data to a `taxify()` result by looking up `accepted_name`.

## Usage

```
add_animaltraits(x, verbose = TRUE)
```

## Arguments

**x** A data.frame returned by `taxify()`.  
**verbose** Logical. Default TRUE.

### Details

Source: AnimalTraits (Hebert et al. 2022, CC0). Coverage: ~2k species across arthropods, vertebrates, molluscs, and annelids. Individual-level observations aggregated to species medians.

### Value

The same data.frame with additional columns:

**animaltraits\_body\_mass\_kg** Median body mass in kg.

**animaltraits\_metabolic\_rate\_w** Median metabolic rate in watts.

### References

Hebert K et al. (2022) AnimalTraits – a curated animal trait database for body mass, metabolic rate and brain size. Scientific Data 9:265.

### Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Drosophila melanogaster", backend = "gbif") |>
  add_animaltraits()

options(old)
```

---

add\_arthropod\_traits *Add arthropod life-history traits (NW European Arthropods)*

---

### Description

Joins the Northwestern European Arthropod Life Histories dataset to a `taxify()` result by looking up `accepted_name`.

### Usage

```
add_arthropod_traits(x, verbose = TRUE)
```

### Arguments

x	A data.frame returned by <code>taxify()</code> .
verbose	Logical. Default TRUE.

### Details

Source: Logghe et al. (2025, CC BY-NC). Coverage: ~4.9k arthropod species from NW Europe across 10 orders (Coleoptera, Hemiptera, Orthoptera, Araneae, Diptera, Hymenoptera, Lepidoptera, etc.).

**Value**

The same data.frame with additional columns:

- arthropod\_body\_size\_mm** Body size in mm.
- arthropod\_dispersal** Dispersal ability (0–1 ratio within order).
- arthropod\_voltinism** Mean number of generations per year.
- arthropod\_fecundity** Fecundity (number of eggs/offspring).
- arthropod\_development\_d** Development time in days.
- arthropod\_lifespan\_d** Adult lifespan in days.
- arthropod\_thermal\_mean** Mean thermal niche (degrees C).
- arthropod\_diurnality** Activity period (diurnal/nocturnal/both).
- arthropod\_feeding\_guild** Feeding guild of adult.
- arthropod\_trophic\_range** Trophic range of adult (specialist/generalist).

**References**

Logghe A et al. (2025) An in-depth dataset of northwestern European arthropod life histories and ecological traits. *Biodiversity Data Journal* 13:e146785.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Abax parallelepipedus", backend = "gbif") |>
  add_arthropod_traits()

options(old)
```

---

add_avonet	<i>Add bird morphology and migration (AVONET)</i>
------------	---

---

**Description**

Joins AVONET species-level averages for bird morphology, ecology, and migration to a [taxify\(\)](#) result by looking up accepted\_name.

**Usage**

```
add_avonet(x, verbose = TRUE)
```

**Arguments**

x	A data.frame returned by <a href="#">taxify()</a> .
verbose	Logical. Default TRUE.

**Details**

Source: AVONET (Tobias et al. 2022, Figshare, CC BY 4.0). Coverage: ~11k bird species. Birds only.

**Value**

The same data.frame with additional columns:

**beak\_length** Beak length in mm (culmen, species mean).

**beak\_depth** Beak depth in mm (species mean).

**wing\_length** Wing length in mm (species mean).

**tail\_length** Tail length in mm (species mean).

**tarsus\_length** Tarsus length in mm (species mean).

**avonet\_body\_mass\_g** Body mass in grams (species mean).

**hand\_wing\_index** Hand-wing index (pointedness, species mean).

**habitat** Primary habitat classification.

**trophic\_level** Trophic level classification.

**trophic\_niche** Trophic niche classification.

**migration** Migration strategy: "sedentary", "partial", or "full".

**References**

Tobias JA et al. (2022) AVONET: morphological, ecological and geographical data for all birds. Ecology Letters 25:581-597.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Parus major", backend = "gbif") |>
  add_avonet()

options(old)
```

---

 add\_baseflor

---

*Add plant traits from Baseflor (Catminat / Julve)*


---

**Description**

Joins Baseflor (Julve, Programme Catminat) plant traits to a `taxify()` result by looking up `accepted_name`. Baseflor covers the vascular flora of France and neighbouring regions, providing flowering phenology, pollination and breeding biology, dispersal mode, and floral and fruit morphology.

**Usage**

```
add_baseflor(x, verbose = TRUE)
```

**Arguments**

**x** A data.frame returned by `taxify()`.  
**verbose** Logical. Default TRUE.

**Details**

Source: Baseflor, Programme Catminat (Julve 1998 ff.). Coverage: ~7,000 vascular plant taxa of France and neighbouring regions. Data are released under ODbL 1.0 / CC BY-SA 2.0.

For ecological indicator values on the light, temperature, moisture, reaction, and nutrient axes, see `add_eive()` (European calibration). For Raunkiaer life form and seed, leaf, and clonality traits of the Northwest European flora, see `add_leda()`.

**Value**

The same data.frame with additional columns:

**flower\_begin\_month** First month of flowering (1-12).

**flower\_end\_month** Last month of flowering (1-12). A value smaller than `flower_begin_month` denotes a flowering period that wraps across the new year (e.g. begin 10, end 6).

**pollination\_vector** Pollination vector(s): insect, wind, water, self, apogamy. Comma-separated when more than one applies.

**dispersal\_mode** Diaspore dispersal mode(s): anemochory, barochory, epizoochory, endozoochory, myrmecochory, hydrochory, autochory, dyszoochory. Comma-separated when more than one applies.

**breeding\_system** Sexual system: hermaphroditic, monoecious, dioecious, gynodioecious, androdioecious, gynomonocious, polygamous.

**flower\_colour** Flower colour(s): white, yellow, pink, green, blue, brown, black. Comma-separated when more than one applies.

**fruit\_type** Fruit type: achene, capsule, caryopsis, drupe, legume, silique, berry, follicle, cone, samara, pyxid.

**woody\_growth\_form** Woody growth form for woody taxa: tree, small tree, large tree, shrub, bush, subshrub, liana, parasite. NA for non-woody (herbaceous) taxa.

**continentality** Ellenberg-style continentality indicator value (1-9), the axis absent from EIVE.

**salinity** Ellenberg-style salinity indicator value (0-9), the axis absent from EIVE.

**References**

Julve, Ph. (1998 ff.) baseflor. Index botanique, ecologique et chorologique de la Flore de France. Programme Catminat.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Bellis perennis") |>
  add_baseflor()

options(old)
```

---

add_col_info	<i>Add COL-specific columns</i>
--------------	---------------------------------

---

**Description**

Joins extra Catalogue of Life columns to a `taxify()` result by looking up `taxon_id` in the COL backbone. Only enriches rows where `backend == "col"`.

**Usage**

```
add_col_info(x)
```

**Arguments**

`x` A data.frame returned by `taxify()` with `backend == "col"`.

**Value**

The same data.frame with additional columns:

**notho** Hybrid type from COL: "generic", "specific", "infrageneric", or "infraspecific".

**nomenclaturalCode** Nomenclatural code ("ICN", "ICZN", etc.).

**nomenclaturalStatus** Nomenclatural status.

**namePublishedIn** Original publication reference.

**kingdom** Kingdom classification.

**phylum** Phylum classification.

**col\_class** Class classification (renamed to avoid conflict with R's `class` function).

**order** Order classification.

**infraspecificEpithet** Infraspecific epithet.

**is\_extinct** Logical. Whether the species is extinct (from SpeciesProfile, if available).

**is\_marine** Logical. Whether the species is marine.

**is\_freshwater** Logical. Whether the species is freshwater.

**is\_terrestrial** Logical. Whether the species is terrestrial.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Quercus robur", backend = "col") |>
  add_col_info()

options(old)
```

---

add_common_names	<i>Add common (vernacular) names</i>
------------------	--------------------------------------

---

**Description**

Joins vernacular names to a `taxify()` result by looking up `accepted_name`, filtered by language.

**Usage**

```
add_common_names(x, lang = "en", verbose = TRUE)
```

**Arguments**

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>lang</code>	Character. ISO 639-1 language code (e.g., "en", "de", "fr"), or NA to return names without a language tag (NCBI/OTT sources). Default "en".
<code>verbose</code>	Logical. Default TRUE.

**Details**

Common names are merged from three sources:

- GBIF backbone vernacular names (CC0) — multi-language via ISO 639-1 codes.
- NCBI Taxonomy common names (public domain) — no language tag (`lang = NA`).
- Open Tree of Life common names (CC0) — no language tag (`lang = NA`).

When multiple common names exist for a species in the requested language, the first (most commonly used) is returned.

**Value**

The same data.frame with an additional column:

**common\_name** The vernacular name in the requested language, or NA if none is available.

## Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Quercus robur") |>
  add_common_names()

taxify("Quercus robur") |>
  add_common_names(lang = "de")

options(old)
```

---

add\_conservation\_status

*Add conservation status*

---

## Description

Joins IUCN Red List conservation status to a `taxify()` result by looking up `accepted_name` in the conservation status enrichment.

## Usage

```
add_conservation_status(x, verbose = TRUE)
```

## Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>verbose</code>	Logical. Show download progress if enrichment data needs to be fetched. Default TRUE.

## Details

Conservation status values are compiled from publicly available sources including GBIF and the IUCN Red List API. Coverage is global across all taxonomic groups (~166k species).

## Value

The same data.frame with an additional column:

**conservation\_status** IUCN category: "LC" (Least Concern), "NT" (Near Threatened), "VU" (Vulnerable), "EN" (Endangered), "CR" (Critically Endangered), "EW" (Extinct in the Wild), "EX" (Extinct), or NA if not assessed.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Panthera tigris", backend = "gbif") |>
  add_conservation_status()

options(old)
```

add\_data

*Add custom data by taxonomic matching***Description**

Joins an external data source (CSV file or data.frame) to a `taxify()` result. Species names in the external data are matched through the same backbone(s) used in the original `taxify()` call, and the join is performed on `accepted_id` — so synonyms in either dataset resolve to the same key.

**Usage**

```
add_data(
  x,
  data,
  species_col = NULL,
  table = NULL,
  sheet = NULL,
  start_row = NULL,
  cols = NULL,
  group_col = NULL,
  groups = "all",
  fuzzy = TRUE,
  fuzzy_threshold = 0.2,
  verbose = TRUE
)
```

**Arguments**

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>data</code>	One of: <ul style="list-style-type: none"> <li>• A <b>data.frame</b> already in R.</li> <li>• A <b>file path</b> to a <code>.csv</code>, <code>.csv.gz</code>, <code>.tsv</code>, <code>.tsv.gz</code>, <code>.xlsx</code>, <code>.sqlite/.db</code>, or <code>.vtr</code> file (read via <code>vectra</code>).</li> </ul>
<code>species_col</code>	Character. Name of the column in <code>data</code> that contains species names. If <code>NULL</code> (default), auto-detected by matching <code>head(10)</code> of each character column against the backbone.

table	Character. Required when data is a SQLite file — the table name to read.
sheet	Integer or character. Sheet to read when data is an .xlsx file. Default NULL (auto-detect the sheet containing species names). Set explicitly to skip auto-detection.
start_row	Integer. Row where column headers begin in an .xlsx file. Default NULL (auto-detect by scanning the first 20 rows for a header row that produces species name matches). Set explicitly when the layout is known.
cols	Character vector of column names from data to join. If NULL (default), all columns except species_col are joined.
group_col	Character or NULL. Column in data that defines groups (e.g., country codes, regions). When set, the output is pivoted to wide format with one column per group (e.g., trait_AT, trait_DE), just like the built-in grouped enrichments. Use <a href="#">taxify_long()</a> to reshape back to long format. Default NULL (flat join, one row per species).
groups	Character vector or "all". Which groups to include when group_col is set. Default "all".
fuzzy	Logical. Enable fuzzy matching for names in data. Default TRUE.
fuzzy_threshold	Numeric. Maximum allowed distance for fuzzy matches. Default 0.2.
verbose	Logical. Default TRUE.

## Details

The workflow:

1. Read data (CSV or data.frame).
2. Identify the species column (explicit or auto-detected).
3. Match species names through the same backbone(s) as the original `taxify()` call, obtaining `accepted_id` for each row.
4. Check for conflicting duplicates: if multiple rows in data resolve to the same `accepted_id` with different values, an error is raised (unless `group_col` is set). Exact duplicates produce a warning and are deduplicated.
5. Left-join on `accepted_id`.

### Grouped data:

When your data has multiple rows per species (e.g., one row per species per country), set `group_col` to produce wide output with suffixed columns. This is the same format as the built-in grouped enrichments.

### Auto-detection:

When `species_col` is not specified, `add_data()` takes the first 10 rows of each character column and runs them through `taxify()`. The column with the highest match rate is selected. If no column achieves at least 50% matches, an error is raised asking the user to specify `species_col` explicitly.

**Value**

The input data.frame with additional columns from data, joined via backbone-resolved accepted\_id. Columns from data that collide with existing columns in x are prefixed with "data\_".

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

result <- taxify(c("Quercus robur", "Pinus sylvestris"))
traits <- data.frame(species = c("Quercus robur", "Pinus sylvestris"),
                    height = c(30, 25))
result |> add_data(traits, species_col = "species")

options(old)
```

---

add_diaz_traits	<i>Add seed mass and plant height (Diaz et al. 2022)</i>
-----------------	--

---

**Description**

Joins species-level mean seed mass and plant height from Diaz et al. (2022) to a `taxify()` result by looking up accepted\_name.

**Usage**

```
add_diaz_traits(x, verbose = TRUE)
```

**Arguments**

x	A data.frame returned by <code>taxify()</code> .
verbose	Logical. Default TRUE.

**Details**

Source: Diaz et al. 2022, TRY File Archive (CC BY 3.0). Coverage: ~46k plant species. Plants only.

**Value**

The same data.frame with additional columns:

**seed\_mass\_mg** Seed mass in milligrams (species-level mean).

**plant\_height\_m** Plant height in metres (species-level mean).

## References

Diaz S et al. (2022) The global spectrum of plant form and function: enhanced species-level trait data. TRY File Archive.

## Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Quercus robur") |>
  add_diaz_traits()

options(old)
```

---

add\_ecoflora

*Add British plant traits from Ecoflora*

---

## Description

Joins traits from the Ecological Flora of the British Isles (Fitter & Peat 1994) to a `taxify()` result by looking up `accepted_name`. Ecoflora covers the vascular flora of the British Isles, providing canopy height, leaf traits, life form, flowering phenology, pollination and reproduction, seed weight, and British-calibrated Ellenberg indicator values. Every column carries a `_uk` suffix to mark the British-flora calibration and to avoid collisions when chained with other plant-trait enrichments (e.g. `add_baseflor()` for France, `add_floraweb()` for Germany).

## Usage

```
add_ecoflora(x, verbose = TRUE)
```

## Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>verbose</code>	Logical. Default TRUE.

## Details

Source: Ecoflora (Ecological Flora of the British Isles). Ecoflora has no bulk download or API; the bundled dataset was collected one species at a time and is redistributed under the source licence (CC BY-NC-SA 4.0). The `.vtr` is downloaded from the taxify release on first use and cached.

For French-flora traits see `add_baseflor()`; for German-flora traits see `add_floraweb()`; for European-calibration indicator values see `add_eive()`.

**Value**

The same data.frame with additional `_uk` columns:

**height\_max\_mm\_uk, height\_min\_mm\_uk** Canopy height range (mm).

**leaf\_area\_uk** Leaf area class.

**leaf\_longevity\_uk** Leaf longevity (e.g. evergreen, deciduous).

**root\_system\_uk** Root system type.

**photosynthetic\_pathway\_uk** Photosynthetic pathway (C3/C4/CAM).

**life\_form\_uk** Raunkiaer life form.

**reproduction\_uk** Reproduction method.

**flower\_begin\_month\_uk, flower\_end\_month\_uk** Flowering months (1-12).

**pollination\_vector\_uk** Pollen vector(s).

**seed\_weight\_mg\_uk** Seed weight (mg).

**propagule\_uk** Propagule / dispersule type.

**ell\_light\_uk, ell\_moisture\_uk, ell\_reaction\_uk, ell\_nitrogen\_uk, ell\_salt\_uk** Ellenberg indicator values calibrated for the British flora (light, moisture, reaction, nitrogen, salt).

**References**

Fitter AH, Peat HJ (1994) The Ecological Flora Database. *Journal of Ecology* 82:415-425.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Bellis perennis") |>
  add_ecoflora()

options(old)
```

---

add\_eive

*Add EIVE ecological indicator values*

---

**Description**

Joins EIVE 1.0 (Dengler et al. 2023) ecological indicator values to a `taxify()` result by looking up `accepted_name`. EIVE provides continuous indicator values for European vascular plants, superseding the original ordinal Ellenberg values.

**Usage**

```
add_eive(x, verbose = TRUE)
```

## Arguments

`x` A data.frame returned by `taxify()`.  
`verbose` Logical. Default TRUE.

## Details

Source: EIVE 1.0 (Dengler et al. 2023, Zenodo, CC BY 4.0). Coverage: ~14.5k European vascular plant species.

## Value

The same data.frame with additional columns:

**eive\_light** Light indicator value (continuous).  
**eive\_temperature** Temperature indicator value (continuous).  
**eive\_moisture** Moisture indicator value (continuous).  
**eive\_reaction** Soil reaction (pH) indicator value (continuous).  
**eive\_nutrients** Nutrient indicator value (continuous).

## References

Dengler J et al. (2023) EIVE 1.0 – a standardized set of Ecological Indicator Values for Europe. Vegetation Classification and Survey 4:7-29. doi:10.3897/VCS.98324

## Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Arrhenatherum elatius") |>
  add_eive()

options(old)
```

---

add\_elton\_traits      *Add diet, foraging, and body mass (EltonTraits 1.0)*

---

## Description

Joins EltonTraits 1.0 diet composition, foraging strata, body mass, and activity data to a `taxify()` result by looking up `accepted_name`.

## Usage

```
add_elton_traits(x, verbose = TRUE)
```

**Arguments**

`x` A data.frame returned by `taxify()`.  
`verbose` Logical. Default TRUE.

**Details**

Source: EltonTraits 1.0 (Wilman et al. 2014, Figshare, CC0). Coverage: ~15.4k species. Birds and mammals only.

**Value**

The same data.frame with additional columns:

**diet\_inv** Percentage of diet: invertebrates.  
**diet\_vend** Percentage of diet: endothermic vertebrates.  
**diet\_vect** Percentage of diet: ectothermic vertebrates.  
**diet\_vfish** Percentage of diet: fish.  
**diet\_vunk** Percentage of diet: unknown vertebrates.  
**diet\_scav** Percentage of diet: scavenging.  
**diet\_fruit** Percentage of diet: fruit.  
**diet\_nect** Percentage of diet: nectar.  
**diet\_seed** Percentage of diet: seeds and nuts.  
**diet\_plantother** Percentage of diet: other plant material.  
**foraging\_water** Percentage of foraging: below water surface.  
**foraging\_ground** Percentage of foraging: on ground.  
**foraging\_understory** Percentage of foraging: in understory.  
**foraging\_midhigh** Percentage of foraging: in mid to high strata.  
**foraging\_canopy** Percentage of foraging: in canopy.  
**foraging\_aerial** Percentage of foraging: aerial.  
**elton\_body\_mass\_g** Body mass in grams.  
**nocturnal** Nocturnal activity (0 = diurnal, 1 = nocturnal).

**References**

Wilman H et al. (2014) EltonTraits 1.0: Species-level foraging attributes of the world's birds and mammals. *Ecology* 95:2027.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Parus major", backend = "gbif") |>
  add_elton_traits()

options(old)
```

---

add_fishbase	<i>Add fish traits (FishBase)</i>
--------------	-----------------------------------

---

## Description

Joins FishBase morphological and ecological traits to a `taxify()` result by looking up `accepted_name`.

## Usage

```
add_fishbase(x, verbose = TRUE)
```

## Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>verbose</code>	Logical. Default TRUE.

## Details

Source: FishBase via `rfishbase` (Froese & Pauly, CC BY-NC 3.0). Coverage: ~35k fish species. Fishes only.

The build-from-source fallback requires the **`rfishbase`** package (available on CRAN). Pre-built `.vtr` files do not require `rfishbase`.

## Value

The same data.frame with additional columns:

**`fb_body_length_cm`** Maximum body length in centimetres.

**`fb_body_mass_g`** Body mass in grams (estimated from length-weight relationships where available).

**`fb_trophic_level`** Trophic level.

**`fb_depth_min_m`** Minimum depth in metres.

**`fb_depth_max_m`** Maximum depth in metres.

**`fb_vulnerability`** Vulnerability index (0–100).

**`fb_habitat`** Habitat type (e.g. demersal, pelagic).

**`fb_importance`** Commercial importance category.

## References

Froese R, Pauly D (eds.) (2024) FishBase. World Wide Web electronic publication, <https://www.fishbase.org>.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Gadus morhua", backend = "gbif") |>
  add_fishbase()

options(old)
```

---

add_fish_traits	<i>Add freshwater fish morphological traits (FISHMORPH)</i>
-----------------	---

---

**Description**

Joins FISHMORPH morphological trait data to a `taxify()` result by looking up `accepted_name`.

**Usage**

```
add_fish_traits(x, verbose = TRUE)
```

**Arguments**

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>verbose</code>	Logical. Default TRUE.

**Details**

Source: FISHMORPH (Brosse et al. 2021, Figshare, CC BY 4.0). Coverage: ~8.3k freshwater fish species.

**Value**

The same data.frame with additional columns:

- fish\_max\_body\_length** Maximum body length (cm).
- fish\_body\_elongation** Body elongation (body length / body depth).
- fish\_vertical\_eye\_position** Vertical eye position (eye position / head depth).
- fish\_relative\_eye\_size** Relative eye size (eye diameter / head length).
- fish\_oral\_gape\_position** Oral gape position (mouth position: 0 = inferior, 0.5 = terminal, 1 = superior).
- fish\_relative\_maxillary\_length** Relative maxillary length (maxillary length / head length).
- fish\_body\_lateral\_shape** Body lateral shape (body depth / caudal peduncle depth).
- fish\_pectoral\_fin\_position** Pectoral fin vertical position (fin insertion depth / body depth).
- fish\_pectoral\_fin\_size** Pectoral fin size (fin length / body length).
- fish\_caudal\_peduncle\_throttling** Caudal peduncle throttling (caudal peduncle depth / caudal fin depth).

## References

Brosse S, Charpin N, Su G, Toussaint A, Herrera-R GA, Tedesco PA, Vilgert S (2021) FISH-MORPH: A global database on morphological traits of freshwater fishes. *Global Ecology and Biogeography* 30:2330-2336. doi:10.1111/geb.13395

## Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Salmo trutta", backend = "gbif") |>
  add_fish_traits()

options(old)
```

---

add\_floraweb

*Add German plant traits from FloraWeb*

---

## Description

Joins traits from FloraWeb (Bundesamt fuer Naturschutz) to a `taxify()` result by looking up `accepted_name`. FloraWeb is the live national portal carrying the BiolFlor trait data (Klotz, Kuehn & Durka 2002) together with Rothmaler morphology and Ellenberg indicator values. This enrichment covers the full per-species trait profile scraped from the four FloraWeb trait pages: morphology, reproductive biology, the nine Ellenberg indicator values, ploidy and chromosome number, and chorological distribution. Every column carries a `_de` suffix to mark the German-flora calibration and to avoid collisions when chained with other plant-trait enrichments (e.g. `add_ecoflora()` for Britain, `add_baseflor()` for France).

## Usage

```
add_floraweb(x, verbose = TRUE)
```

## Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>verbose</code>	Logical. Default TRUE.

## Details

Source: FloraWeb (<https://www.floraweb.de/>), Bundesamt fuer Naturschutz, Bonn. FloraWeb has no bulk export or API; the bundled dataset was scraped per species (accessed 2026-06-24) and that access date is the dataset version. The trait data largely derive from BiolFlor, which per the BioFresh metadata statement is publicly available and may be used without restrictions provided it is acknowledged and cited correctly. The `.vtr` is downloaded from the taxify release on first use and cached.

For British-flora traits see `add_ecoflora()`; for French-flora traits see `add_baseflor()`; for European-calibration indicator values see `add_eive()`.

**Value**

The same data.frame with German trait columns (all suffixed `_de`), grouped as:

**Morphology** `height_de, life_form_de, leaf_shape_de, leaf_anatomy_de, leaf_persistence_de, storage_organ_de, flowering_months_de, flowering_months_biolflor_de, flowering_phase_de, phenological_season_de, description_de.`

**Reproductive biology** `pollination_vector_de, pollinator_de, pollinator_reward_de, flower_type_de, flower_class_de, dispersal_type_de, diaspore_type_de, germinule_type_de, reproduction_type_de, vegetative_spread_de, fertilization_type_de, apomixis_de, dicliny_de, dichogamy_de, self_incompatibility_de, si_mechanism_de, ploidy_de, chromosome_number_de, chromosome_freq_de, chromosomes_de.`

**Ecology** the nine Ellenberg indicator values `ell_light_de, ell_temperature_de, ell_continentality_de, ell_moisture_de, ell_moisture_variability_de, ell_reaction_de, ell_nitrogen_de, ell_salt_de, heavy_metal_resistance_de, plus_strategy_type_de` (Grime CSR), `habitat_site_de, formation_de, plant_community_de, biotope_type_de, forest_binding_de, hemeroby_de, urbanity_de.`

**Distribution** `floristic_zones_de, areal_formula_de, areal_type_de, oceanity_de, range_centre_de, world_range_size_de, world_range_frequency_de, world_range_position_de, world_range_hazard_de, germany_range_share_de, germany_responsibility_de.`

Categorical traits with several applicable values are joined with "; ". Trait values are German (as published by FloraWeb / BiolFlor).

**References**

Klotz S, Kuehn I, Durka W (2002) BIOLFLOR - Eine Datenbank zu biologisch-oekologischen Merkmalen der Gefaesspflanzen in Deutschland. Schriftenreihe fuer Vegetationskunde 38. Bundesamt fuer Naturschutz, Bonn.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Bellis perennis") |>
  add_floraweb()

options(old)
```

---

add\_fungal\_traits      *Add fungal lifestyle and trait data (FungalTraits)*

---

**Description**

Joins FungalTraits (Polme et al. 2020) genus-level trait data to a `taxify()` result by looking up genus. Unlike other enrichments that join on species-level `accepted_name`, FungalTraits is a genus-level database and joins on the genus column already present in `taxify` output.

**Usage**

```
add_fungal_traits(x, verbose = TRUE)
```

**Arguments**

x	A data.frame returned by <code>taxify()</code> .
verbose	Logical. Default TRUE.

**Details**

Source: FungalTraits (Polme et al. 2020, Fungal Diversity, CC BY 4.0). Coverage: ~10k fungal genera. Genus-level only (not species-level).

**Value**

The same data.frame with additional columns:

**primary\_lifestyle** Primary ecological role (e.g., saprotroph, mycorrhizal, pathogen, endophyte, lichenized, parasite).

**secondary\_lifestyle** Secondary ecological role, if any.

**growth\_form** Morphological growth form (e.g., agaricoid, corticioid, polyporoid, yeast).

**fruitbody\_type** Fruiting body morphology (e.g., gasteroid, pileate, resupinate).

**decay\_substrate** Substrate type for saprotrophic genera (e.g., wood, litter, dung, soil).

**plant\_pathogenic\_capacity** Capacity to cause plant disease (e.g., high, medium, low, none).

**animal\_biotrophic\_capacity** Capacity for animal biotrophy.

**endophytic\_interaction\_capability** Capacity for endophytic interactions with plants.

**ectomycorrhiza\_exploration\_type** Exploration type for ectomycorrhizal genera (e.g., contact, short, medium, long).

**References**

Polme S et al. (2020) FungalTraits: a user-friendly traits database of fungi and fungus-like stramenopiles. Fungal Diversity 105:1-16. doi:10.1007/s13225-020-00466-2

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Amanita muscaria", backend = "gbif") |>
  add_fungal_traits()

options(old)
```

---

add_funguild	<i>Add fungal functional guild data (FUNGuild)</i>
--------------	--

---

### Description

Joins FUNGuild trophic mode, guild, growth morphology, and confidence data to a `taxify()` result by looking up `accepted_name`. Species-level matches take priority; genus-level guild assignments are used as fallback for unmatched species.

### Usage

```
add_funguild(x, verbose = TRUE)
```

### Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>verbose</code>	Logical. Default TRUE.

### Details

Source: FUNGuild (Nguyen et al. 2016, CC BY 4.0). Coverage: ~13k taxa. Fungi only.

The enrichment first attempts species-level matching. For species without a direct match, it falls back to genus-level guild assignments from FUNGuild's genus-rank entries.

### Value

The same data.frame with additional columns:

**trophic\_mode** Trophic mode (e.g., Pathotroph, Saprotroph, Symbiotroph, or hyphenated combinations).

**guild** Functional guild (e.g., "Ectomycorrhizal", "Plant Pathogen", "Wood Saprotroph").

**funguild\_growth\_form** Growth morphology (e.g., "Agaricoid", "Microfungus"). Prefixed to avoid collision with FungalTraits.

**confidence\_ranking** Confidence of the guild assignment (Possible, Probable, Highly Probable).

### References

Nguyen NH et al. (2016) FUNGuild: An open annotation tool for parsing fungal community datasets by ecological guild. *Fungal Ecology* 20:241-248.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Amanita muscaria", backend = "gbif") |>
  add_funguild()

options(old)
```

---

add_gbif_info	<i>Add GBIF-specific columns</i>
---------------	----------------------------------

---

**Description**

Joins extra GBIF backbone columns to a `taxify()` result by looking up `taxon_id` in the GBIF backbone. Only enriches rows where `backend == "gbif"`.

**Usage**

```
add_gbif_info(x)
```

**Arguments**

`x` A data.frame returned by `taxify()` with `backend == "gbif"`.

**Value**

The same data.frame with additional columns:

**notho\_type** Hybrid type: "GENERIC", "SPECIFIC", or "INFRASPECIFIC".

**nom\_status** Nomenclatural status (may contain multiple values).

**bracket\_authorship** Basionym author in parentheses.

**bracket\_year** Basionym author year.

**gbif\_year** Combining author year.

**name\_published\_in** Publication citation.

**origin** How the name entered the backbone.

**infra\_specific\_epithet** Intraspecific epithet.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Quercus robur", backend = "gbif") |>
  add_gbif_info()

options(old)
```

---

add_glonaf	<i>Add naturalized alien flora status (GloNAF)</i>
------------	--

---

### Description

Joins GloNAF (Global Naturalized Alien Flora) data to a `taxify()` result, filtered by region.

### Usage

```
add_glonaf(x, region, verbose = TRUE)
```

### Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>region</code>	Character. GloNAF region identifier(s), or "all". Regions use TDWG-compatible codes extended with dot notation for sub-national units (e.g., "USA.CA" for California). <ul style="list-style-type: none"> <li>• Single region: adds naturalized column (no suffix).</li> <li>• Multiple regions: adds naturalized_&lt;region&gt; columns.</li> <li>• "all": adds one column per region in the dataset.</li> </ul>
<code>verbose</code>	Logical. Default TRUE.

### Details

Source: GloNAF v2.0 (van Kleunen et al. 2019, Davis et al. 2025, CC BY 4.0). Coverage: ~16k alien plant taxa across ~1,300 regions. Plants only.

### Value

The same data.frame with additional column(s):

**naturalized** Integer 1 if the species is recorded as naturalized in that region, NA otherwise.

### References

van Kleunen M et al. (2019) The Global Naturalized Alien Flora (GloNAF) database. *Ecology* 100:e02542.

Davis K et al. (2025) The updated Global Naturalized Alien Flora (GloNAF 2.0) database. *Ecology*, e70245.

### Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Robinia pseudoacacia") |>
  add_glonaf(region = "EUR")

taxify("Robinia pseudoacacia") |>
  add_glonaf(region = c("EUR", "NAM"))

options(old)
```

---

add_hybrid_info	<i>Add hybrid parent and type information</i>
-----------------	---

---

### Description

Parses the `input_name` column from a `taxify()` result to extract hybrid parent names and classify the hybrid type.

### Usage

```
add_hybrid_info(x)
```

### Arguments

`x` A data.frame returned by `taxify()`.

### Value

The same data.frame with additional columns:

**hybrid\_parent\_1** First parent (full binomial), NA if not a hybrid formula.

**hybrid\_parent\_2** Second parent (full binomial, abbreviated genus expanded), NA if not a hybrid formula.

**hybrid\_type** One of "nothogenus", "nothospecies", "formula", or NA if not a hybrid.

### Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Quercus pyrenaica x Q. petraea") |>
  add_hybrid_info()

options(old)
```

---

add\_invasive\_status    *Add invasive species status*

---

## Description

Joins GRIIS (Global Register of Introduced and Invasive Species) data to a `taxify()` result, filtered by country.

## Usage

```
add_invasive_status(x, country, verbose = TRUE)
```

## Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>country</code>	Character. ISO 3166-1 alpha-2 country code(s), or "all". <ul style="list-style-type: none"><li>• Single code (e.g., "AT"): adds <code>invasive_status</code> column (no suffix).</li><li>• Multiple codes (e.g., <code>c("AT", "DE")</code>): adds <code>invasive_status_AT</code>, <code>invasive_status_DE</code>.</li><li>• "all": adds one column per country in the dataset.</li></ul>
<code>verbose</code>	Logical. Default TRUE.

## Details

Source: GRIIS (Zenodo combined CSV, CC BY 4.0, 196 countries). Coverage: ~23k name x country combinations.

## Value

The same data.frame with additional column(s):

**invasive\_status** One of "native", "introduced", "invasive", or NA if not recorded for that country.

## Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Robinia pseudoacacia") |>
  add_invasive_status(country = "AT")

taxify("Robinia pseudoacacia") |>
  add_invasive_status(country = c("AT", "DE"))

options(old)
```

---

 add\_leda

*Add plant traits from LEDA Traitbase*


---

### Description

Joins LEDA Traitbase (Kleyer et al. 2008) plant functional traits to a `taxify()` result by looking up `accepted_name`. LEDA provides species-level trait data for NW European plant species, covering life form, dispersal, seed, leaf, and clonality traits.

### Usage

```
add_leda(x, verbose = TRUE)
```

### Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>verbose</code>	Logical. Default TRUE.

### Details

Source: LEDA Traitbase (Kleyer et al. 2008). Coverage: ~8,000 NW European plant species.

The Raunkiaer life form is a bud-position classification system: phanerophyte = buds >25 cm above soil, chamaephyte = buds near soil surface, hemicryptophyte = buds at soil surface, geophyte (cryptophyte) = buds below soil, therophyte = annual that survives as seed.

### Value

The same data.frame with additional columns:

**raunkiaer\_life\_form** Primary Raunkiaer life form classification (phanerophyte, chamaephyte, hemicryptophyte, geophyte, therophyte, helophyte, hydrophyte).

**raunkiaer\_variable** 1 if species assigned to multiple Raunkiaer forms, 0 otherwise.

**dispersal\_type** Primary dispersal type (anemochory, zoochory, hydrochory, autochory, barochory, dysochory).

**terminal\_velocity\_ms** Seed terminal velocity in m/s (species median).

**seed\_mass\_mg** Seed mass in mg (species median). Prefixed with `leda_` in the `.vtr` to avoid collision with Diaz traits.

**canopy\_height\_m** Canopy height in meters (species median).

**leaf\_mass\_mg** Leaf dry mass in mg (species median).

**sla\_mm2\_mg** Specific leaf area in  $\text{mm}^2/\text{mg}$  (species median).

**clonal\_growth** Capable of clonal growth (1 = yes, 0 = no).

**buoyancy** Seed buoyancy classification.

## References

Kleyer M et al. (2008) The LEDA Traitbase: a database of life-history traits of the Northwest European flora. *Journal of Ecology* 96:1266-1274.

## Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Arrhenatherum elatius") |>
  add_leda()

options(old)
```

---

add_leptraits	<i>Add butterfly traits (LepTraits)</i>
---------------	---

---

## Description

Joins LepTraits 1.0 butterfly life-history and ecological traits to a `taxify()` result by looking up `accepted_name`.

## Usage

```
add_leptraits(x, verbose = TRUE)
```

## Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>verbose</code>	Logical. Default TRUE.

## Details

Source: LepTraits 1.0 (Shirey et al. 2022, CC0). Coverage: ~12.4k butterfly species globally (Papilionoidea).

## Value

The same data.frame with additional columns:

**wingspan\_mm** Wingspan in mm (midpoint of lower and upper bounds).

**voltinism** Number of generations per year.

**diapause\_stage** Overwintering/diapause life stage.

**canopy\_affinity** Canopy association category.

**edge\_affinity** Edge/gap affinity category.

**moisture\_affinity** Moisture affinity category.  
**disturbance\_affinity** Disturbance affinity category.  
**n\_hostplant\_families** Number of host plant families used.  
**flight\_months** Number of months with adult flight activity.

## References

Shirey V et al. (2022) LepTraits 1.0: A globally comprehensive dataset of butterfly traits. *Scientific Data* 9:398.

## Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Vanessa cardui", backend = "gbif") |>
  add_leptraits()

options(old)
```

---

add\_lizard\_traits      *Add lizard life-history and ecological traits (Meiri 2018)*

---

## Description

Joins lizard trait data from Meiri (2018) to a `taxify()` result by looking up `accepted_name`.

## Usage

```
add_lizard_traits(x, verbose = TRUE)
```

## Arguments

x	A data.frame returned by <code>taxify()</code> .
verbose	Logical. Default TRUE.

## Details

Source: Meiri (2018, *Global Ecology and Biogeography*, CC BY 4.0). Coverage: ~6,600 lizard species. Lizards only.

**Value**

The same data.frame with additional columns:

**lizard\_body\_mass\_g** Body mass in grams.

**lizard\_svl\_mm** Snout-vent length in mm.

**lizard\_tail\_length\_mm** Tail length in mm.

**lizard\_clutch\_size** Clutch size.

**lizard\_clutch\_frequency** Clutches per year.

**lizard\_longevity\_yr** Maximum longevity in years.

**lizard\_diet** Diet category.

**lizard\_habitat** Habitat type.

**lizard\_activity\_time** Activity time (diurnal/nocturnal/crepuscular).

**lizard\_foraging\_mode** Foraging mode (sit-and-wait/active).

**References**

Meiri S (2018) Traits of lizards of the world: Variation around a successful evolutionary design. *Global Ecology and Biogeography* 27:1168-1172.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Pogona vitticeps", backend = "gbif") |>
  add_lizard_traits()

options(old)
```

---

add_pantheria	<i>Add mammal life-history traits (PanTHERIA)</i>
---------------	---

---

**Description**

Joins PanTHERIA mammal life-history and ecological traits to a `taxify()` result by looking up `accepted_name`.

**Usage**

```
add_pantheria(x, verbose = TRUE)
```

**Arguments**

x	A data.frame returned by <code>taxify()</code> .
verbose	Logical. Default TRUE.

**Details**

Source: PanTHERIA (Jones et al. 2009, Ecological Archives, CC0). Coverage: ~5.4k mammal species. Mammals only.

**Value**

The same data.frame with additional columns:

**pantheria\_body\_mass\_g** Adult body mass in grams.

**longevity\_mo** Maximum longevity in months.

**litter\_size** Litter size (mean).

**gestation\_d** Gestation length in days.

**weaning\_d** Weaning age in days.

**home\_range\_km2** Home range size in km<sup>2</sup>.

**diet\_breadth** Diet breadth (number of diet categories).

**habitat\_breadth** Habitat breadth (number of habitat types).

**References**

Jones KE et al. (2009) PanTHERIA: a species-level database of life history, ecology, and geography of extant and recently extinct mammals. *Ecology* 90:2648.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Vulpes vulpes", backend = "gbif") |>
  add_pantheria()

options(old)
```

---

 add\_pignatti

---

*Add Italian plant traits from Pignatti (on demand, via TR8)*


---

**Description**

Fetches Italian Ellenberg-type indicator values, life form, and chorotype from Pignatti's *Flora d'Italia* (Pignatti, Menegoni & Pietrosanti 2005) for the species in a `taxify()` result, using the TR8 package, and joins them by `accepted_name`. TR8 ships these values bundled, so this works offline.

**Usage**

```
add_pignatti(x, verbose = TRUE)
```

**Arguments**

`x` A data.frame returned by `taxify()`.  
`verbose` Logical. Default TRUE.

**Details**

These values originate in a copyrighted publication, so `taxify` does not redistribute them. This function reads the copy bundled in the suggested package TR8 (which redistributes it under TR8's GPL, with attribution); `taxify` ships none of it and no internet access is required. For European-calibration indicator values see `add_eive()`.

**Value**

The same data.frame with additional columns:

**light\_it, temperature\_it, continentality\_it, moisture\_it, reaction\_it, nutrients\_it, salinity\_it** Ellenberg-type indicator values calibrated for the Italian flora (codes; X = indifferent, 0 = not applicable).

**life\_form\_it** Life form for the Italian flora.

**chorotype\_it** Chorological type (distribution).

**References**

Pignatti S, Menegoni P, Pietrosanti S (2005) Bioindicazione attraverso le piante vascolari. *Braun-Blanquetia* 39. Bocci G (2015) TR8: an R package for easily retrieving plant species traits. *Methods in Ecology and Evolution* 6:347-350.

**Examples**

```
old <- options(taxify.data_dir = taxify_example_data())

# add_pignatti() fetches Italian trait data on demand via the TR8 package.
taxify("Abies alba") |>
  add_pignatti()

options(old)
```

---

add\_qualifier\_info      *Add qualifier information*

---

**Description**

Parses the `input_name` column from a `taxify()` result to extract taxonomic qualifiers (cf., aff., s.l., etc.) and their positions.

**Usage**

```
add_qualifier_info(x)
```

**Arguments**

x                    A data.frame returned by `taxify()`.

**Value**

The same data.frame with additional columns:

**qualifier** The qualifier found (e.g., "cf. ", "aff. "), or NA if none.

**qualifier\_position** Integer position (character index) of the qualifier in the original name, or NA if none.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Pinus cf. sylvestris") |>
  add_qualifier_info()

options(old)
```

---

add_wcvp	<i>Add WCVP native range status</i>
----------	-------------------------------------

---

**Description**

Joins WCVP (World Checklist of Vascular Plants, Kew) native range data to a `taxify()` result, filtered by TDWG botanical region.

**Usage**

```
add_wcvp(x, region, verbose = TRUE)
```

**Arguments**

x                    A data.frame returned by `taxify()`.

region               Character. TDWG Level 2 region code(s), or "all".

- Single code (e.g., "EUR"): adds `native_status` column (no suffix).
- Multiple codes (e.g., `c("EUR", "NAM")`): adds `native_status_EUR`, `native_status_NAM`.
- "all": adds one column per region in the dataset.

verbose              Logical. Default TRUE.

**Details**

Source: WCVP (Kew, CC BY). Coverage: ~340k plant species. Plants only.

**Value**

The same data.frame with additional column(s):

**native\_status** One of "native", "introduced", "extinct", or NA if not recorded for that region.

**Examples**

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Quercus robur") |>
  add_wcvp(region = "EUR")

taxify("Quercus robur") |>
  add_wcvp(region = c("EUR", "NAM"))

options(old)
```

---

add_wfo_info	<i>Add WFO-specific columns</i>
--------------	---------------------------------

---

**Description**

Joins extra World Flora Online columns to a `taxify()` result by looking up `taxon_id` in the WFO backbone.

**Usage**

```
add_wfo_info(x)
```

**Arguments**

`x` A data.frame returned by `taxify()` with `backend == "wfo"`.

**Value**

The same data.frame with additional columns:

**scientificNameID** WFO scientificNameID.

**parentNameUsageID** WFO parentNameUsageID.

**namePublishedIn** Publication reference.

**higherClassification** Higher classification string.

**taxonRemarks** Taxonomic remarks.

**infraspecificEpithet** Infraspecific epithet (for subspecies, varieties, forms).

## Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Quercus robur") |>
  add_wfo_info()

options(old)
```

---

add_woodiness	<i>Add woodiness classification</i>
---------------	-------------------------------------

---

## Description

Joins woodiness data from Zanne et al. (2014) to a `taxify()` result by looking up `accepted_name`.

## Usage

```
add_woodiness(x, verbose = TRUE)
```

## Arguments

x	A data.frame returned by <code>taxify()</code> .
verbose	Logical. Default TRUE.

## Details

Source: Zanne et al. 2014, Nature (Dryad, CC0). Coverage: ~50k plant species. Plants only.

## Value

The same data.frame with an additional column:

**woodiness** One of "woody", "herbaceous", "variable", or NA if not in the dataset.

## References

Zanne AE et al. (2014) Three keys to the radiation of angiosperms into freezing environments. Nature 506:89-92.

## Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

taxify("Quercus robur") |>
  add_woodiness()

options(old)
```

---

cite

*Cite data sources used in a taxify result*

---

## Description

Prints formatted citations for the taxonomic backbone(s), enrichment layers, and the taxify package itself. Optionally writes a BibTeX file.

## Usage

```
cite(x, file = NULL)
```

## Arguments

x	A taxify_result object.
file	Optional file path. If provided, BibTeX entries are written to this file (extension should be .bib).

## Value

x, invisibly (pipe-friendly).

## Examples

```
result <- taxify("Quercus robur", backend = "wfo")
result |> cite()
result |> cite(file = tempfile(fileext = ".bib"))
```

---

export_data	<i>Export a taxify result to file</i>
-------------	---------------------------------------

---

### Description

Writes a `taxify()` result (with any enrichments) to disk in one of several formats. The default `.vtr` format preserves column types and is fast to re-read with `add_data()`.

### Usage

```
export_data(x, path, overwrite = FALSE)
```

### Arguments

x	A data.frame returned by <code>taxify()</code> .
path	Character. Output file path. The format is inferred from the extension: <code>.vtr</code> , <code>.csv</code> , <code>.tsv</code> , or <code>.xlsx</code> .
overwrite	Logical. Overwrite an existing file? Default FALSE.

### Value

Invisibly returns path.

### Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

result <- taxify(c("Quercus robur", "Pinus sylvestris"))
result |> export_data(tempfile(fileext = ".vtr"))
result |> export_data(tempfile(fileext = ".csv"))
result |> export_data(tempfile(fileext = ".tsv"))

options(old)
```

---

list_enrichments	<i>List available enrichments</i>
------------------	-----------------------------------

---

### Description

Returns a summary of all enrichment layers available in the taxify manifest, including version, row count, whether the dataset is static, and which trait columns are provided.

### Usage

```
list_enrichments(verbose = TRUE)
```

**Arguments**

verbose            Logical. Default TRUE.

**Value**

A data.frame with columns: name, version, nrow, static, trait\_cols (comma-separated), and source\_url.

**Examples**

```
list_enrichments()
```

---

lookup_genus	<i>Look up a genus in the register</i>
--------------	--

---

**Description**

Returns the register row for the given genus, or NULL if not found. Auto-loads the register on first call.

**Usage**

```
lookup_genus(genus)
```

**Arguments**

genus            Character scalar. The genus name to look up.

**Value**

A one-row data.frame, or NULL if the genus is not in the register.

---

print.taxify_result	<i>Print a taxify_result</i>
---------------------	------------------------------

---

**Description**

Delegates to the standard data.frame print method.

**Usage**

```
## S3 method for class 'taxify_result'
print(x, ...)
```

**Arguments**

x                    A `taxify_result` object.  
 ...                  Passed to the next method.

**Value**

x, invisibly.

---

`summary.taxify_result` *Summarise a taxify\_result*

---

**Description**

Prints a compact digest of match quality and life-form scope to the console. Uses `cat()` so output is captured by `capture.output()` and rendered correctly in knitr chunks.

**Usage**

```
## S3 method for class 'taxify_result'
summary(object, ...)
```

**Arguments**

object              A `taxify_result` object.  
 ...                  Ignored.

**Value**

object, invisibly.

---

`taxify`                    *Match taxonomic names against local backbone databases*

---

**Description**

Matches a vector of taxonomic names against locally stored Darwin Core backbone databases. Returns a `data.frame` with one row per input name containing the matched name, accepted name, taxonomic hierarchy, and match quality information.

**Usage**

```
taxify(
  x,
  backend = "wfo",
  fuzzy = TRUE,
  fuzzy_threshold = 0.2,
  fuzzy_method = c("d1", "levenshtein", "jw"),
  verbose = TRUE
)
```

**Arguments**

x	Character vector of taxonomic names.
backend	Character vector of backend names (e.g., "wfo", "col", "gbif") or a single taxify_backend object. When multiple backends are given, they are tried in order as a fallback chain. Default "wfo".
fuzzy	Logical. Enable fuzzy matching for names that fail exact match. Default TRUE.
fuzzy_threshold	Numeric. Maximum allowed distance for fuzzy matches. Two modes depending on the value: <ul style="list-style-type: none"> <li>• <b>Fractional</b> (<math>0 &lt; \text{fuzzy\_threshold} &lt; 1</math>): normalized distance (edits / max name length). Default 0.2 is about 1 edit per 5 characters.</li> <li>• <b>Integer</b> (<math>\text{fuzzy\_threshold} \geq 1</math>): maximum raw edit count, e.g. fuzzy_threshold = 2L allows at most 2 insertions/deletions/substitutions regardless of name length. Not supported for fuzzy_method = "jw".</li> </ul>
fuzzy_method	Character. One of "d1" (Damerau-Levenshtein, default), "levenshtein", or "jw" (Jaro-Winkler).
verbose	Logical. Print progress messages. Default TRUE.

**Details**

When multiple backends are specified, names are matched against each backend in order. Names matched by an earlier backend are not re-matched by later ones (fallback chain).

**Value**

A data.frame with one row per input name and the following columns:

- input\_name** The original name as provided.
- matched\_name** Full name in the backbone that matched.
- accepted\_name** Resolved accepted name (equals matched\_name if not a synonym).
- taxon\_id** Backend-specific ID of the matched name.
- accepted\_id** ID of the accepted name.
- rank** Taxonomic rank (species, subspecies, genus, etc.).
- family** Family name.

- genus** Genus name.
- epithet** Specific epithet.
- authorship** Authorship of the matched name.
- accepted\_authorship** Authorship of the accepted name. For a synonym this is the author of the resolved accepted name, not the synonym's own author, so `accepted_name` and `accepted_authorship` together form the accepted name's full citation.
- is\_synonym** Logical. Was the match a synonym?
- is\_hybrid** Logical. Was a hybrid marker detected in the input?
- match\_type** One of "exact", "exact\_ci", "fuzzy", "abbrev" (an abbreviated genus such as "Q. robur" resolved via genus initial plus epithet), or "none".
- fuzzy\_dist** Normalized string distance (0–1), NA if exact.
- is\_ambiguous** Logical. TRUE when the matched scientificName had multiple synonym rows pointing to different accepted taxa at the same priority tier (homonym ambiguity). Disambiguated via `nomenclaturalStatus = "Valid"` when that column is in the backbone; for irreducible ambiguity, the scalar columns hold one candidate.
- ambiguous\_targets** Character. |-joined list of conflicting accepted taxon IDs when `is_ambiguous = TRUE`; NA otherwise.
- backend** Which backend was used (e.g., "wfo", "col", "gbif").
- backbone\_version** Backend name, version, and download date (e.g., "wfo:2024-12 (2026-04-01)"). Useful for reproducibility.

## Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

# Match a few names
taxify(c("Quercus robur", "Pinus sylvestris"))

# Disable fuzzy matching
taxify("Quercus robur", fuzzy = FALSE)

# Fallback chain: try WFO first, then COL for unmatched
taxify(c("Quercus robur", "Panthera leo"),
       backend = c("wfo", "col"))

options(old)
```

---

`taxify_clear_cache`      *Clear all cached backbones*

---

## Description

Removes all loaded backbone handles from memory. The next call to `taxify()` will re-load from disk.

**Usage**

```
taxify_clear_cache()
```

**Value**

No return value, called for side effects.

---

taxify_data_dir	<i>Get the taxify data directory</i>
-----------------	--------------------------------------

---

**Description**

Returns the directory where taxify stores downloaded backbone and enrichment .vtr files. By default this is the platform-appropriate per-user cache returned by `tools::R_user_dir()` (available since R 4.0).

**Usage**

```
taxify_data_dir()
```

**Details**

The location can be overridden, in order of precedence, by the `taxify.data_dir` option (`getOption("taxify.data_dir")`) or the `TAXIFY_DATA_DIR` environment variable. This is useful to point taxify at a shared cache, or at the small bundled example database returned by `taxify_example_data()`.

**Value**

Character string. Path to the data directory.

---

taxify_download	<i>Download a backbone database</i>
-----------------	-------------------------------------

---

**Description**

Downloads the latest Darwin Core snapshot for the specified backend and converts it to vectra's .vtr format for fast repeated queries.

**Usage**

```
taxify_download(backend, dest = NULL, verbose = TRUE, ...)
```

**Arguments**

backend	A taxify_backend object or a character string (e.g., "wfo").
dest	Character. Destination directory. Defaults to <code>taxify_data_dir()</code> .
verbose	Logical. Print progress messages.
...	Additional arguments passed to methods.

**Details**

Always re-downloads the latest release, overwriting any existing backbone. Use `taxify()` for day-to-day matching — it auto-downloads on first use and reuses the local copy thereafter.

**Value**

The path to the .vtr file (invisibly).

---

taxify\_download\_enrichment

*Download one or more enrichment .vtr files*

---

**Description**

Downloads pre-built enrichment .vtr files from the taxify manifest.

**Usage**

```
taxify_download_enrichment(enrichment, version = "latest", verbose = TRUE)
```

**Arguments**

enrichment	Character. One or more enrichment names (e.g., "conservation_status", "griis", "woodiness").
version	Character. "latest" (default) or a specific version string.
verbose	Logical. Default TRUE.

**Details**

Available enrichments:

**conservation\_status** IUCN conservation status (LC/NT/VU/EN/CR/EW/EX)

**griis** GRIIS invasive species status by country

**woodiness** Zanne et al. 2014 woody/herbaceous classification

**wcyp** WCVP native range by TDWG botanical region

**eive** EIVE 1.0 ecological indicator values (European plants)

**diaz\_traits** Diaz et al. 2022 seed mass and plant height

**elton\_traits** EltonTraits 1.0 diet and foraging (birds + mammals)  
**avonet** AVONET bird morphology and migration  
**pantheria** PanTHERIA mammal life-history traits  
**common\_names** GBIF vernacular names (multi-language)  
**amphibio** AmphiBIO amphibian life-history and ecological traits  
**leda** LEDA Traitbase NW European plant traits (Kleyer et al. 2008)

### Value

The path(s) to the downloaded .vtr file(s) (invisibly).

---

taxify\_download\_vtr *Download a taxify backbone*

---

### Description

Downloads a pre-built .vtr backbone from Zenodo using the taxify manifest. Progress is always shown. No prompts are shown — calling this function is consent.

### Usage

```
taxify_download_vtr(backend = "wfo", version = "latest", verbose = TRUE)
```

### Arguments

backend	Character. One of "wfo", "col", "gbif", or "register". Multiple backends can be specified as a character vector.
version	Character. "latest" (default) downloads into <data_dir>/<backend>/latest/ and will be overwritten on future updates. A specific version string (e.g., "2024.01") downloads into a pinned folder that is never overwritten.
verbose	Logical. Default TRUE.

### Value

The path(s) to the downloaded .vtr file(s) (invisibly).

---

taxify\_example\_data *Path to the bundled example database*

---

### Description

taxify ships a tiny example database (a handful of species per backbone plus matching enrichment tables) so that examples and quick experiments run offline, without downloading the full multi-million-row backbones.

### Usage

```
taxify_example_data()
```

### Details

Point taxify at it for the current session by setting the `taxify.data_dir` option:

```
old <- options(taxify.data_dir = taxify_example_data())
taxify("Quercus robur") |> add_woodiness()
options(old) # restore the real data directory
```

The example database is read-only and covers only the species used in the package examples; use the full downloaded backbones for real work.

### Value

Character string. Path to the bundled example database directory, or "" if it is not installed.

### See Also

[taxify\\_data\\_dir\(\)](#)

---

taxify\_load\_register *Load the unified genus register into memory*

---

### Description

Reads `genus_register.vtr` from disk and caches it as a `data.frame` in `.taxify_env$register`. Subsequent calls reuse the cached version unless `force = TRUE`.

### Usage

```
taxify_load_register(force = FALSE, verbose = TRUE)
```

**Arguments**

force                Logical. If TRUE, reloads from disk even if already cached. Default FALSE.  
 verbose             Logical. Print progress messages. Default TRUE.

**Details**

The register contains one row per genus with columns: genus, kingdom, phylum, class, order, family, life\_form.

**Value**

The register data.frame (invisibly).

---

taxify_long	<i>Reshape grouped enrichment columns to long format</i>
-------------	--

---

**Description**

Converts wide-format columns produced by grouped enrichments (e.g., `invasive_status_AT`, `invasive_status_DE`) back to long format with one row per species x group combination.

**Usage**

```
taxify_long(x, cols = NULL, group_col = NULL, drop_na = FALSE)
```

**Arguments**

x                    A data.frame, typically a `taxify()` result after applying a grouped enrichment like `add_invasive_status()`, `add_alien_first_records()`, or `add_wcvp()`.

cols                Character vector of base column names to reshape. These are the column names without the group suffix (e.g., "invasive\_status", not "invasive\_status\_AT"). If omitted, auto-detected from the enrichment metadata stamped by the `add_*`() functions.

group\_col          Character. Name for the output group column. If omitted, auto-detected from enrichment metadata (e.g., "country\_code" for invasive status or alien first records).

drop\_na            Logical. If TRUE, drop rows where all value columns are NA. Default FALSE.

**Details**

When `cols` and `group_col` are omitted, `taxify_long()` reads the reshape metadata attached by grouped enrichment functions (`add_invasive_status()`, `add_alien_first_records()`, `add_wcvp()`, `add_common_names()`). If multiple grouped enrichments were applied, all are reshaped together (they must share the same group column).

Column matching uses the explicit base names in `cols` to avoid ambiguity. For example, given `cols = c("alien_first_record", "alien_first_record_source")`, the column `alien_first_record_source_AT`

is correctly matched to base `alien_first_record_source` (not `alien_first_record` with suffix `source_AT`), because longer base names are matched first.

If the columns in `x` exactly match `cols` (no suffixed variants), the data is already in single-group format. In this case, the `data.frame` is returned unchanged with `group_col` set to `NA`.

### Value

A `data.frame` in long format. All columns from `x` that are not part of the reshape are preserved. The reshaped columns use their base names (without suffix), and a new `group_col` column contains the group code extracted from the suffix.

### Examples

```
# Runs offline against the bundled example database.
old <- options(taxify.data_dir = taxify_example_data())

# Auto-detected: no cols or group_col needed
taxify("Robinia pseudoacacia") |>
  add_alien_first_records(country = c("AT", "DE")) |>
  taxify_long()

# Explicit: override auto-detection
taxify("Robinia pseudoacacia") |>
  add_invasive_status(country = c("AT", "DE")) |>
  taxify_long(cols = "invasive_status", group_col = "country")

options(old)
```

---

taxify\_refresh\_manifest

*Invalidate the session manifest cache*

---

### Description

Forces the next `fetch_manifest()` call to re-fetch from the network. Useful after the maintainer updates the manifest between R sessions without restarting R.

### Usage

```
taxify_refresh_manifest()
```

### Value

No return value, called for side effects.

---

`taxify_register_coverage`*Show backend coverage for a genus*

---

**Description**

Queries `backend_coverage.vtr` to determine which backends contain the given genus, along with the backbone version at time of indexing.

**Usage**

```
taxify_register_coverage(genus)
```

**Arguments**

`genus`                      Character scalar. The genus name to query.

**Value**

A `data.frame` with columns `genus`, `backend`, `version`, `date_added`. Returns a zero-row `data.frame` if the genus is not found in any backend.

# Index

add\_algae\_traits, 3  
add\_alien\_first\_records, 4  
add\_alien\_first\_records(), 51  
add\_amphibio, 5  
add\_anage, 6  
add\_animaltraits, 7  
add\_arthropod\_traits, 8  
add\_avonet, 9  
add\_baseflor, 10  
add\_baseflor(), 18, 24  
add\_col\_info, 12  
add\_common\_names, 13  
add\_common\_names(), 51  
add\_conservation\_status, 14  
add\_data, 15  
add\_data(), 42  
add\_diaz\_traits, 17  
add\_ecoflora, 18  
add\_ecoflora(), 24  
add\_eive, 19  
add\_eive(), 11, 18, 24, 37  
add\_elton\_traits, 20  
add\_fish\_traits, 23  
add\_fishbase, 22  
add\_floraweb, 24  
add\_floraweb(), 18  
add\_fungal\_traits, 25  
add\_funguild, 27  
add\_gbif\_info, 28  
add\_glonaf, 29  
add\_hybrid\_info, 30  
add\_invasive\_status, 31  
add\_invasive\_status(), 51  
add\_leda, 32  
add\_leda(), 11  
add\_leptrait, 33  
add\_lizard\_traits, 34  
add\_pantheria, 35  
add\_pignatti, 36  
add\_qualifier\_info, 37  
add\_wcvp, 38  
add\_wcvp(), 51  
add\_wfo\_info, 39  
add\_woodiness, 40  
cite, 41  
export\_data, 42  
list\_enrichments, 42  
lookup\_genus, 43  
print.taxify\_result, 43  
summary.taxify\_result, 44  
taxify, 44  
taxify(), 3–15, 17–40, 42, 46, 48, 51  
taxify\_clear\_cache, 46  
taxify\_data\_dir, 47  
taxify\_data\_dir(), 48, 50  
taxify\_download, 47  
taxify\_download\_enrichment, 48  
taxify\_download\_vtr, 49  
taxify\_example\_data, 50  
taxify\_example\_data(), 47  
taxify\_load\_register, 50  
taxify\_long, 51  
taxify\_long(), 16  
taxify\_refresh\_manifest, 52  
taxify\_register\_coverage, 53  
tools::R\_user\_dir(), 47