Package: taxify (via r-universe)

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`>

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

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

RemoteUrl https://github.com/cran/taxify

RemoteRef HEAD

RemoteSha 327dd964ce28011407e43dd3b5acfadac963d326

Contents

`add_algae_traits` *Add macroalgal functional traits (AlgaeTraits)*

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

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

`x`	A data.frame returned by `taxify()`.
`country`	Character. ISO 3166-1 alpha-2 country code(s), or `"all"`. • Single code (e.g., `"AT"`): adds columns without suffix. • Multiple codes (e.g., `c("AT", "DE")`): adds columns with country suffix (e.g., `alien_first_record_AT`). • `"all"`: adds one column set per country in the dataset.
`verbose`	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` *Add amphibian life-history traits (AmphiBIO)*

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

`x`	A data.frame returned by `taxify()`.
`verbose`	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` *Add cross-taxon body mass and metabolic rate (AnimalTraits)*

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 `taxify()`.
`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` *Add bird morphology and migration (AVONET)*

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 `taxify()`.
`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, gynomonoecious, 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` *Add COL-specific columns*

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` *Add common (vernacular) names*

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

`x`	A data.frame returned by `taxify()`.
`lang`	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"`.
`verbose`	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

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

`x`	A data.frame returned by `taxify()`.
`data`	One of: • A **data.frame** already in R. • A **file path** to a `.csv`, `.csv.gz`, `.tsv`, `.tsv.gz`, `.xlsx`, `.sqlite`/`.db`, or `.vtr` file (read via vectra).
`species_col`	Character. Name of the column in `data` that contains species names. If `NULL` (default), auto-detected by matching `head(10)` 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 `taxify_long()` 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` *Add seed mass and plant height (Diaz et al. 2022)*

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 `taxify()`.
`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

`x`	A data.frame returned by `taxify()`.
`verbose`	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_fish_traits` *Add freshwater fish morphological traits (FISHMORPH)*

Description

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

Usage

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

Arguments

`x`	A data.frame returned by `taxify()`.
`verbose`	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, Villeg ér S (2021) FISHMORPH: 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_fishbase` *Add fish traits (FishBase)*

Description

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

Usage

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

Arguments

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

`x`	A data.frame returned by `taxify()`.
`verbose`	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_organs_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 `taxify()`.
`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` *Add fungal functional guild data (FUNGuild)*

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

`x`	A data.frame returned by `taxify()`.
`verbose`	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` *Add GBIF-specific columns*

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 Infraspecific 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` *Add naturalized alien flora status (GloNAF)*

Description

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

Usage

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

Arguments

`x`	A data.frame returned by `taxify()`.
`region`	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). • Single region: adds `naturalized` column (no suffix). • Multiple regions: adds `naturalized_<region>` columns. • `"all"`: adds one column per region in the dataset.
`verbose`	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` *Add hybrid parent and type information*

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

`x`	A data.frame returned by `taxify()`.
`country`	Character. ISO 3166-1 alpha-2 country code(s), or `"all"`. • Single code (e.g., `"AT"`): adds `invasive_status` column (no suffix). • Multiple codes (e.g., `c("AT", "DE")`): adds `invasive_status_AT`, `invasive_status_DE`. • `"all"`: adds one column per country in the dataset.
`verbose`	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

`x`	A data.frame returned by `taxify()`.
`verbose`	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 mm^2/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` *Add butterfly traits (LepTraits)*

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

`x`	A data.frame returned by `taxify()`.
`verbose`	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 `taxify()`.
`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` *Add mammal life-history traits (PanTHERIA)*

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 `taxify()`.
`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^2.
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` *Add WCVP native range status*

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` *Add WFO-specific columns*

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` *Add woodiness classification*

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 `taxify()`.
`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` *Export a taxify result to file*

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 `taxify()`.
`path`	Character. Output file path. The format is inferred from the extension: `.vtr`, `.csv`, `.tsv`, or `.xlsx`.
`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` *List available enrichments*

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` *Look up a genus in the register*

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` *Print a taxify_result*

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("dl", "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: • **Fractional** (`0 < fuzzy_threshold < 1`): normalized distance (edits / max name length). Default `0.2` is about 1 edit per 5 characters. • **Integer** (`fuzzy_threshold >= 1`): 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"`.
`fuzzy_method`	Character. One of `"dl"` (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 robus", 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` *Get the taxify data directory*

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` *Download a backbone database*

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 `taxify_data_dir()`.
`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
wcvp	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` *Reshape grouped enrichment columns to long format*

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