

Package ‘taxify’

July 9, 2026

Title Offline Taxonomic Name Matching Against Local Darwin Core
Snapshots

Version 0.3.4

Description Match taxonomic names against locally stored Darwin Core backbone
databases ('WFO', 'COL', 'GBIF', 'ITIS', 'NCBI Taxonomy', 'Open Tree of Life',
'WoRMS', 'Euro+Med', 'Species Fungorum', 'AlgaeBase', 'FishBase',
'SeaLifeBase', 'Reptile Database'). 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://gillescolling.com/taxify/>,
<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, GIFT, knitr, openxlsx2, rmarkdown, RSQLite, sf,
taxifydb, terra, testthat (>= 3.0.0), TR8, withr

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 CRAN

Date/Publication 2026-07-09 08:50:02 UTC

Contents

add_algae_traits	4
add_alien_first_records	5
add_amniote	6
add_amphibio	8
add_anage	9
add_animaltraits	10
add_arctic_traits	11
add_arthropod_traits	12
add_austraits	13
add_avonet	15
add_bacdiv	16
add_baseflor	17
add_bee_ostwald	18
add_bet	19
add_beukhof	21
add_bien	22
add_birdbase	23
add_blanchard	24
add_brot	25
add_chelonians	26
add_classification	27
add_col_info	28
add_combine	29
add_common_names	30
add_coral_traits	31
add_data	32
add_diaz_traits	35
add_disperse	36
add_ecoflora	37
add_edwards_phyto	38
add_eive	39
add_elton_traits	40
add_eupolltrait	42
add_eurobat	43
add_fishbase	44
add_fishmorph	45
add_floraweb	46
add_freshwater_insects_conus	48
add_frugivoria	49
add_fungalroot	50
add_fungal_traits	51
add_funguild	52
add_gbif_info	54
add_gift	55
add_globi	56
add_globtherm	57

add_glonaf	58
add_griis	59
add_groot	60
add_gwdd	62
add_homerange	63
add_hosts	64
add_huang_amph	65
add_hybrid_info	66
add_italic	66
add_iucn	68
add_kew_sid	69
add_leda	70
add_leptraits	71
add_madin	72
add_nesttrait	73
add_nztd	74
add_octocoral	75
add_odonata	76
add_pantheria	77
add_parravicini	78
add_pelagic	79
add_phylacine	80
add_pignatti	81
add_pottier	82
add_quimbayo	83
add_ramond	84
add_repttraits	85
add_rimet_phyto	86
add_saproxylic	87
add_sealifebase	88
add_sharkipedia	89
add_sheld	90
add_spider_traits	91
add_tetradensity	92
add_thermofresh	93
add_trait	94
add_tree_of_sex	96
add_usda_fungus_host	98
add_useful_plants	99
add_wcvp	100
add_wfo_info	101
add_zanne	102
add_zooplankton	103
children	104
cite	105
enrichment_cols	105
export_data	106
gift_traits	107

inspect	107
list_enrichments	110
list_traits	110
lookup_genus	111
print.taxify_inspection	112
print.taxify_result	112
summary.taxify_result	113
synonyms	113
taxify	114
taxify_candidates	118
taxify_clear_cache	119
taxify_data_dir	119
taxify_download	120
taxify_download_enrichment	120
taxify_download_vtr	121
taxify_example_data	122
taxify_load_register	123
taxify_long	123
taxify_refresh_manifest	125
taxify_regions	125
taxify_register_coverage	126
trait_info	126

Index	128
--------------	------------

add_algae_traits	<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, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
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, cols = NULL, 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". • Single code (e.g., "AT"): adds columns without suffix. • Multiple codes (e.g., c("AT", "DE")): adds columns with country suffix (e.g., <code>alien_first_record_AT</code>). • "all": adds one column set per country in the dataset.
<code>cols</code>	Character vector of value columns to attach (from <code>alien_first_record</code> , <code>alien_first_record_source</code> , <code>alien_first_record_reference</code>), or "all". NULL (default) attaches all three.
<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_amniote

Add amniote life-history traits (Amniote Life History Database)

Description

Joins uniform life-history traits for birds, mammals and reptiles to a `taxify()` result by looking up `accepted_name`.

Usage

```
add_amniote(x, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
<code>verbose</code>	Logical. Default TRUE.

Details

Source: Amniote Life History Database (Myhrvold et al. 2015, Ecology, CC0). Coverage: 21,322 species across birds, mammals and reptiles.

Value

The same data.frame with additional columns:

amniote_class Taxonomic class (Aves/Mammalia/Reptilia).
amniote_adult_body_mass_g Adult body mass (g).
amniote_no_sex_body_mass_g Unsexed adult body mass (g).
amniote_female_body_mass_g Female body mass (g).
amniote_male_body_mass_g Male body mass (g).
amniote_adult_svl_cm Adult snout-vent length (cm).
amniote_maximum_longevity_y Maximum longevity (years).
amniote_litter_clutch_size Litter or clutch size (count).
amniote_clutches_per_y Litters or clutches per year (count).
amniote_egg_mass_g Egg mass (g).
amniote_incubation_d Incubation period (days).
amniote_female_maturity_d Age at female maturity (days).
amniote_gestation_d Gestation length (days).
amniote_weaning_d Weaning age (days).
amniote_birth_hatching_wt_g Birth or hatching weight (g).

References

Myhrvold NP et al. (2015) An amniote life-history database to perform comparative analyses with birds, mammals, and reptiles. Ecology 96:3109. doi:[10.1890/150846R.1](https://doi.org/10.1890/150846R.1)

Examples

```
taxify("Accipiter badius", backend = "gbif") |>
  add_amniote()
```

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, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</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_y Age at maturity in years.

longevity_yr Maximum longevity in years.

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	<i>Add longevity and life-history traits (AnAge)</i>
-----------	--

Description

Joins AnAge (Animal Ageing and Longevity Database) traits to a [taxify\(\)](#) result by looking up accepted_name.

Usage

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

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
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, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
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_arctic_traits	<i>Add Arctic marine benthos traits</i>
-------------------	---

Description

Joins Arctic Traits Database functional traits to a [taxify\(\)](#) result by accepted_name. Each fuzzy-coded trait is reduced to its dominant category.

Usage

```
add_arctic_traits(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: Arctic Traits Database (Degen & Faulwetter 2019, University of Vienna PHAIDRA, CC-BY 4.0).

Value

The same data.frame with categorical arctic_traits_ columns: feeding_habit, skeleton, reproduction, larval_development, size, living_habit, body_form, mobility, bioturbation, depth_range, trophic_level, fragility, sociability, longevity.

References

Degen R, Faulwetter S (2019) The Arctic Traits Database. University of Vienna. doi:[10.25365/phaidra.49](https://doi.org/10.25365/phaidra.49)

Examples

```
taxify("Astarte borealis", backend = "gbif") |>
  add_arctic_traits()
```

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, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by <code>taxify()</code> .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</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_austraits	<i>Add Australian plant traits (AusTraits)</i>
---------------	--

Description

Joins species-level plant functional traits to a `taxify()` result by looking up `accepted_name`. Values are aggregated from the long-format AusTraits database (numeric traits by median, categorical traits by mode).

Usage

```
add_austraits(x, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
<code>verbose</code>	Logical. Default TRUE.

Details

Source: AusTraits (Falster et al. 2021, Scientific Data, CC BY 4.0). Coverage: ~33k Australian plant taxa.

Value

The same data.frame with additional columns:

austraits_plant_growth_form Plant growth form.
austraits_life_history Life history (annual/perennial/...).
austraits_woodiness Woodiness.
austraits_photosynthetic_pathway Photosynthetic pathway (C3/C4/CAM).
austraits_dispersal_syndrome Dispersal syndrome.
austraits_resprouting_capacity Resprouting capacity (fire response).
austraits_flowering_time Flowering time.
austraits_plant_height_m Plant height (m).
austraits_leaf_length_mm Leaf length (mm).
austraits_leaf_width_mm Leaf width (mm).
austraits_leaf_area_mm2 Leaf area (mm2).
austraits_leaf_mass_per_area Leaf mass per area (g/m2; SLA is its reciprocal).
austraits_leaf_n_per_dry_mass Leaf nitrogen per dry mass (mg/g).
austraits_leaf_p_per_dry_mass Leaf phosphorus per dry mass (mg/g).
austraits_seed_dry_mass_mg Seed dry mass (mg).
austraits_wood_density_g_cm3 Wood density (g/cm3).

References

Falster D et al. (2021) AusTraits, a curated plant trait database for the Australian flora. Scientific Data 8:254. doi:[10.1038/s41597021010066](https://doi.org/10.1038/s41597021010066)

Examples

```
taxify("Eucalyptus globulus", backend = "gbif") |>
  add_austraits()
```

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, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach. NULL (the default) attaches the curated set; a character vector of column names attaches just those, and "all" attaches every column the source carries (see enrichment_cols()).
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_bacdiv

Add bacterial and archaeal strain phenotypes (BacDive)

Description

Joins per-species microbial phenotype and growth-condition traits from BacDive, the Bacterial Diversity Metadatabase (DSMZ), to a `taxify()` result by looking up `accepted_name`. Strain-level records are aggregated to one row per species (categorical traits by mode, numeric by median); temperature and pH prefer the optimum measurement, falling back to the growth measurement.

Usage

```
add_bacdiv(x, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
<code>verbose</code>	Logical. Default TRUE.

Details

Source: BacDive (Reimer et al.), DSMZ, CC BY 4.0. ~18.6k bacterial and archaeal species with at least one phenotypic trait.

Value

The same data.frame with additional columns:

gram_stain Gram reaction (positive / negative / variable).

cell_shape Cell morphology (rod, coccus, ...).

motility motile / non-motile.

oxygen_metabolism Oxygen tolerance (aerobe, anaerobe, facultative anaerobe, microaerophile, ...).

cell_length_um, cell_width_um Cell dimensions in micrometres.

optimal_growth_temp_c Optimal (or reported growth) temperature, C.

optimal_growth_ph Optimal (or reported growth) pH.

References

Reimer LC et al. (2022) BacDive in 2022: the knowledge base for standardized bacterial and archaeal data. *Nucleic Acids Research* 50:D741-D746. doi:10.1093/nar/gkab961

Examples

```
taxify("Escherichia coli", backend = "gbif") |>
  add_bacdive()
```

add_baseflor	<i>Add plant traits from Baseflor (Catminat / Julve)</i>
--------------	--

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, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
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, écologique 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_bee_ostwald	<i>Add bee morphometrics (Ostwald)</i>
-----------------	--

Description

Joins global bee morphological traits to a `taxify()` result by `accepted_name`. Long-format measurements are reduced to species medians.

Usage

```
add_bee_ostwald(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by <code>taxify()</code> .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
verbose	Logical. Default TRUE.

Details

Source: Ostwald et al. global bee morphology (Zenodo, CC-BY 4.0).

Value

The same data.frame with numeric columns `bee_ostwald_itd_mm` (intertegular distance), `bee_ostwald_forewing_length_mm`, `bee_ostwald_tongue_length_mm`, `bee_ostwald_tongue_width_mm`, `bee_ostwald_body_length_mm`, `bee_ostwald_thorax_length_mm`, `bee_ostwald_hair_length_mm`, `bee_ostwald_hair_coverage_pct`.

References

Ostwald MM et al. (2024) A global database of bee morphological traits. Zenodo. [doi:10.5281/zenodo.13366989](https://doi.org/10.5281/zenodo.13366989)

Examples

```
taxify("Apis mellifera", backend = "gbif") |>
  add_bee_ostwald()
```

add_bet	<i>Add bryophyte traits (Bryophytes of Europe Traits)</i>
---------	---

Description

Joins species-level bryophyte traits to a `taxify()` result by looking up `accepted_name`.

Usage

```
add_bet(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by <code>taxify()</code> .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
verbose	Logical. Default TRUE.

Details

Source: Bryophytes of Europe Traits (van Zuijlen et al. 2023, EnviDat, CC BY-SA 4.0). Coverage: ~1.8k bryophyte species.

Value

The same data.frame with additional columns:

bet_growth_form Growth form (acrocarpous/pleurocarpous/thalloid/...).

bet_life_form Life form (cushion/mat/turf/weft/...).

bet_life_strategy Life strategy (During).

bet_sexual_condition Sexual condition (monoicous/dioicous).

bet_shoot_size_mm Mean shoot size (mm).

bet_generation_length_y Generation length (years).

bet_spore_diameter_um Mean spore diameter (micrometres).

bet_ind_light Ellenberg light indicator value.

bet_ind_temperature Ellenberg temperature indicator value.

bet_ind_moisture Ellenberg moisture indicator value.

bet_ind_reaction_ph Ellenberg reaction (pH) indicator value.

bet_ind_nitrogen Ellenberg nitrogen indicator value.

bet_substrate_soil Occurs on soil (0/1).

bet_substrate_rock Occurs on rock (0/1).

bet_substrate_bark Occurs on bark (0/1).

bet_substrate_deadwood Occurs on deadwood (0/1).

bet_epiphyte Epiphytic (0/1).

bet_redlist_category IUCN European Red List category.

References

van Zuijlen K et al. (2023) Bryophytes of Europe Traits (BET): a fundamental dataset for European bryophyte ecology. EnviDat. [doi:10.16904/envi.dat.348](https://doi.org/10.16904/envi.dat.348)

Examples

```
taxify("Abietinella abietina", backend = "gbif") |>
  add_bet()
```

add_beukhof	<i>Add marine fish traits (Beukhof)</i>
-------------	---

Description

Joins North Atlantic / NE Pacific shelf marine-fish life-history and ecology traits to a [taxify\(\)](#) result by accepted_name (species-level summaries).

Usage

```
add_beukhof(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: Beukhof et al. (2019) marine fish trait collection (PANGAEA, CC-BY 4.0).

Value

The same data.frame with beukhof_ columns: numeric trophic_level, aspect_ratio, offspring_size, age_maturity, fecundity, length_infinity_cm, growth_coefficient, length_max_cm; categorical habitat, feeding_mode, body_shape, fin_shape, spawning_type.

References

Beukhof E et al. (2019) A trait collection of marine fish species from North Atlantic and Northeast Pacific continental shelf seas. PANGAEA. doi:[10.1594/PANGAEA.900866](https://doi.org/10.1594/PANGAEA.900866)

Examples

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

add_bien	<i>Add plant traits (BIEN)</i>
----------	--------------------------------

Description

Joins BIEN plant functional traits to a [taxify\(\)](#) result by looking up accepted_name. Values are species-level aggregates of public-access BIEN records (numeric by median, categorical by mode).

Usage

```
add_bien(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: BIEN (Botanical Information and Ecology Network; Maitner et al. 2018, Methods Ecol Evol, CC BY). Coverage: tens of thousands of vascular plant species.

Value

The same data.frame with additional columns:

bien_plant_height_m Whole-plant height (m).

bien_max_plant_height_m Maximum whole-plant height (m).

bien_dbh_cm Diameter at breast height (cm).

bien_sla_mm2_mg Leaf area per leaf dry mass (SLA).

bien_leaf_area_mm2 Leaf area.

bien_leaf_dry_mass_mg Leaf dry mass.

bien_leaf_n_per_dry_mass Leaf nitrogen per dry mass.

bien_leaf_p_per_dry_mass Leaf phosphorus per dry mass.

bien_leaf_thickness_mm Leaf thickness.

bien_seed_mass_mg Seed mass.

bien_wood_density_g_cm3 Stem wood density (g/cm3).

bien_leaf_lifespan Leaf life span.

bien_growth_form Whole-plant growth form.

bien_woodiness Whole-plant woodiness.

bien_dispersal_syndrome Whole-plant dispersal syndrome.

bien_flower_color Flower colour.

References

Maitner BS et al. (2018) The BIEN R package: A tool to access the Botanical Information and Ecology Network (BIEN) database. *Methods in Ecology and Evolution* 9:373-379. doi:10.1111/2041210X.12861

Examples

```
taxify("Quercus alba", backend = "gbif") |>
  add_bien()
```

add_birdbase	<i>Add bird traits (BIRDBASE)</i>
--------------	-----------------------------------

Description

Joins BIRDBASE biogeography, conservation and life-history traits to a `taxify()` result by looking up accepted_name. Traits redundant with `add_avonet()` morphology are not carried.

Usage

```
add_birdbase(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by <code>taxify()</code> .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
verbose	Logical. Default TRUE.

Details

Source: BIRDBASE (Sekercioglu et al. 2025, figshare, CC BY 4.0). Coverage: ~11.6k bird species.

Value

- The same data.frame with additional columns:
- birdbase_iucn_status** IUCN Red List category.
 - birdbase_realm** Biogeographic realm.
 - birdbase_latitudinal_zone** Latitudinal zone (1 tropical to 5).
 - birdbase_island_endemic** Island-restricted breeding (0/1).
 - birdbase_restricted_range** Restricted-range species (0/1).
 - birdbase_elevation_min_m** Lower elevation limit (m).

- birdbase_elevation_max_m** Upper elevation limit (m).
- birdbase_elevation_range_m** Elevational breadth (m).
- birdbase_primary_habitat** Primary habitat.
- birdbase_habitat_breadth** Habitat breadth (number of habitats).
- birdbase_primary_diet** Primary diet.
- birdbase_diet_breadth** Diet breadth (number of food types).
- birdbase_specialization_esi** Ecological specialization index.
- birdbase_clutch_min** Minimum clutch size (eggs).
- birdbase_clutch_max** Maximum clutch size (eggs).
- birdbase_nest_type** Nest architecture.
- birdbase_flightlessness** Volancy (yes/no/partial).

References

Sekercioglu CH et al. (2025) BIRDBASE: a global database of bird ecological and life-history traits. Scientific Data. doi:10.1038/s41597025056153

Examples

```
taxify("Struthio camelus", backend = "gbif") |>
  add_birdbase()
```

add_blanchard	Add ant genus defensive traits (Blanchard & Moreau)
---------------	---

Description

Joins genus-level ant defensive and ecological traits to a `taxify()` result by genus.

Usage

```
add_blanchard(x, cols = NULL, verbose = TRUE)
```

Arguments

- x** A data.frame returned by `taxify()`.
- cols** Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See `enrichment_cols`.
- verbose** Logical. Default TRUE.

Details

Source: Blanchard & Moreau (2017) ant defensive traits (Dryad, CC0). Joins on genus because the database is genus-resolved.

Value

The same data.frame with blanchard_ columns: categorical subfamily, spines, sting, diet, nesting, foraging; numeric colony_size_workers (joined on genus).

References

Blanchard BD, Moreau CS (2017) Defensive traits in the ant genera database. Dryad. [doi:10.5061/dryad.st6sc](https://doi.org/10.5061/dryad.st6sc)

Examples

```
taxify("Camponotus pennsylvanicus", backend = "gbif") |>
  add_blanhard()
```

add_brot	<i>Add Mediterranean plant traits (BROT 2.0)</i>
----------	--

Description

Joins Mediterranean-Basin plant fire-response, regeneration and functional traits to a [taxify\(\)](#) result by accepted_name.

Usage

```
add_brot(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: BROT 2.0 (Tavsanoglu & Pausas 2018, Scientific Data, CC-BY 4.0).

Value

The same data.frame with brot_ columns: numeric seed_mass_mg, sla_mm2_mg, height_m, leaf_area_mm2; categorical resp_fire, growth_form, disp_mode, fruit_type, soil_seed_bank, seedling_emergence.

References

Tavsanoglu C, Pausas JG (2018) A functional trait database for Mediterranean Basin plants (BROT 2.0). Scientific Data 5:180135. [doi:10.6084/m9.figshare.c.3843841](https://doi.org/10.6084/m9.figshare.c.3843841)

Examples

```
taxify("Quercus coccifera", backend = "gbif") |>
  add_brot()
```

add_chelonians	<i>Add turtle traits (CheloniansTraits)</i>
----------------	---

Description

Joins species-level turtle and tortoise traits to a `taxify()` result by looking up `accepted_name`.

Usage

```
add_chelonians(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by <code>taxify()</code> .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
verbose	Logical. Default TRUE.

Details

Source: CheloniansTraits (Wang et al. 2025, figshare, CC BY 4.0). Coverage: 358 turtle and tortoise species. Numeric values reported as "min-max" ranges in the source are reduced to their midpoint.

Value

The same data.frame with additional columns:

- chelonian_carapace_length_mm** Maximum straight-line carapace length (mm).
- chelonian_max_mass_g** Maximum body mass (g).
- chelonian_clutch_size_mean** Mean clutch size (count).
- chelonian_clutch_size_max** Maximum clutch size (count).
- chelonian_clutches_per_year** Clutches per year (count).
- chelonian_incubation_d** Incubation period (days).
- chelonian_age_maturity_y** Age at sexual maturity (years).
- chelonian_max_lifespan_y** Maximum lifespan (years).
- chelonian_range_size_km2** Range size (km2).
- chelonian_diet** Diet (herbivorous/carnivorous/omnivorous).

chelonian_activity_time Activity time.

chelonian_microhabitat Microhabitat (aquatic/terrestrial/...).

chelonian_habitat_type Habitat type.

chelonian_shell_type Shell type (hardshell/softshell).

References

Wang Y et al. (2025) CheloniansTraits: a comprehensive trait database of global turtles and tortoises. figshare. doi:10.6084/m9.figshare.28828241

Examples

```
taxify("Chelonia mydas", backend = "gbif") |>
  add_chelonians()
```

add_classification	<i>Add the full higher classification to a taxify result</i>
--------------------	--

Description

Attaches the Linnaean ranks above family (kingdom, phylum, class, order) to a `taxify()` result by joining each matched row back to its backbone. The core `taxify()` output already carries family and genus; this fills the ranks above them, for whichever ranks the matched backbone stores. Rows matched by different backends are each joined against their own backbone.

Usage

```
add_classification(
  x,
  ranks = c("kingdom", "phylum", "class", "order"),
  verbose = TRUE
)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>ranks</code>	Character vector of ranks to attach. Default <code>c("kingdom", "phylum", "class", "order")</code> .
<code>verbose</code>	Logical. Default TRUE.

Value

`x` with the requested rank columns added. A rank a backbone does not store is left NA (WFO, for example, carries no ranks above family).

See Also

`add_col_info()`, `add_gbif_info()` for backend-specific extras.

Examples

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

taxify("Naja naja", backend = "reptiledb") |>
  add_classification()

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_combine	<i>Add mammal traits (COMBINE)</i>
-------------	------------------------------------

Description

Joins COMBINE mammal traits to a [taxify\(\)](#) result by looking up accepted_name. COMBINE is a separate, coalesced mammal trait source; it is offered alongside [add_pantheria\(\)](#), not as a replacement. Reported (not phylogenetically imputed) values are used.

Usage

```
add_combine(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: COMBINE (Soria et al. 2021, Ecology, CC0). Coverage: ~6.2k mammal species. Keyed on the IUCN 2020 binomial.

Value

The same data.frame with additional columns:

combine_adult_mass_g Adult body mass (g).
combine_adult_body_length_mm Adult head-body length (mm).
combine_litter_size_n Litter size (count).
combine_litters_per_year_n Litters per year (count).
combine_max_longevity_d Maximum longevity (days).
combine_gestation_length_d Gestation length (days).
combine_weaning_age_d Weaning age (days).

combine_generation_length_d Generation length (days).
combine_dispersal_km Natal dispersal distance (km).
combine_habitat_breadth_n Number of IUCN habitats (count).
combine_diet_breadth_n Number of diet categories (count).
combine_trophic_level Trophic level (1 herbivore, 2 omnivore, 3 carnivore).
combine_activity_cycle Activity cycle (1 nocturnal, 2 cathemeral, 3 diurnal).
combine_foraging_stratum Foraging stratum (G/Ar/A/S/M).
combine_biogeographical_realm Biogeographical realm(s).

References

Soria CD et al. (2021) COMBINE: a coalesced mammal database of intrinsic and extrinsic traits. Ecology 102:e03344. doi:10.1002/ecy.3344

Examples

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

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", cols = NULL, 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>cols</code>	Which columns to attach. NULL (the default) attaches the curated set; a character vector of column names attaches just those, and "all" attaches every column the source carries (see <code>enrichment_cols()</code>).
<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_coral_traits	<i>Add scleractinian coral traits (Coral Trait Database)</i>
------------------	--

Description

Joins species-level coral functional traits to a [taxify\(\)](#) result by looking up accepted_name. Values are aggregated from the long-format Coral Trait Database (numeric traits by median, categorical traits by mode).

Usage

```
add_coral_traits(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: Coral Trait Database (Madin et al. 2016, Scientific Data, CC BY 4.0). Coverage: ~1.5k coral species.

Value

The same data.frame with additional columns:

coral_symbiotic_state Zooxanthellate / azooxanthellate.

coral_growth_form Typical growth form (massive/branching/...).

coral_coloniality Colonial / solitary.

coral_substrate_attachment Attached / unattached.

coral_sexual_system Hermaphrodite / gonochore.

coral_larval_development_mode Spawner / brooder.

coral_symbiont_clade Symbiodinium clade.

coral_corallite_width_max_mm Maximum corallite width (mm).

coral_colony_max_diameter_cm Maximum colony diameter (cm).

coral_growth_rate_mm_yr Linear extension rate (mm/year).

coral_depth_lower_m Lower depth limit (m).

coral_depth_upper_m Upper depth limit (m).

coral_skeletal_density_g_cm3 Skeletal density (g/cm3).

References

Madin JS et al. (2016) The Coral Trait Database, a curated database of trait information for coral species from the global oceans. Scientific Data 3:160017. [doi:10.1038/sdata.2016.17](https://doi.org/10.1038/sdata.2016.17)

Examples

```
taxify("Acropora millepora", backend = "gbif") |>
  add_coral_traits()
```

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	<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, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
<code>verbose</code>	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_disperse	<i>Add aquatic-invertebrate dispersal traits (DISPERSE)</i>
--------------	---

Description

Joins genus-level dispersal-related traits for European aquatic macroinvertebrates to a [taxify\(\)](#) result by genus. Each fuzzy-coded trait is reduced to its dominant modality (with the database's own labels).

Usage

```
add_disperse(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: DISPERSE (Sarremejane et al. 2020, Scientific Data, CC-BY 4.0). Joins on genus because the database is genus-resolved.

Value

The same data.frame with categorical disperse_body_size_cm, disperse_life_cycle, disperse_repro_cycles, disperse_dispersal, disperse_adult_lifespan, disperse_female_wing_mm, disperse_wing_type, disperse_fecundity, disperse_drift, and the numeric bin-midpoint columns disperse_body_size_cm_mid, disperse_female_wing_mm_mid, disperse_fecundity_mid (all joined on genus).

References

Sarremejane R et al. (2020) DISPERSE, a trait database to assess the dispersal potential of European aquatic macroinvertebrates. Scientific Data 7:386. [doi:10.6084/m9.figshare.c.5000633](https://doi.org/10.6084/m9.figshare.c.5000633)

Examples

```
taxify("Baetis rhodani", backend = "gbif") |>
  add_disperse()
```

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, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by taxify() .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
<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_edwards_phyto	<i>Add phytoplankton nutrient-uptake traits (Edwards et al.)</i>
-------------------	--

Description

Joins species-level phytoplankton nutrient physiology (Droop/Monod uptake and growth parameters for ammonium, nitrate and phosphorus, plus cell size and carbon content) to a [taxify\(\)](#) result by looking up accepted_name.

Usage

```
add_edwards_phyto(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: Edwards et al. (2015, *Ecology*, CC BY 4.0), a compilation of phytoplankton nutrient-utilization traits for ~130 species. Single-source physiological data with no cross-source analogue, so it is surfaced through this door rather than the cross-source [add_trait\(\)](#) verb.

Value

The same data.frame with additional columns. The curated set:

edwards_taxon_group Coarse phytoplankton group (diatom, green, ...).

edwards_habitat_system Habitat (marine/freshwater).

edwards_cell_volume Cell volume (micron³).

edwards_carbon_per_cell Carbon content per cell (pg C).

edwards_mu_inf_nit Maximum growth rate on nitrate (per day).

edwards_k_nit Half-saturation constant for growth on nitrate.

edwards_qmin_nit Minimum cell nitrogen quota.

edwards_mu_inf_p Maximum growth rate on phosphorus (per day).

edwards_k_p Half-saturation constant for growth on phosphorus.

edwards_qmin_p Minimum cell phosphorus quota.

With `cols = "all"` the full set of ammonium/nitrate/phosphorus uptake and quota parameters (`vmax_amm`, `mu_nit`, `vmax_p`, `qmax_p`, ...) is attached under their source names. All uptake/quota traits are joined on `accepted_name`.

References

Edwards KF, Thomas MK, Klausmeier CA, Litchman E (2015) Phytoplankton growth and the interaction of light and temperature: A synthesis at the species and community level. *Ecology* 96(9):2554-2564. doi:[10.1890/142252.1](https://doi.org/10.1890/142252.1)

Examples

```
taxify("Thalassiosira pseudonana", backend = "gbif") |>
  add_edwards_phyto()
```

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, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
<code>verbose</code>	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)
```

<code>add_elton_traits</code>	<i>Add diet, foraging, and body mass (EltonTraits 1.0)</i>
-------------------------------	--

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, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
<code>verbose</code>	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).

diet_guild Dominant diet guild derived from the ten diet-fraction columns (fractions summed within guild, the guild reaching 50 percent wins, otherwise omnivore): carnivore, herbivore, omnivore, invertivore, frugivore, granivore, nectarivore, scavenger. Also reachable across sources via `add_trait(x, "diet_guild")`.

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_eupolltrait	<i>Add European pollinator traits (EuPollTrait)</i>
-----------------	---

Description

Joins European bee and hoverfly morphological, biogeographic and ecological traits to a [taxify\(\)](#) result by accepted_name.

Usage

```
add_eupolltrait(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: EuPollTrait (Milicic et al. 2025, Zenodo, CC-BY 4.0).

Value

The same data.frame with eupolltrait_ columns: numeric itd_mm, tongue_length_mm, species_temperature_index, species_continental_index, area_of_occupancy, extent_of_occurrence; categorical sociality, nest, larval_nutrition, body_length_category.

References

Milicic M et al. (2025) EuPollTrait: a trait database for European bees and hoverflies. Zenodo. [doi:10.5281/zenodo.18032357](https://doi.org/10.5281/zenodo.18032357)

Examples

```
taxify("Bombus terrestris", backend = "gbif") |>
  add_eupolltrait()
```

add_eurobat	Add European bat traits (<i>EuroBaTrait</i>)
-------------	--

Description

Joins species-level traits of European bats (morphology, life history, diet, foraging habitat, roost type) to a `taxify()` result by looking up `accepted_name`.

Usage

```
add_eurobat(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by <code>taxify()</code> .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
verbose	Logical. Default TRUE.

Details

Source: Froidevaux et al. (2023, Scientific Data, CC BY 4.0), EuroBaTrait 1.0. Thematic measurement-or-fact tables are reduced to species-level values (numeric by median, categorical by mode).

Value

The same data.frame with additional columns. The curated set:

- eurobat_forearm_length_mm** Forearm length (mm).
- eurobat_body_mass_g** Body mass (g).
- eurobat_max_longevity_yr** Maximum recorded longevity (years).
- eurobat_litter_size** Litter size.
- eurobat_diet_type** Diet type (insectivorous, frugivorous, ...).
- eurobat_first_main_preay** First main prey item.

With `cols = "all"` the full trait set (digit lengths, wing indices, habitat affinity scores, critical feeding areas, roost dependence, phenology, ...) is attached under their source names.

References

Froidevaux JSP et al. (2023) EuroBaTrait 1.0: a species-level trait dataset of bats in Europe and beyond. Scientific Data. figshare [doi:10.6084/m9.figshare.21777161](https://doi.org/10.6084/m9.figshare.21777161)

Examples

```
taxify("Myotis myotis", backend = "gbif") |>
  add_eurobat()
```

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, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: FishBase via rfishbase (Froese & Pauly, CC BY-NC 4.0). 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_fishmorph	<i>Add freshwater fish morphological traits (FISHMORPH)</i>
---------------	---

Description

Joins FISHMORPH morphological trait data to a [taxify\(\)](#) result by looking up accepted_name. This is the source-named door for FISHMORPH; for the fish reference database FishBase see [add_fishbase\(\)](#).

Usage

```
add_fishmorph(x, cols = NULL, verbose = TRUE)
```

Arguments

- x A data.frame returned by [taxify\(\)](#).
- cols Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See [enrichment_cols](#).
- 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) 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_fishmorph()

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, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which FloraWeb columns to attach. NULL (the default) attaches all of them; a character vector of <code>_de</code> column names attaches just those (see <code>enrichment_cols()</code> , or the columns listed below). Pass "all" for the full set explicitly.
<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 BioFlor, 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 / BioFlor).

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_freshwater_insects_conus
```

Add freshwater-insect genus traits (Freshwater Insects CONUS)

Description

Joins genus-level ecological and life-history trait modalities of North American freshwater insects to a [taxify\(\)](#) result by looking up genus, so any species in a covered genus is annotated. The modalities are the source's own abbreviation codes, kept verbatim.

Usage

```
add_freshwater_insects_conus(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: Twardochleb et al. (2021, Environmental Data Initiative, CC BY 4.0), the Freshwater Insects CONUS genus trait table. Traits are genus-level categorical modalities.

Value

The same data.frame with additional columns. The curated set:

fwinsect_thermal_pref Thermal preference class.

fwinsect_feed_prim Primary functional feeding group code.

fwinsect_habit_prim Primary habit (swimmer, clinger, burrower, ...).

fwinsect_rheophily Rheophily (current preference) code.

fwinsect_voltinism Voltinism (generations per year) code.

fwinsect_max_body_size Maximum body size class.

With cols = "all" the remaining trait groups (emergence, dispersal, respiration, ...) are attached under their source names.

References

Twardochleb LA et al. (2021) Freshwater insect occurrences and traits for the contiguous United States, 2001-2018. Environmental Data Initiative. doi:10.6073/pasta/8238ea9bc15840844b3a023b6b6ed158

Examples

```
taxify("Baetis", backend = "gbif") |>
  add_freshwater_insects_conus()
```

add_frugivoria	<i>Add Neotropical frugivore traits (Frugivoria)</i>
----------------	--

Description

Joins shared bird/mammal frugivore traits to a `taxify()` result by accepted_name.

Usage

```
add_frugivoria(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by <code>taxify()</code> .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
verbose	Logical. Default TRUE.

Details

Source: Gerstner et al. (2023) Frugivoria (EDI, CC-BY 4.0).

Value

The same data.frame with frugivoria_ columns: categorical taxon_group, diet_category; numeric diet_breadth, body_mass_g, body_size_mm, longevity, generation_time.

References

Gerstner BE et al. (2023) Frugivoria: a trait database for birds and mammals exhibiting frugivory across contiguous Neotropical moist forests. EDI (edi.1220.5).

Examples

```
taxify("Ramphastos toco", backend = "gbif") |>
  add_frugivoria()
```

add_fungalroot	<i>Add mycorrhizal type from FungalRoot</i>
----------------	---

Description

Joins genus-level mycorrhizal type from the FungalRoot database (Soudzilovskaia et al. 2020) to a [taxify\(\)](#) result by looking up genus. Mycorrhizal type is phylogenetically conserved at the genus level, which is the resolution FungalRoot recommends for inference, so this enrichment joins on genus rather than accepted_name.

Usage

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

Arguments

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

Details

Source: FungalRoot, published on GBIF as a Darwin Core Archive ([doi:10.15468/a7ujmj](https://doi.org/10.15468/a7ujmj)), CC BY-NC 4.0. The per-genus value is a majority consensus computed from the per-observation mycorrhiza type labels, not FungalRoot's own published per-genus assignment. Plant genera only. The .vtr is downloaded from the taxify release on first use and cached.

Value

The same data.frame with three additional columns:

mycorrhizal_type Genus-level majority-consensus type, one of AM (arbuscular), EcM (ecto), ErM (ericoid), OM (orchid), NM (non-mycorrhizal), the dual types EcM-AM / ErM-EcM / ErM-AM, Other, or uncertain. NA if the genus is not in FungalRoot.

mycorrhizal_status Coarse status derived from the type: "mycorrhizal", "non-mycorrhizal", or "uncertain".

mycorrhizal_records Number of FungalRoot observations supporting the genus-level consensus.

References

Soudzilovskaia NA et al. (2020) FungalRoot: global online database of plant mycorrhizal associations. *New Phytologist* 227:955-966.

Examples

```
# Joins on genus, so any species in a covered genus is annotated.
taxify(c("Quercus robur", "Trifolium pratense")) |>
  add_fungalroot()
```

add_fungal_traits	<i>Add fungal lifestyle and trait data (FungalTraits)</i>
-------------------	---

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, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
<code>verbose</code>	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, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</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 Infrspecific 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_gift

Add plant traits from GIFT

Description

Joins species-level plant traits from GIFT, the Global Inventory of Floras and Traits (Weigelt et al. 2020), to a `taxify()` result by `accepted_name`. GIFT aggregates published trait records to one value per species (mean for numeric traits, most frequent entry for categorical ones). You choose which traits to attach with `cols`; browse the available columns with `gift_traits()`.

Usage

```
add_gift(x, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which GIFT trait columns to attach. One of: <code>NULL</code> (the default) for a convenient set of well-populated traits; the string <code>"all"</code> for every bundled trait; or a character vector of <code>gift_</code> column names (e.g. <code>"plant_height_max"</code> , with or without the <code>gift_</code> prefix). See <code>gift_traits()</code> . When left <code>NULL</code> , a one-time message notes the default set and how to request all traits.
<code>verbose</code>	Logical. Default <code>TRUE</code> .

Details

The GIFT trait table is bundled as a pre-built `.vtr` and joined offline, so no internet access is needed once it is present (the first use downloads it, or builds it from source if `taxifydb` is installed). GIFT's API exposes only the redistributable subset of its data (CC BY 4.0; references whose underlying source is restricted are excluded), and that subset is what is bundled here. Cite GIFT and, where applicable, the underlying references (`GIFT::GIFT_references()`) when you use the values.

Value

The same data.frame with one added column per requested trait, named `gift_<trait>`. Numeric traits (heights, masses, areas) are doubles, the rest character. Rows with no value in GIFT get `NA`. With the default `cols`, the added columns are `gift_woodiness_1`, `gift_growth_form_1`, `gift_lifecycle_1`, `gift_life_form_1`, `gift_climber_1`, `gift_epiphyte_1`, `gift_parasite_1`, `gift_aquatic_1`, `gift_plant_height_max`, `gift_photosynthetic_pathway`, `gift_seed_mass_mean`, `gift_dispersal_syndrome_1`, `gift_flowering_start`, `gift_flowering_end`, `gift_deciduousness_1`, and `gift_sla_mean`.

References

Weigelt P, König C, Kreft H (2020) GIFT - A Global Inventory of Floras and Traits for macroecology and biogeography. *Journal of Biogeography* 47:16-43. doi:10.1111/jbi.13623 Denelle P, Weigelt P, Kreft H (2023) GIFT: an R package to access the Global Inventory of Floras and Traits. *Methods in Ecology and Evolution* 14:2738-2748. doi:10.1111/2041210X.14213

See Also

[gift_traits\(\)](#) to browse the available columns.

Examples

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

taxify("Abies alba") |>
  add_gift()

options(old)
```

add_globi	<i>Add biotic interaction degree (GloBI)</i>
-----------	--

Description

Joins per-species biotic interaction breadth from GloBI (Global Biotic Interactions) to a [taxify\(\)](#) result by looking up accepted_name. GloBI's aggregated interaction records are reduced to per-species counts: how many distinct partner taxa a species interacts with (undirected), across how many distinct interaction types, over how many interaction records.

Usage

```
add_globi(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: GloBI (Poelen et al. 2014), an open index of biotic interactions aggregated from many contributed datasets. Only derived per-species counts are distributed here; the underlying interaction records carry the licenses of their original data contributors, who should be cited in derivative work. Partner counts are resolved to accepted names before counting, so synonymous partners are not double-counted.

Value

The same data.frame with additional columns:

interaction_degree Number of distinct partner taxa recorded interacting with the species (both directions).

n_interaction_types Number of distinct interaction types (eats, pollinates, parasitises, ...).

n_interaction_records Total number of interaction records touching the species.

References

Poelen JH, Simons JD, Mungall CJ (2014) Global Biotic Interactions: An open infrastructure to share and analyze species-interaction datasets. *Ecological Informatics* 24:148-159. doi:10.1016/j.ecoinf.2014.08.005

Examples

```
taxify("Apis mellifera", backend = "gbif") |>
  add_globi()
```

add_globtherm	<i>Add thermal tolerance limits (GlobTherm)</i>
---------------	---

Description

Joins GlobTherm upper and lower thermal tolerance limits to a [taxify\(\)](#) result by looking up accepted_name.

Usage

```
add_globtherm(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: GlobTherm (Bennett et al. 2018, Scientific Data, CC0). Coverage: ~2.1k species across aquatic and terrestrial groups.

Value

The same data.frame with additional columns:

globtherm_thermal_max_c Upper thermal limit (degrees Celsius).

globtherm_thermal_max_metric Definition of the upper limit (e.g. ctmx, LT50, UTMZ); the value is ambiguous without it.

globtherm_thermal_min_c Lower thermal limit (degrees Celsius).

globtherm_thermal_min_metric Definition of the lower limit (e.g. ctmin, LT50, LTMZ).

globtherm_thermal_max_error Reported error on the upper limit.

globtherm_thermal_min_error Reported error on the lower limit.

References

Bennett JM et al. (2018) GlobTherm, a database on the thermal tolerance for aquatic and terrestrial organisms. Scientific Data 5:180022. doi:10.1038/sdata.2018.22

Examples

```
taxify("Lepomis gibbosus", backend = "gbif") |>
  add_globtherm()
```

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 <code>taxify()</code> .
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_griis	Add invasive species status (GRIIS)
-----------	-------------------------------------

Description

Joins GRIIS (Global Register of Introduced and Invasive Species) status to a `taxify()` result, filtered by country. This is the source-named door for GRIIS; GloNAF (`add_glonaf()`) carries related naturalized-alien status.

Usage

```
add_griis(x, country, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by <code>taxify()</code> .
country	Character. ISO 3166-1 alpha-2 country code(s), or "all". • Single code (e.g., "AT"): adds <code>invasive_status</code> column (no suffix). • Multiple codes (e.g., <code>c("AT", "DE")</code>): adds <code>invasive_status_AT</code>, <code>invasive_status_DE</code>. • "all": adds one column per country in the dataset.
cols	Which columns to attach. NULL (the default) attaches the curated set; a character vector of column names attaches just those, and "all" attaches every column the source carries (see <code>enrichment_cols()</code>).
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_griis(country = "AT")

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

options(old)
```

add_groot	<i>Add root traits (GRooT)</i>
-----------	--------------------------------

Description

Joins species-level root traits from the Global Root Traits (GRooT) database to a `taxify()` result by looking up `accepted_name`. GRooT aggregates root trait records to per-species means. The `.vtr` carries the full GRooT trait set (38 root traits); the default attaches the nine best-populated key traits, and `cols = "all"` attaches every one. Run `enrichment_cols("groot")` to list them.

Usage

```
add_groot(x, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the nine key traits below, "all" every GRooT trait, or a character vector of names. See <code>enrichment_cols</code> .
<code>verbose</code>	Logical. Default TRUE.

Details

Source: GRooT database (Guerrero-Ramirez et al. 2021). Vascular plants. GRooT data are publicly available and used here with the data-paper citation requested by the authors.

Value

The same data.frame with additional columns (per-species means). The default set:

root_diameter Mean root diameter.
specific_root_length Specific root length.
root_tissue_density Root tissue density.
root_n_concentration Root nitrogen concentration.
root_c_concentration Root carbon concentration.
root_mass_fraction Root mass fraction.
lateral_spread Lateral spread.
root_mycorrhizal_colonization Root mycorrhizal colonization intensity.
rooting_depth Maximum rooting depth.

`cols = "all"` additionally attaches root chemistry (P/K/Ca/Mg/Mn concentrations, C:N and N:P ratios), architecture (branching density and ratio, stele diameter and fraction, cortex thickness, vessel diameter and number), turnover (root lifespan, production, turnover rate, litter mass-loss rate), and specific root area, respiration, and dry-matter content, among others. Units follow the GRooT data paper; see the reference below.

References

Guerrero-Ramirez NR et al. (2021) Global root traits (GRooT) database. *Global Ecology and Biogeography* 30:25-37. doi:10.1111/geb.13179

Examples

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

taxify("Abies alba") |>
  add_groot()
```

options(old)

add_gwdd	<i>Add wood density (Global Wood Density Database v2)</i>
----------	---

Description

Joins species-level wood density to a `taxify()` result by looking up `accepted_name`. Wood density is reported as wood specific gravity (oven-dry mass / green volume), dimensionless and numerically equal to g/cm3.

Usage

```
add_gwdd(x, cols = NULL, verbose = TRUE)
```

Arguments

- `x` A data.frame returned by `taxify()`.
- `cols` Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See `enrichment_cols`.
- `verbose` Logical. Default TRUE.

Details

Source: Global Wood Density Database v2 (Fischer et al. 2026, New Phytologist, CC BY 4.0). Coverage: ~17.3k species. Bark density is not part of the aggregated source and is not included.

Value

The same data.frame with additional columns:

- gwdd_wood_density_g_cm3** Species-mean wood density (g/cm3).
- gwdd_wood_density_trunk_g_cm3** Trunk wood density (g/cm3).
- gwdd_wood_density_branch_g_cm3** Branch wood density (g/cm3).
- gwdd_n_measurements** Number of underlying measurements.

References

Fischer FJ et al. (2026) The Global Wood Density Database version 2. New Phytologist. doi:10.1111/nph.70860

Examples

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

add_homerange	<i>Add mammal home-range size (HomeRange)</i>
---------------	---

Description

Joins species-median home-range size and body mass to a [taxify\(\)](#) result by accepted_name.

Usage

```
add_homerange(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: Broekman et al. (2023) HomeRange database (Dryad, CC0). Per-individual records are reduced to species medians.

Value

The same data.frame with numeric homerange_home_range_km2 and homerange_body_mass_kg.

References

Broekman MJE et al. (2023) HomeRange: a global database of mammalian home ranges. Dryad. [doi:10.5061/dryad.d2547d85x](https://doi.org/10.5061/dryad.d2547d85x)

Examples

```
taxify("Panthera leo", backend = "gbif") |>  
  add_homerange()
```

add_hosts	<i>Add Lepidoptera hostplant breadth (NHM HOSTS)</i>
-----------	--

Description

Joins per-insect hostplant breadth from the Natural History Museum HOSTS database to a [taxify\(\)](#) result by looking up `accepted_name`. Each moth or butterfly species is summarised by how many distinct hostplants and hostplant families it has been recorded feeding on.

Usage

```
add_hosts(x, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by taxify() .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
<code>verbose</code>	Logical. Default TRUE.

Details

Source: Robinson et al. (2010) HOSTS, Natural History Museum, London (CC0). Lepidoptera only; ~24k species with at least one recorded hostplant.

Value

The same data.frame with additional columns:

host_plant_count Number of distinct hostplant species recorded.

host_family_count Number of distinct hostplant families recorded.

References

Robinson GS, Ackery PR, Kitching IJ, Beccaloni GW, Hernandez LM (2010) HOSTS - a Database of the World's Lepidopteran Hostplants. Natural History Museum, London. doi:10.5519/havt50xw

Examples

```
taxify("Papilio machaon", backend = "gbif") |>
  add_hosts()
```

add_huang_amph	<i>Add amphibian morphometrics (Huang)</i>
----------------	--

Description

Joins species-level amphibian body measurements to a `taxify()` result by `accepted_name`. Only measurements comparable across Anura, Caudata and Gymnophiona are carried; per-specimen values are reduced to species medians.

Usage

```
add_huang_amph(x, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
<code>verbose</code>	Logical. Default TRUE.

Details

Source: Huang amphibian morphological dataset (figshare, CC-BY 4.0).

Value

The same data.frame with numeric `huang_amph_svl_mm`, `huang_amph_head_length_mm`, `huang_amph_head_width_mm`, `huang_amph_eye_diameter_mm`, `huang_amph_forelimb_length_mm`, `huang_amph_hindlimb_length_mm` and categorical `huang_amph_taxon_order`.

References

Huang et al. A global amphibian morphological trait dataset. figshare. doi:10.6084/m9.figshare.21159229

Examples

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

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_italic	<i>Add Italian-lichen taxon-page traits (ITALIC)</i>
------------	--

Description

Joins per-species morphological and ecological descriptors from ITALIC, the Information System on Italian Lichens, to a `taxify()` result by looking up accepted_name. One row per species, scraped from the ITALIC 8.0 taxon pages.

Usage

```
add_italic(x, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) or "all" attaches every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
<code>verbose</code>	Logical. Default TRUE.

Details

Source: ITALIC 8.0 (Nimis; Univ. of Trieste), taxon-page descriptors, CC BY-SA 4.0. Lichens are otherwise almost absent from the bundled trait databases.

Value

The same data.frame with additional columns:

growth_form Thallus growth form (crustose, foliose, fruticose, ...).

substrata Substrata the species grows on.

photobiont Photosynthetic partner.

reproductive_strategy Reproductive strategy.

References

Nimis PL. ITALIC - The Information System on Italian Lichens, Version 8.0. University of Trieste, Dept. of Biology (<https://italic.units.it>), accessed 2026-07. System paper: Martellos S, Conti M, Nimis PL (2023) Aggregation of Italian Lichen Data in ITALIC 7.0. *Journal of Fungi* 9(5):556. doi:10.3390/jof9050556

Examples

```
taxify("Xanthoria parietina", backend = "gbif") |>  
  add_italic()
```

add_iucn

Add IUCN Red List conservation status

Description

Joins IUCN Red List conservation status to a [taxify\(\)](#) result by looking up accepted_name. This is the source-named door for the IUCN Red List; for conservation status reconciled across sources, use [add_trait\(\)](#) once more than one source is registered for it.

Usage

```
add_iucn(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
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_iucn()

options(old)
```

add_kew_sid	<i>Add seed traits from the Kew Seed Information Database (SER-SID)</i>
-------------	---

Description

Joins species-level seed traits from the Kew Seed Information Database (SER-SID) to a `taxify()` result by looking up `accepted_name`.

Usage

```
add_kew_sid(x, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
<code>verbose</code>	Logical. Default TRUE.

Details

Source: Royal Botanic Gardens Kew, Seed Information Database, served as SER-SID (<https://ser-sid.org/>), CC BY 2.0. Per-record measurements are reduced to per-species medians (numeric) and modes (categorical); a thousand-seed weight in grams equals the per-seed mass in milligrams.

Value

The same data.frame with additional columns. The curated set:

sid_thousand_seed_weight Thousand-seed weight (grams per 1000 seeds, median of all records).

sid_storage_behaviour Seed storage behaviour (Orthodox/Recalcitrant/Intermediate/Uncertain).

sid_oil_content_pct Seed oil content (percent, median).

sid_protein_content_pct Seed protein content (percent, median).

sid_lifeform Raunkiaer life-form code as recorded by SID.

With `cols = "all"` the seed-weight record count (`n_seed_weight_records`) and modal `fruit_type` are also attached under their source names. Joined on `accepted_name`.

References

Royal Botanic Gardens Kew. Seed Information Database (SID). <https://ser-sid.org/>

Examples

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

add_leda	<i>Add plant traits from LEDA Traitbase</i>
----------	---

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, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</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 mm²/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, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
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_madin	<i>Add bacterial and archaeal traits (Madin et al.)</i>
-----------	---

Description

Joins species-level bacterial and archaeal phenotypic and genome traits to a [taxify\(\)](#) result by looking up accepted_name.

Usage

```
add_madin(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: Madin et al. (2020, Scientific Data, CC BY 4.0). Coverage: ~14.9k bacterial and archaeal species.

Value

The same data.frame with additional columns:

madin_gram_stain Gram stain (positive/negative).
madin_metabolism Metabolism (aerobic/anaerobic/facultative/...).
madin_cell_shape Cell shape (bacillus/coccus/spiral/...).
madin_motility Motility (yes/no/flagella/gliding/...).
madin_sporulation Sporulation (yes/no).
madin_isolation_source Isolation source category.
madin_growth_temp_c Recorded growth temperature (degrees Celsius).
madin_optimum_temp_c Optimum growth temperature (degrees Celsius).
madin_optimum_ph Optimum growth pH.
madin_genome_size_bp Genome size (base pairs).
madin_gc_content_pct Genomic G+C content (percent).

References

Madin JS et al. (2020) A synthesis of bacterial and archaeal phenotypic trait data. Scientific Data 7:170. doi:10.1038/s4159702004974

Examples

```
taxify("Escherichia coli", backend = "gbif") |>
  add_madin()
```

add_nesttrait	<i>Add bird nest traits (NestTrait)</i>
---------------	---

Description

Joins bird nest-site, nest-structure and nest-attachment indicators to a `taxify()` result by accepted_name. Each trait is a 0/1 presence flag; a species may carry several flags within a group.

Usage

```
add_nesttrait(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by <code>taxify()</code> .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
verbose	Logical. Default TRUE.

Details

Source: NestTrait v2 (Chia et al. 2023, Scientific Data, CC-BY 4.0).

Value

The same data.frame with three collapsed categorical columns (nesttrait_structure, nesttrait_site, nesttrait_attachment, each a pipe-delimited set of modalities for multi-modal species) plus 20 raw 0/1 indicator columns prefixed nesttrait_: brood_parasite, mound_builder, seven nestsite_*, seven neststr_* and four nestatt_* flags.

References

Chia SY et al. (2023) A global database of bird nest traits. Scientific Data. doi:10.1038/s41597-023028371

Examples

```
taxify("Turdus merula", backend = "gbif") |>
  add_nesttrait()
```

add_nztd	Add NZ marine benthos traits (NZTD)
----------	-------------------------------------

Description

Joins New Zealand marine benthic-invertebrate functional traits to a `taxify()` result by `accepted_name`. Each fuzzy-coded trait is reduced to its dominant modality.

Usage

```
add_nztd(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by <code>taxify()</code> .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
verbose	Logical. Default TRUE.

Details

Source: NZTD (Lam-Gordillo et al. 2023, figshare, CC-BY 4.0).

Value

The same data.frame with categorical nztd_ columns: bioturbation, body_size, degree_of_attachment, feeding_mode, living_habit, mobility, morphology, movement_method, rigidity.

References

Lam-Gordillo O et al. (2023) New Zealand Trait Database (NZTD) for marine benthic invertebrates. figshare. doi:[10.6084/m9.figshare.21939647](https://doi.org/10.6084/m9.figshare.21939647)

Examples

```
taxify("Macomona liliana", backend = "gbif") |>
  add_nztd()
```

add_octocoral	<i>Add octocoral traits (Octocoral Trait Database)</i>
---------------	--

Description

Joins soft-coral (octocoral) colony, polyp, skeleton, symbiosis and feeding traits to a [taxify\(\)](#) result by accepted_name. Built from long-format records reduced to one value per species.

Usage

```
add_octocoral(x, cols = NULL, verbose = TRUE)
```

Arguments

- x A data.frame returned by [taxify\(\)](#).
- cols Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See [enrichment_cols](#).
- verbose Logical. Default TRUE.

Details

Source: Octocoral Trait Database v2.2 (Gomez-Gras et al., CC-BY 4.0).

Value

The same data.frame with octocoral_ columns: numeric colony_height, colony_width, tentacles_per_polyp; categorical growth_form, type_of_growth, type_of_skeleton, polyp_retractability, polyp_dimorphism, zooxanthellate, axis_presence, feeding_mechanism, coloniality, skeletal_rigidity, calcareous_sclerites_pr

References

Octocoral Trait Database v2.2. Zenodo. doi:[10.5281/zenodo.14228404](https://doi.org/10.5281/zenodo.14228404)

Examples

```
taxify("Gorgonia ventalina", backend = "gbif") |>
  add_octocoral()
```

add_odonata	<i>Add odonate behavioural/ecological traits (OPD)</i>
-------------	--

Description

Joins Odonate Phenotypic Database categorical traits to a `taxify()` result by `accepted_name` (modal value per species).

Usage

```
add_odonata(x, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
<code>verbose</code>	Logical. Default TRUE.

Details

Source: Odonate Phenotypic Database (Waller et al., Dryad, CC-BY 4.0).

Value

The same data.frame with categorical `odonata_territoriality`, `odonata_flight_mode`, `odonata_mate_guarding`, `odonata_habitat_openness`, `odonata_has_wing_pigment`.

References

Waller JT et al. The Odonate Phenotypic Database. Dryad. [doi:10.5061/dryad.15pm5qc](https://doi.org/10.5061/dryad.15pm5qc)

Examples

```
taxify("Calopteryx splendens", backend = "gbif") |>
  add_odonata()
```

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, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
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 km².

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_parravicini	<i>Add reef-fish trophic guild (Parravicini)</i>
-----------------	--

Description

Joins the consensus reef-fish trophic-guild assignment to a `taxify()` result by `accepted_name`. The guild is the modal expert classification.

Usage

```
add_parravicini(x, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
<code>verbose</code>	Logical. Default TRUE.

Details

Source: Parravicini et al. (2020) reef-fish trophic guilds (PLoS Biology, CC-BY 4.0).

Value

The same data.frame with categorical `parravicini_trophic_guild`.

References

Parravicini V et al. (2020) Delineating reef fish trophic guilds with global gut content data synthesis and phylogeny. PLoS Biology 18:e3000702. doi:10.1371/journal.pbio.3000702

Examples

```
taxify("Zebrasoma scopas", backend = "gbif") |>
  add_parravicini()
```

add_pelagic	<i>Add pelagic species traits</i>
-------------	-----------------------------------

Description

Joins pelagic fish/cephalopod/gelatinous traits to a `taxify()` result by `accepted_name`.

Usage

```
add_pelagic(x, cols = NULL, verbose = TRUE)
```

Arguments

- `x` A data.frame returned by `taxify()`.
- `cols` Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See `enrichment_cols`.
- `verbose` Logical. Default TRUE.

Details

Source: Gleiber et al. (2024) Pelagic Species Trait Database (Borealis, CC-BY 4.0).

Value

The same data.frame with pelagic_ columns: numeric `depth_min_m`, `depth_max_m`, `temp_min_c`, `temp_max_c`, `temp_mean_c`, `length_min_tl_cm`, `length_max_tl_cm`, `trophic_level`; categorical `vert_habitat`, `horz_habitat`, `body_shape`, `phys_defense`, `gregarious`.

References

Gleiber MR et al. (2024) A trait database for pelagic species. Scientific Data. doi:10.5683/SP3/0YFJED

Examples

```
taxify("Thunnus albacares", backend = "gbif") |>
  add_pelagic()
```

add_phylacine	<i>Add mammal traits including extinct species (PHYLACINE)</i>
---------------	--

Description

Joins PHYLACINE mammal traits to a [taxify\(\)](#) result by looking up accepted_name. PHYLACINE covers extant plus recently and prehistorically extinct mammals; it is offered alongside [add_pantheria\(\)](#) and [add_combine\(\)](#), not as a replacement.

Usage

```
add_phylacine(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: PHYLACINE v1.2 (Faurby et al. 2018, Ecology, CC0). Coverage: ~5.8k mammal species including extinct taxa.

Value

The same data.frame with additional columns:

phylacine_mass_g Body mass (g).
phylacine_diet_plant_pct Percent of diet that is plant.
phylacine_diet_vertibrate_pct Percent of diet that is vertebrate.
phylacine_diet_invertebrate_pct Percent of diet that is invertebrate.
phylacine_terrestrial Terrestrial habit (0/1).
phylacine_marine Marine habit (0/1).
phylacine_freshwater Freshwater habit (0/1).
phylacine_aerial Aerial habit (0/1).
phylacine_island_endemicity Island endemicity class.
phylacine_iucn_status IUCN status (includes EP = extinct in prehistory, EX, EW).

References

Faurby S et al. (2018) PHYLACINE 1.2: The Phylogenetic Atlas of Mammal Macroecology. Ecology 99:2626. doi:[10.1002/ecy.2443](https://doi.org/10.1002/ecy.2443)

Examples

```
taxify("Mammuthus primigenius", backend = "gbif") |>
  add_phylacine()
```

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

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>verbose</code>	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_pottier	<i>Add amphibian heat tolerance (Pottier)</i>
-------------	---

Description

Joins amphibian upper thermal-limit and body-size summaries to a [taxify\(\)](#) result by accepted_name. Per-measurement records are reduced to species medians; heat tolerance pools across metrics and acclimation conditions, so it is an approximate species-level upper thermal limit.

Usage

```
add_pottier(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: Pottier et al. (2022) amphibian heat tolerance database (Scientific Data, CC-BY 4.0).

Value

The same data.frame with numeric columns pottier_heat_tolerance_c, pottier_acclimation_temp_c, pottier_svl_mm, pottier_body_mass_g.

References

Pottier P et al. (2022) A comprehensive database of amphibian heat tolerance. Scientific Data 9:600. doi:[10.1038/s41597022017049](https://doi.org/10.1038/s41597022017049)

Examples

```
taxify("Rana temporaria", backend = "gbif") |>
  add_pottier()
```

add_quimbayo

Add reef-fish traits (Quimbayo)

Description

Joins Atlantic and Eastern-Pacific reef-fish life-history, ecology and behaviour traits to a [taxify\(\)](#) result by accepted_name.

Usage

```
add_quimbayo(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: Quimbayo et al. (2021) reef-fish trait database (ESA data paper; Zenodo, open).

Value

The same data.frame with quimbayo_ columns: numeric body_size_max_cm, aspect_ratio, trophic_level, depth_min_m, depth_max_m, temp_occurrence_mean_c; categorical home_range, diel_activity, water_level, body_shape, mouth_position, diet, spawning, size_group.

References

Quimbayo JP et al. (2021) Life-history traits, geographical range, and conservation aspects of reef fishes. Ecology. [doi:10.5281/zenodo.4455016](https://doi.org/10.5281/zenodo.4455016)

Examples

```
taxify("Thalassoma bifasciatum", backend = "gbif") |>
  add_quimbayo()
```

add_ramond	<i>Add marine protist functional traits (Ramond et al.)</i>
------------	---

Description

Joins genus-level morphological, behavioural and ecological traits of marine protists to a [taxify\(\)](#) result by looking up genus, so any species in a covered genus is annotated.

Usage

```
add_ramond(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: Ramond et al. (SEANOE, CC BY 4.0), functional traits of marine protists. Traits are genus-level; numeric cell size is aggregated by median, categorical traits by mode.

Value

The same data.frame with additional columns. The curated set:

ramond_shape Cell shape (round, elongated, amoeboid, ...).
ramond_motility Motility mode (swimmer, floater, gliding, ...).
ramond_ingestion Ingestion / trophic mode (phagotrophic, ...).
ramond_chloroplast Chloroplast presence (1 = present).
ramond_symbiotic Symbiotic relationship (parasite, mutualist, ...).
ramond_colony Colony form.
ramond_salinity Salinity preference.
ramond_size_min_um Minimum cell size (micrometres).
ramond_size_max_um Maximum cell size (micrometres).

With cols = "all" the full set of behavioural, ecological and symbiosis descriptors is attached under their source names.

References

Ramond P, Siano R, Sourisseau M (2018) Functional traits of marine protists. SEANOE. doi:[10.17882/51662](https://doi.org/10.17882/51662)

Examples

```
taxify("Alexandrium", backend = "gbif") |>
  add_ramond()
```

add_repttraits	Add reptile ecological traits and distribution (<i>ReptTraits</i>)
----------------	--

Description

Joins species-level reptile traits from ReptTraits (Oskyrko et al. 2024) to a `taxify()` result by looking up `accepted_name`. ReptTraits is built on the Reptile Database taxonomy, so it joins cleanly against the `reptiledb` backbone (and any backbone that resolves to Reptile Database accepted names).

Usage

```
add_repttraits(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by <code>taxify()</code> .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
verbose	Logical. Default TRUE.

Details

The layer carries a per-species distribution signal – biogeographic realm, elevation range and mean climate – alongside body-size and life-history traits, across all reptiles (snakes, lizards, amphisbaenians, turtles, crocodiles and the tuatara), not lizards only.

Source: ReptTraits v1.2 (Oskyrko et al. 2024, Scientific Data, CC BY 4.0). Coverage: 12,060 reptile species. The biogeographic realm and climate fields give a coarse, realm-level range signal; they are not a fine-grained (TDWG-level) range like the plant ranges used by the region constraint.

Value

The same data.frame with additional columns:

biogeographic_realm Main biogeographic realm (e.g. Neotropic, Palearctic, Afrotropic, Australo-Pacific, Marine).

microhabitat Microhabitat (e.g. Terrestrial, Saxicolous, Arboreal).

habitat_type Habitat type(s) (e.g. Forest, Desert, Wetlands).

elevation_min_m Minimum recorded elevation in metres.

elevation_max_m Maximum recorded elevation in metres.

- mean_annual_temp_c** Mean annual temperature across the range (degrees Celsius).
- insular_endemic** Whether the species is insular/endemic ("Yes"/"No").
- body_mass_g** Maximum body mass in grams.
- svl_mm** Maximum snout-vent length (straight carapace length for turtles) in mm.
- total_length_mm** Maximum total length in mm.
- longevity_yr** Maximum longevity in years.
- diet** Diet category (e.g. Carnivorous, Herbivorous, Omnivorous).
- reproductive_mode** Reproductive mode (oviparous/viviparous/...).
- clutch_size** Mean clutch or litter size.
- active_time** Activity time (Diurnal/Nocturnal/Cathemeral).
- foraging_mode** Foraging mode (ACT active / AMB ambush / Mixed).

References

Oskyrko O, Mi C, Meiri S, Du W (2024) ReptTraits: a comprehensive dataset of ecological traits in reptiles. Scientific Data 11:243. [doi:10.1038/s41597024030795](https://doi.org/10.1038/s41597024030795)

Examples

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

taxify("Pogona vitticeps", backend = "reptiledb") |>
  add_repttraits()

options(old)
```

add_rimet_phyto	Add phytoplankton cell metrics (Rimet & Druart)
-----------------	---

Description

Joins cell-level morphometrics for temperate-lake phytoplankton to a `taxify()` result by `accepted_name`.

Usage

```
add_rimet_phyto(x, cols = NULL, verbose = TRUE)
```

Arguments

- x** A data.frame returned by `taxify()`.
- cols** Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See `enrichment_cols`.
- verbose** Logical. Default TRUE.

Details

Source: Rimet & Druart (2018) phytoplankton metrics database (Zenodo, CC-BY 4.0).

Value

The same data.frame with numeric columns rimet_phyto_cell_length_um, rimet_phyto_cell_width_um, rimet_phyto_cell_thickness_um, rimet_phyto_cell_surface_area_um2, rimet_phyto_cell_biovolume_um3.

References

Rimet F, Druart JC (2018) A trait database for phytoplankton of temperate lakes. Zenodo. doi:10.5281/zenodo.1164834

Examples

```
taxify("Asterionella formosa", backend = "gbif") |>
  add_rimet_phyto()
```

add_saproxylic	<i>Add saproxylic beetle morphology (Hagge)</i>
----------------	---

Description

Joins European deadwood-beetle body and appendage morphometrics to a `taxify()` result by accepted_name.

Usage

```
add_saproxylic(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by <code>taxify()</code> .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
verbose	Logical. Default TRUE.

Details

Source: Hagge et al. (2021) saproxylic beetle morphology (Dryad, CC0).

Value

The same data.frame with numeric saproxylic_ columns: body_length_mm, body_width_mm, body_height_mm, mass_mg, colour_lightness, head_length_mm, pronotum_length_mm, elytra_length_mm, wing_length_mm, wing_aspect, antenna_length_mm, eye_length_mm.

References

Hagge J et al. (2021) Morphological trait database of European saproxylic beetles. Dryad. doi:10.5061/dryad.2fqz612p3

Examples

```
taxify("Rhysodes sulcatus", backend = "gbif") |>
  add_saproxylic()
```

add_sealifebase	<i>Add aquatic-life traits (SeaLifeBase)</i>
-----------------	--

Description

Joins SeaLifeBase morphological and ecological traits to a `taxify()` result by looking up `accepted_name`. SeaLifeBase is the non-fish companion to FishBase: molluscs, crustaceans, echinoderms, marine mammals, reptiles and other aquatic organisms. For fishes, use `add_fishbase()`.

Usage

```
add_sealifebase(x, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
<code>verbose</code>	Logical. Default TRUE.

Details

Source: SeaLifeBase via rfishbase (Palomares & Pauly, CC BY-NC 4.0). Non-fish aquatic life 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:

sb_body_length_cm Maximum body length in centimetres.

sb_body_mass_g Body mass in grams where available.

sb_trophic_level Trophic level.

sb_depth_min_m Minimum depth in metres.

sb_depth_max_m Maximum depth in metres.

- sb_vulnerability** Vulnerability index (0–100).
- sb_habitat** Habitat type (e.g. benthic, pelagic).
- sb_importance** Commercial importance category.

References

Palomares MLD, Pauly D (eds.) (2024) SeaLifeBase. World Wide Web electronic publication, <https://www.sealifebase.org>.

Examples

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

taxify("Octopus vulgaris", backend = "gbif") |>
  add_sealifebase()

options(old)
```

add_sharkipedia	<i>Add elasmobranch life-history traits (Sharkipedia)</i>
-----------------	---

Description

Joins Sharkipedia shark and ray life-history traits to a `taxify()` result by `accepted_name`. Long-format observations are reduced to one value per species (numeric traits by median) at build time.

Usage

```
add_sharkipedia(x, cols = NULL, verbose = TRUE)
```

Arguments

- `x` A data.frame returned by `taxify()`.
- `cols` Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See `enrichment_cols`.
- `verbose` Logical. Default TRUE.

Details

Source: Sharkipedia (Mull et al. 2022, Scientific Data, CC-BY 4.0).

Value

The same data.frame with additional columns:

- sharkipedia_lmax_cm** Maximum observed length (cm).
- sharkipedia_vbgf_linf_cm** von Bertalanffy asymptotic length Linf (cm).
- sharkipedia_vbgf_k** von Bertalanffy growth coefficient k (per year).
- sharkipedia_vbgf_t0** von Bertalanffy t0 (years).
- sharkipedia_length_first_maturity_cm** Length at first maturity (cm).
- sharkipedia_length_birth_cm** Length at birth (cm).
- sharkipedia_amax_observed_yr** Maximum observed age (years).
- sharkipedia_age_first_maturity_yr** Age at first maturity (years).
- sharkipedia_uterine_fecundity** Uterine fecundity.
- sharkipedia_gestation_length** Gestation length.
- sharkipedia_natural_mortality** Natural mortality M.

References

Mull CG et al. (2022) Sharkipedia: a curated open access database of shark and ray life history traits and abundance time-series. Scientific Data 9:559. [doi:10.1038/s41597022016551](https://doi.org/10.1038/s41597022016551)

Examples

```
taxify("Carcharodon carcharias", backend = "gbif") |>
  add_sharkipedia()
```

add_sheld	Add freshwater mussel traits (SHELD)
-----------	--------------------------------------

Description

Joins US freshwater-mussel life-history and host traits to a `taxify()` result by accepted_name.

Usage

```
add_sheld(x, cols = NULL, verbose = TRUE)
```

Arguments

- x** A data.frame returned by `taxify()`.
- cols** Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See `enrichment_cols`.
- verbose** Logical. Default TRUE.

Details

Source: SHELD (Hopper et al. 2023, Scientific Data, CC-BY 4.0).

Value

The same data.frame with sheld_ columns: numeric mean_length_mm, max_length_mm, mature_age, max_age, growth_rate, fecundity, n_host_species, n_host_family; categorical brood, marsupial_gills, hermaphrodite, shell_sculpture.

References

Hopper GW et al. (2023) A trait dataset for freshwater mussels of the United States of America. Scientific Data 10:745. doi:10.1038/s41597023026359

Examples

```
taxify("Lampsilis cardium", backend = "gbif") |>
  add_sheld()
```

add_spider_traits	Add spider traits (World Spider Trait Database)
-------------------	---

Description

Joins species-level spider morphometric and ecological traits to a `taxify()` result by looking up accepted_name. Values are aggregated from the World Spider Trait Database (numeric traits by median, categorical traits by mode); access-restricted source records are excluded.

Usage

```
add_spider_traits(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by <code>taxify()</code> .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
verbose	Logical. Default TRUE.

Details

Source: World Spider Trait Database (Pekar et al. 2021, Database, CC BY 4.0). Coverage: ~7.3k spider species. Morphometry is sexually dimorphic in spiders; the value here is the across-record median and is not split by sex.

Value

The same data.frame with additional columns:

spider_body_length_mm Body length (mm).
spider_prosoma_length_mm Cephalothorax (prosoma) length (mm).
spider_prosoma_width_mm Cephalothorax (prosoma) width (mm).
spider_abdomen_length_mm Abdomen (opisthosoma) length (mm).
spider_leg1_length_mm Leg I length (mm).
spider_ballooning Ballooning (aerial dispersal): yes/no.
spider_web_building Web building: yes/no.
spider_hunting_guild Hunting guild.
spider_web_type Web type.
spider_circadian_activity Circadian activity (diurnal/nocturnal).
spider_stratum Vertical stratum (habitat layer).

References

Pekar S et al. (2021) The World Spider Trait database: a centralized global open repository for curated data on spider traits. Database 2021:baab064. doi:[10.1093/database/baab064](https://doi.org/10.1093/database/baab064)

Examples

```
taxify("Araneus diadematus", backend = "gbif") |>
  add_spider_traits()
```

add_tetradensity	<i>Add population density (TetraDENSITY)</i>
------------------	--

Description

Joins species-median terrestrial-vertebrate population density to a `taxify()` result by accepted_name. Only ind/km2 records are used.

Usage

```
add_tetradensity(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by <code>taxify()</code> .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
verbose	Logical. Default TRUE.

Details

Source: Santini et al. TetraDENSITY (figshare, CC-BY 4.0). Records in other density units are excluded to avoid mixing.

Value

The same data.frame with numeric tetradensity_density_ind_km2.

References

Santini L et al. TetraDENSITY: a database of population density estimates in terrestrial vertebrates. figshare. doi:10.6084/m9.figshare.5371633

Examples

```
taxify("Capreolus capreolus", backend = "gbif") |>
  add_tetradensity()
```

add_thermofresh	<i>Add freshwater thermal-tolerance traits (ThermoFresh)</i>
-----------------	--

Description

Joins species-level critical thermal limits for freshwater fish, invertebrates and amphibians to a [taxify\(\)](#) result by looking up accepted_name. All values are in degrees Celsius.

Usage

```
add_thermofresh(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: the Freshwater thermal-tolerance database (Helena Bayat and contributors, Zenodo, CC BY 4.0). Each source record is one tolerance test; values are reduced to species-level medians per metric.

Value

The same data.frame with additional columns:

thermofresh_ctmax Critical thermal maximum (degrees C).

thermofresh_ctmin Critical thermal minimum (degrees C).

thermofresh_It50 Median lethal temperature (degrees C).

thermofresh_Itmax Lethal thermal maximum (degrees C).

thermofresh_Itmin Lethal thermal minimum (degrees C).

References

Freshwater thermal-tolerance database. Zenodo. [doi:10.5281/zenodo.14056760](https://doi.org/10.5281/zenodo.14056760)

Examples

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

add_trait

Add a trait from every source that carries it

Description

Attaches a single harmonized trait (e.g. woodiness, plant height) to a `taxify()` result, pulling from every enrichment source that provides it and reconciling their differing vocabularies and units. Where the per-source `add_*`() doors each join one dataset, `add_trait()` is the cross-source verb: you name the trait, it gathers the sources.

Usage

```
add_trait(
  x,
  trait,
  sources = "all",
  mode = c("coalesce", "wide"),
  combine = NULL,
  priority = NULL,
  verbose = TRUE
)
```

Arguments

x	A data.frame returned by <code>taxify()</code> .
trait	Character. A single trait name; see <code>list_traits()</code> for the available traits and <code>trait_info()</code> for a trait's sources and units.
sources	Which sources to use. Either the string "all" (the default) for every source registered for the trait, or a character vector of source names (see <code>trait_info()</code>).
mode	One of "coalesce" (default) or "wide". "coalesce" reduces the sources to one value per row (see <code>combine</code>). "wide" attaches one harmonized column per source.
combine	How mode = "coalesce" reduces the per-source values for a row. NULL (default) is method-aware: when the trait's sources agree in method it uses "median" for numeric traits and "first" for categorical traits; when sources measure the trait differently (a source carries a caution, e.g. maximum vs fine-root diameter) it uses "complete" instead of blending them. Numeric options: "median", "mean", "first" (highest-priority source that has a value), "min", "max", "complete" (the single most populated source, reported verbatim). Categorical options: "first", "vote" (majority across sources, ties broken by priority), or "complete". Median is the numeric default for concordant sources because trait values are skewed, so a single outlier should not decide the value; "complete" is the default for discordant sources because a median across methods matches no method. Passing combine explicitly overrides this and is applied to all sources.
priority	Character vector of source names giving the priority order (highest priority first), used by combine = "first", for tie-breaking combine = "vote", and to break ties in combine = "complete". Only used when mode = "coalesce"; defaults to the registered order for the trait (see <code>trait_info()</code>).
verbose	Logical. Default TRUE.

Details

Each source keeps its provenance. The default "coalesce" mode reduces the sources to one value per row plus the columns that document it – its unit, the sources that contributed, how many, and a caution when the sources measure the trait by different methods. The opt-in "wide" mode instead gives every source its own column (`<trait>_<source>`), so per-source agreement and conflict stay fully visible.

Harmonization is per source: a categorical source is mapped to the trait's shared vocabulary, and a numeric source is converted to the trait's canonical unit. For example, GIFT seed mass (grams) and Diaz et al. seed mass (milligrams) both arrive as milligrams. The mappings and units for a trait are listed by `trait_info()`.

Some sources report the same trait in the right unit but under a different definition (for example AusTraits root diameter is a maximum including coarse roots, while GRooT is fine-root only). Such a source carries a caution in the registry. In mode = "coalesce" with the default combine, a trait whose sources disagree in method is not blended: the most complete source is reported and `<trait>_caution` records the difference. `trait_info()` lists each source's harmonization note and caution.

A source enrichment that is not installed and cannot be downloaded or built is skipped with a warning, and the trait is assembled from the sources that are available.

Value

The same data.frame with added columns.

mode = "coalesce" <trait> (the reduced value); <trait>_unit (the canonical unit, numeric traits only); <trait>_sources (the source it came from, or the comma-separated contributing sources with an aggregating combine); <trait>_n (how many sources backed the value); and, only when a source measured the trait differently, <trait>_caution explaining the method difference. To inspect every source, use mode = "wide".

mode = "wide" One column per source, <trait>_<source>, each harmonized to the trait's shared vocabulary (categorical) or unit (numeric); <trait>_unit; and <trait>_caution on rows where a cautioned source supplied a value.

Numeric traits are returned in the trait's canonical unit (see [trait_info\(\)](#)); rows absent from a source get NA.

See Also

[list_traits\(\)](#) to see available traits, [trait_info\(\)](#) for a trait's sources and units. The per-source doors ([add_zanne\(\)](#), [add_gift\(\)](#), [add_diaz_traits\(\)](#), [add_leda\(\)](#)) join one dataset each.

Examples

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

# One coalesced value plus its provenance (unit, sources, count):
taxify("Abies alba") |>
  add_trait("seed_mass")

# One column per source, to inspect agreement and conflict:
taxify("Abies alba") |>
  add_trait("woodiness", mode = "wide")

options(old)
```

add_tree_of_sex	<i>Add sex-determination traits (Tree of Sex)</i>
-----------------	---

Description

Joins sexual-system and sex-determination traits to a [taxify\(\)](#) result by looking up accepted_name. Covers plants, vertebrates and invertebrates; some traits are group-specific (selfing for plants, environmental sex determination for vertebrates, haplodiploidy for invertebrates).

Usage

```
add_tree_of_sex(x, cols = NULL, verbose = TRUE)
```


Arguments

x	A data.frame returned by <code>taxify()</code> .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
verbose	Logical. Default TRUE.

Details

Source: Tree of Sex (Tree of Sex Consortium 2014, Scientific Data, CC0). Coverage: ~37.5k species across plants, vertebrates and invertebrates.

Value

The same data.frame with additional columns:

- tos_taxon_group** Source group (plants/vertebrates/invertebrates).
- tos_sexual_system** Sexual system (vocabulary differs by group).
- tos_karyotype** Sex-chromosome system (XY/ZW/XO/homomorphic/...).
- tos_genotypic** Heterogamety (male/female heterogametic/GSD/...).
- tos_molecular_basis** Molecular basis (Y dominant/W dominant/dosage).
- tos_selfing** Selfing (plants; self compatible/incompatible).
- tos_environmental_sd** Environmental sex determination (vertebrates; TSD/...).
- tos_haplodiploidy** Haplodiploidy (invertebrates).

References

The Tree of Sex Consortium (2014) Tree of Sex: a database of sexual systems. Scientific Data 1:140015. doi:10.1038/sdata.2014.15

Examples

```
taxify("Silene latifolia", backend = "gbif") |>
  add_tree_of_sex()
```

add_usda_fungus_host *Add fungal host breadth (USDA Fungus-Host Dataset)*

Description

Joins per-fungus host breadth from the USDA National Fungus Collections Fungus-Host Dataset to a `taxify()` result by looking up `accepted_name`. Each fungus species is summarised by how many distinct host plants and host plant genera it has been recorded on.

Usage

```
add_usda_fungus_host(x, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See <code>enrichment_cols</code> .
<code>verbose</code>	Logical. Default TRUE.

Details

Source: Farr, Rossman & Castlebury (2021) United States National Fungus Collections Fungus-Host Dataset, Ag Data Commons (U.S. Public Domain). Fungi only; ~99k species with at least one recorded host.

Value

The same data.frame with additional columns:

fungus_host_count Number of distinct host plant species recorded.

fungus_host_genus_count Number of distinct host plant genera recorded.

References

Farr DF, Rossman AY, Castlebury LA (2021) United States National Fungus Collections Fungus-Host Dataset. Ag Data Commons. doi:10.15482/USDA.ADC/1524414

Examples

```
taxify("Puccinia graminis", backend = "gbif") |>
  add_usda_fungus_host()
```

add_useful_plants	<i>Add human-use categories (World Checklist of Useful Plant Species)</i>
-------------------	---

Description

Joins plant human-use categories to a `taxify()` result by looking up `accepted_name`. Each of the ten Level-1 use categories is a 0/1 flag, plus a crop-wild-relative flag.

Usage

```
add_useful_plants(x, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>cols</code>	Character vector of use-category columns to attach, or "all". NULL (default) attaches all eleven. See <code>enrichment_cols()</code> .
<code>verbose</code>	Logical. Default TRUE.

Details

Source: World Checklist of Useful Plant Species (Diazgranados et al. 2020, KNB, CC BY 4.0).
Coverage: ~39k plant species.

Value

The same data.frame with additional columns:

useful_animal_food Animal food (0/1).
useful_environmental_uses Environmental uses (0/1).
useful_fuels Fuels (0/1).
useful_gene_sources Gene sources (0/1).
useful_human_food Human food (0/1).
useful_invertebrate_food Invertebrate food (0/1).
useful_materials Materials (0/1).
useful_medicines Medicines (0/1).
useful_poisons Poisons (0/1).
useful_social_uses Social uses (0/1).
useful_crop_wild_relative Crop wild relative (0/1).

References

Diazgranados M et al. (2020) World Checklist of Useful Plant Species. Knowledge Network for Biocomplexity. doi:10.5063/F1CV4G34

Examples

```
taxify("Acorus calamus", backend = "gbif") |>
  add_useful_plants()
```

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, cols = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	A data.frame returned by <code>taxify()</code> .
<code>region</code>	Character. TDWG Level 3 region code(s), or "all". See <code>taxify_regions()</code> for the full list of codes. • Single code (e.g., "BGM" for Belgium): adds <code>native_status</code> column (no suffix). • Multiple codes (e.g., c("BGM", "GER")): adds <code>native_status_BGM</code>, <code>native_status_GER</code>. • "all": adds one column per region in the dataset.
<code>cols</code>	Which columns to attach. NULL (the default) attaches the curated set; a character vector of column names attaches just those, and "all" attaches every column the source carries (see <code>enrichment_cols()</code>).
<code>verbose</code>	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_zanne

Add woodiness (Zanne et al. 2014)

Description

Joins the woody / herbaceous classification of Zanne et al. (2014) to a [taxify\(\)](#) result by looking up accepted_name. This is the source-named door for the Zanne Global Woodiness Database; for woodiness reconciled across every source that carries it (Zanne, GIFT), use [add_trait\(\)](#) with "woodiness".

Usage

```
add_zanne(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
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.

See Also

[add_trait\(\)](#) for woodiness harmonized across sources.

Examples

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

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

options(old)
```

add_zooplankton	<i>Add marine zooplankton traits</i>
-----------------	--------------------------------------

Description

Joins global marine-zooplankton traits to a [taxify\(\)](#) result by accepted_name (species-level summaries).

Usage

```
add_zooplankton(x, cols = NULL, verbose = TRUE)
```

Arguments

x	A data.frame returned by taxify() .
cols	Which columns to attach: NULL (default) the curated set, "all" every column the source carries, or a character vector of names. See enrichment_cols .
verbose	Logical. Default TRUE.

Details

Source: Pata & Hunt global marine zooplankton trait database (Zenodo, CC-BY-SA 4.0).

Value

The same data.frame with zooplankton_ columns: numeric body_length_max_mm, carbon_weight_mg, nitrogen_pdw_pct; categorical vertical_distribution, reproduction_mode, trophic_group, feeding_mode, myelination, habitat_association, diel_vertical_migration, bioluminescence.

References

Pata PR, Hunt BPV (2025) A global trait database for marine zooplankton. Zenodo. [doi:10.5281/zenodo.8102913](https://doi.org/10.5281/zenodo.8102913)

Examples

```
taxify("Calanus finmarchicus", backend = "gbif") |>
  add_zooplankton()
```

 children

List the accepted taxa within a genus or family

Description

Returns the accepted taxa a backbone places inside a genus or family, so you can build a checklist from the backbone rather than only validating one. The parent is auto-detected: a genus is tried first, then a family.

Usage

```
children(taxon, backend = "wfo", rank = "species", verbose = TRUE)
```

Arguments

taxon	A single genus or family name.
backend	A single backend name (e.g. "wfo") or a taxify_backend object. Default "wfo".
rank	Rank of the children to return ("species" by default), or "any" for every rank below the parent.
verbose	Logical. Default TRUE.

Value

A data.frame of accepted taxa, columns: name, authorship, rank, family, genus, taxon_id, parent_rank ("genus" or "family"), backend. Empty if the parent is not found.

See Also

[synonyms\(\)](#), [taxify\(\)](#).

Examples

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

children("Quercus")

options(old)
```

cite	<i>Cite data sources used in a taxify result</i>
------	--

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"))
```

enrichment_cols	<i>Browse the trait columns an enrichment door can attach</i>
-----------------	---

Description

Lists the columns available from an enrichment's pre-built .vtr, so you can choose which to attach through the doors that accept a cols argument (such as [add_gift\(\)](#) and [add_floraweb\(\)](#)). Read offline from the local .vtr; the first call may trigger the one-time download.

Usage

```
enrichment_cols(source)
```

Arguments

source	Character. An enrichment name (see list_enrichments()).
--------	--

Value

A data.frame with one row per column: column (the name) and type ("numeric" or "character").

See Also

`add_gift()`, `add_floraweb()`, `list_enrichments()`

Examples

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

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 <code>taxify()</code> .
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)
```

gift_traits

Browse the bundled GIFT trait columns

Description

Returns the species-level trait columns available from the bundled GIFT enrichment, so you can pick which to attach in [add_gift\(\)](#). Read offline from the local `.vtr` (downloaded or built once); the first call may trigger that one-time download.

Usage

```
gift_traits()
```

Value

A data.frame with one row per trait column:

column The gift_<trait> column name.

type "numeric" or "character".

See Also

[add_gift\(\)](#), [enrichment_cols\(\)](#) for the same listing on any enrichment.

Examples

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

inspect

Inspect a name list for probable typos and other anomalies

Description

A quality-control pass over a name list. By default `inspect()` does not match names against backbones: on a plain character vector it runs the checks that need no matching – the genus register and the rest of the batch – and is fast and offline. To also surface the match-based anomalies (typo, synonym, ambiguous, geographic), either set `backbones = TRUE` (matches against every installed backbone, listed in the report) or match yourself first and inspect the result (`taxify(x) |> inspect()`). Either way it returns only the rows that look anomalous, each labelled with what stands out and, where known, the name to use instead.

Usage

```
inspect(
  x,
  backbones = FALSE,
  region = NULL,
  coords = NULL,
  range = c("present", "native", "introduced"),
  min_tier = c("note", "review", "unresolved"),
  verbose = TRUE
)
```

Arguments

<code>x</code>	A character vector of names, or a <code>taxify_result</code> from <code>taxify()</code> .
<code>backbones</code>	Logical. When <code>x</code> is a character vector, <code>TRUE</code> matches it against every installed backbone (via <code>taxify()</code>) so the match-based labels are available; <code>FALSE</code> (default) runs the register and list checks only, with no matching. The backbones used are printed in the report header. Ignored when <code>x</code> is already a <code>taxify_result</code> (it was matched already).
<code>region</code> , <code>coords</code> , <code>range</code>	Geographic constraint for the <code>geographic</code> / <code>out_of_range</code> checks, as in <code>taxify()</code> . These act on a <code>taxify_result</code> (which carries the accepted names they need); on a character vector there is nothing matched to place, so they have no effect.
<code>min_tier</code>	Lowest tier to report: "note" (default, everything), "review", or "unresolved".
<code>verbose</code>	Logical. Print progress messages. Default <code>TRUE</code> .

Details

Checks that need no matching (run on a character vector or a result):

`unknown` The genus is not in the genus register – the union of all 13 backbones' genera – so no backbone recognises it. The strong "probably not a real name" signal.

`near_duplicate` A near-twin of a more frequent name in the same list (small edit distance), so probably a misspelling of it. Computed from the list alone, so it catches typos in names no backbone contains.

`outlier_group` The name's kingdom group (from the register) is a tiny minority of an otherwise group-coherent list – the lone animal or fungus among plants, typically a cross-kingdom homonym typo.

Checks read from a `taxify()` result (only present when you inspect one):

`typo` Resolved only after fuzzy correction (`match_type = "fuzzy"`): the input most likely contains a spelling error; suggestion is the name.

`ambiguous` A homonym resolving to more than one accepted taxon.

`geographic` The matched species is real but has no WCVF record in the declared region / coords (vascular plants only).

out_of_range No region declared, yet the matched species' range falls outside the list's main TDWG continents (skipped for globally spread lists).

case Resolved only after ignoring case (`match_type = "exact_ci"`).

synonym The input is an outdated synonym; suggestion is the current accepted name.

Rows with no anomaly are dropped.

Each row gets a tier describing what it needs, not how bad it is: **unresolved** (no usable name – act on it), **review** (a name is there but its identity is uncertain – verify it), or **note** (correct, optional cleanup). **unknown** is unresolved; the identity-uncertain labels are **review**; **case** and **synonym** are **note**. An anomaly may be intended, so the tier is a triage hint, not a verdict.

The list-context labels (**near_duplicate**, **out_of_range**, **outlier_group**) judge a name against the rest of the batch, so they cannot apply to a single name: `inspect()` on one name warns and reports only the per-name labels. The register checks (**unknown**, and the register-derived **outlier_group**) need the genus register installed; without it they are skipped (with a message at `verbose`).

Value

A `taxify_inspection` data.frame (one row per anomalous name, ordered most-notable first) with columns `input_name`, `suggestion` (the name to use instead, or NA), `anomalies` (| -joined labels), `tier` (ordered factor `note < review < unresolved`), `reason`, `fuzzy_dist`, and `backend`. Zero rows means nothing stood out.

See Also

`taxify()`, `taxify_regions()`

Examples

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

# On raw names: register + list checks (no matching)
inspect(c("Quercus robur", "Bogusus fakus", "Carexus mysteriosa",
          "Carexus mysteriosa", "Carexus mysteryosa"))

# Opt in to matching to also get typos, synonyms, ambiguity
inspect(c("Quercus robur", "Quercus robus"), backbone = TRUE)

# Or match yourself and inspect the result
taxify(c("Quercus robur", "Quercus robus")) |> inspect()

options(old)
```

<code>list_enrichments</code>	<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()
```

<code>list_traits</code>	<i>List the traits available to add_trait()</i>
--------------------------	---

Description

Returns the traits that `add_trait()` can attach across sources, with their kind, canonical unit, and the number and names of contributing sources.

Usage

```
list_traits()
```

Value

A data.frame with one row per trait:

trait The trait name to pass to `add_trait()`.

label Human-readable label.

kind "numeric" or "categorical".

unit Canonical unit for numeric traits, NA for categorical.

n_sources Number of sources providing the trait.

sources Comma-separated source names.

See Also

`add_trait()`, `trait_info()`

Examples

```
list_traits()
```

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_inspection
```

Print a taxify inspection report

Description

Print a taxify inspection report

Usage

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

Arguments

x	A taxify_inspection object.
...	Ignored.

Value

x, invisibly.

```
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	<i>Summarise a taxify_result</i>
-----------------------	----------------------------------

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.

synonyms	<i>List the synonyms of a name</i>
----------	------------------------------------

Description

The reverse of the forward resolution `taxify()` does: each input name is resolved to its accepted taxon, then every synonym that points to that accepted taxon in the backbone is returned. Useful for auditing which historical names collapse onto a current name.

Usage

```
synonyms(x, backend = "wfo", verbose = TRUE)
```

Arguments

x	Character vector of names (accepted names or synonyms; each is resolved to its accepted taxon first).
backend	A single backend name (e.g. "wfo") or a taxify_backend object. Default "wfo".
verbose	Logical. Default TRUE.

Value

A data.frame with one row per synonym found, columns:

input_name The queried name.

accepted_name The accepted name the query resolved to.

synonym A synonym of that accepted taxon.

authorship Authorship of the synonym.

rank Rank of the synonym.

taxon_id Backend ID of the synonym.

backend Backend used.

Names that resolve to an accepted taxon with no synonyms contribute no rows.

See Also

[taxify\(\)](#) for the forward direction, [children\(\)](#) to list the accepted taxa within a genus or family.

Examples

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

# Amphibolurus vitticeps is a synonym of Pogona vitticeps
synonyms("Pogona vitticeps", backend = "reptiledb")

options(old)
```

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"),
  aggregates = c("preserve", "collapse"),
```

```

    region = NULL,
    coords = NULL,
    range = c("present", "native", "introduced"),
    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 ($0 < \text{fuzzy_threshold} < 1$): normalized distance (edits / max name length). Default 0.2 is about 1 edit per 5 characters. • Integer ($\text{fuzzy_threshold} \geq 1$): 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).
aggregates	Character. How to treat species aggregates (names with an agg. / s.l. qualifier). "preserve" (default) keeps the aggregate as its own concept: it matches the backbone's aggregate taxon (" <binomial> aggr. ") where one exists, otherwise falls back to the binomial. When it falls back, the aggregate_fallback column is set TRUE so the aggregate-to-species collapse is visible rather than silent (only the aggregate-bearing backbones – Euro+Med, WoRMS – carry aggregate taxa, so preserve falls back for the others). "collapse" strips the marker and matches the binomial species, the way any non-aggregate name is matched. Either way the qualifier is recorded in the qualifier column.
region	TDWG botanical region(s) to constrain fuzzy matching to, or NULL (default) for no geographic constraint. Accepts Level 3 codes ("BGM", c("BGM", "GER")) or region names at any level, matched case- and accent-insensitively against the bundled WGSRPD crosswalk: a Level 3 name ("Belgium"), a Level 2 region ("Middle Europe"), or a Level 1 continent ("Europe", which expands to all its codes). See taxify_regions() for the full list. When set, fuzzy candidates are restricted to species with WCVF records in the region(s); exact matches are always kept. The filter only narrows genuinely ambiguous fuzzy candidates: a candidate is dropped only when the same input name has another candidate that is in-region or has no WCVF range data, so non-plant matches (no WCVF coverage) are never affected and a name whose only candidate is out-of-region is still returned. WCVF is vascular plants only, so this disambiguates plant names.
coords	Coordinates to constrain fuzzy matching to, mapped to TDWG regions by point-in-polygon and unioned with region. A single c(lon, lat) pair, a matrix/data.frame

	of longitude/latitude columns (named lon/lat or x/y, else the first two columns as lon, lat), or a point-geometry spatial object (an sf /sfc object or a terra SpatVector, reprojected to longitude/latitude automatically). NULL (default) for none. The WGSRPD boundary file is downloaded once and cached; coordinate lookup needs that download (or a prior cache). The point-in- polygon test uses terra or sf when installed, otherwise a native fallback; force the engine with <code>options(taxify.pip_engine = "terra" "sf" "native")</code> .
range	Character. Which WCVF statuses count as in-region when region or coords is set. "present" (default) accepts any record (native, introduced, or extinct) – the right choice for name disambiguation. "native" accepts only native records, "introduced" only introduced (alien) records; both fold an ecological filter into matching and are for callers who want that. Ignored when no region is set.
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?

qualifier Canonical taxonomic qualifier found in the input name ("cf.", "aff.", "agg.", "s.l.", "s.str.", "sp.", ...), or NA. Spelling variants are folded to one token ("aggr.", "agg" and "sensu lato" all map to "agg."/"s.l. ").

qualifier_position "genus" when the qualifier leads the name and qualifies the whole name (e.g. "Cf. Pinus sylvestris"), "species" when it qualifies the species (inline cf. or trailing agg.), NA when there is no qualifier.

aggregate_fallback Logical. For an aggregate query under `aggregates = "preserve"`: FALSE when it resolved to the backbone's dedicated aggregate taxon, TRUE when no such taxon existed and it fell back to the nominal binomial. NA for non-aggregate queries and under `aggregates = "collapse"`, where the collapse is explicit.

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)

# Constrain fuzzy candidates to a geographic region: a TDWG Level 3 code,
# or a region name resolved via the bundled WGSRPD crosswalk
taxify("Quercus robur", region = "EUR")
taxify("Quercus robur", region = "Belgium")

# Constrain by coordinates (downloads WGSRPD boundaries on first use)
## Not run:
taxify("Quercus robur", coords = c(4.35, 50.85))

## End(Not run)

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

options(old)
```

taxify_candidates	<i>Expand ambiguous matches into their candidate taxa</i>
-------------------	---

Description

Where `taxify()` meets an irreducible homonym – a name whose synonyms point to several accepted taxa at the same priority tier – it records one candidate in the scalar columns, sets `is_ambiguous = TRUE`, and lists the conflicting accepted taxon IDs in `ambiguous_targets`. This verb expands those rows into one row per candidate, resolved to full names against the backbone, so you can choose the right taxon yourself instead of relying on the automatic tiebreak.

Usage

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

Arguments

<code>x</code>	A <code>taxify()</code> result.
<code>verbose</code>	Logical. Default TRUE.

Value

A data.frame with one row per (ambiguous input, candidate taxon):

input_name The queried name.

chosen The accepted name `taxify()` picked for that row.

candidate A candidate accepted name.

authorship Authorship of the candidate.

rank Rank of the candidate.

family Family of the candidate.

genus Genus of the candidate.

taxon_id Backend ID of the candidate.

backend Backend used.

Empty when no rows were ambiguous.

See Also

`taxify()`, `synonyms()`.

Examples

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

# Homonyms are rare; on an unambiguous result this is an empty frame.
taxify("Quercus robur") |>
  taxify_candidates()

options(old)
```

taxify_clear_cache	<i>Clear all cached backbones</i>
--------------------	-----------------------------------

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>Build a backbone database from source</i>
-----------------	--

Description

Builds the backbone .vtr for a backend from its upstream Darwin Core source, delegating to the sibling taxifydb package (which must be installed). This is the from-source path, for rebuilding a backbone locally.

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

For everyday use you do not need this: [taxify\(\)](#) auto-downloads the pre-built .vtr on first use, and [taxify_download_vtr\(\)](#) fetches a pre-built backbone directly without taxifydb.

Value

The path to the .vtr file (invisibly).

See Also

[taxify_download_vtr\(\)](#) to fetch a pre-built backbone, [taxify\(\)](#) which downloads on first use.

taxify_download_enrichment	<i>Download one or more enrichment .vtr files</i>
----------------------------	---

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., "iucn", "griis", "zanne").
version	Character. "latest" (default) or a specific version string.
verbose	Logical. Default TRUE.

Details

Available enrichments:

iucn IUCN conservation status (LC/NT/VU/EN/CR/EW/EX)
griis GRIIS invasive species status by country
zanne 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	<i>Download a pre-built taxify backbone</i>
---------------------	---

Description

Downloads a pre-built .vtr backbone from GitHub Releases using the taxify manifest. This needs no build tools and does not require taxifydb; it is the fast path `taxify()` uses internally on first use. Call it directly to pre-fetch backbones before an offline session. Progress is always shown; no prompts are shown, so calling this function is consent.

Usage

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

Arguments

backend	Character. A backend name (e.g. "wfo", "col", "gbif", ...; see the backends in <code>list_enrichments()</code> 's companion manifest) or "register" for the genus register. Multiple backends can be given 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).

See Also

`taxify_download()` to build a backbone from source via taxifydb, `taxify_download_enrichment()` for enrichment layers.

taxify_example_data	<i>Path to the bundled example database</i>
---------------------	---

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_zanne()
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	<i>Load the unified genus register into memory</i>
----------------------	--

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

<code>x</code>	A data.frame, typically a <code>taxify()</code> result after applying a grouped enrichment like <code>add_griis()</code> , <code>add_alien_first_records()</code> , or <code>add_wcvp()</code> .
<code>cols</code>	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 <code>add_*</code> () functions.
<code>group_col</code>	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).
<code>drop_na</code>	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_griis()`, `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_griis(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_regions

List TDWG botanical regions

Description

Returns the bundled WGSRPD (World Geographical Scheme for Recording Plant Distributions) Level 3 crosswalk: the botanical-country codes and names used by the `region` argument of `taxify()` and by `add_wcvp()`. Optionally filtered by a search term matched (case- and accent-insensitively) against the code and the Level 1, 2, and 3 names.

Usage

```
taxify_regions(search = NULL)
```

Arguments

`search` Optional character string. If supplied, only regions whose code or name contains it are returned.

Value

A data.frame with columns `code`, `name`, `level2_name`, and `level1_name`, one row per Level 3 region.

Examples

```
head(taxify_regions())
taxify_regions("belgium")
taxify_regions("Europe")
```

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.

trait_info

Describe a trait's sources and units

Description

Prints the kind, canonical unit, and (for categorical traits) the shared vocabulary of a trait, and returns a data.frame of the sources [add_trait\(\)](#) draws from – one row per source, with its enrichment, source column, citation, and the harmonization note (unit conversion or vocabulary mapping).

Usage

```
trait_info(trait)
```

Arguments

trait Character. A single trait name; see [list_traits\(\)](#).

Value

A data.frame (invisibly-friendly) with columns source, enrichment, column, citation, note (unit or vocabulary harmonization), and caution (a method or definition difference from the reference source, or NA). The header line (label, kind, unit, default priority, vocabulary) is printed as a message.

See Also

[add_trait\(\)](#), [list_traits\(\)](#)

Examples

```
trait_info("woodiness")
```

Index

add_algae_traits, 4
add_alien_first_records, 5
add_alien_first_records(), 124
add_amniote, 6
add_amphibio, 8
add_anage, 9
add_animaltraits, 10
add_arctic_traits, 11
add_arthropod_traits, 12
add_austraits, 13
add_avonet, 15
add_avonet(), 23
add_bacdiv, 16
add_baseflor, 17
add_baseflor(), 37, 46, 47
add_bee_ostwald, 18
add_bet, 19
add_beukhof, 21
add_bien, 22
add_birdbase, 23
add_blanchard, 24
add_brot, 25
add_chelonians, 26
add_classification, 27
add_col_info, 28
add_col_info(), 28
add_combine, 29
add_combine(), 80
add_common_names, 30
add_common_names(), 124
add_coral_traits, 31
add_data, 32
add_data(), 106
add_diaz_traits, 35
add_diaz_traits(), 96
add_disperse, 36
add_ecoflora, 37
add_ecoflora(), 46, 47
add_edwards_phyto, 38
add_eive, 39
add_eive(), 17, 37, 47, 81
add_elton_traits, 40
add_eupolltrait, 42
add_eurobat, 43
add_fishbase, 44
add_fishbase(), 45, 88
add_fishmorph, 45
add_floraweb, 46
add_floraweb(), 37, 105, 106
add_freshwater_insects_conus, 48
add_frugivoria, 49
add_fungal_traits, 51
add_fungalroot, 50
add_funguild, 52
add_gbif_info, 54
add_gbif_info(), 28
add_gift, 55
add_gift(), 96, 105–107
add_globi, 56
add_globtherm, 57
add_glonaf, 58
add_glonaf(), 59
add_griis, 59
add_griis(), 124
add_groot, 60
add_gwdd, 62
add_homerange, 63
add_hosts, 64
add_huang_amph, 65
add_hybrid_info, 66
add_italic, 66
add_iucn, 68
add_kew_sid, 69
add_leda, 70
add_leda(), 17, 96
add_leptrait, 71
add_madin, 72
add_nesttrait, 73

- add_nztd, 74
- add_octocoral, 75
- add_odonata, 76
- add_pantheria, 77
- add_pantheria(), 29, 80
- add_parravicini, 78
- add_pelagic, 79
- add_phylacine, 80
- add_pignatti, 81
- add_pottier, 82
- add_quimbayo, 83
- add_ramond, 84
- add_repttraits, 85
- add_rimet_phyto, 86
- add_saproxylic, 87
- add_sealifebase, 88
- add_sharkipedia, 89
- add_sheld, 90
- add_spider_traits, 91
- add_tetradensity, 92
- add_thermofresh, 93
- add_trait, 94
- add_trait(), 38, 68, 102, 110, 111, 126, 127
- add_tree_of_sex, 96
- add_usda_fungus_host, 98
- add_useful_plants, 99
- add_wcvp, 100
- add_wcvp(), 124, 125
- add_wfo_info, 101
- add_zanne, 102
- add_zanne(), 96
- add_zooplankton, 103

- children, 104
- children(), 114
- cite, 105

- enrichment_cols, 4, 7–12, 14, 16, 17, 19, 21–26, 29, 31, 35–38, 40–45, 48, 49, 51, 53, 56, 57, 61–65, 67–80, 82–93, 97, 98, 102, 103, 105
- enrichment_cols(), 15, 30, 47, 60, 99, 100, 107
- export_data, 106

- gift_traits, 107
- gift_traits(), 55, 56

- inspect, 107

- list_enrichments, 110
- list_enrichments(), 105, 106, 122
- list_traits, 110
- list_traits(), 95, 96, 126, 127
- lookup_genus, 111

- print.taxify_inspection, 112
- print.taxify_result, 112

- summary.taxify_result, 113
- synonyms, 113
- synonyms(), 104, 118

- taxify, 114
- taxify(), 4–19, 21–33, 35–104, 106, 108, 109, 113, 114, 118–121, 124, 125
- taxify_candidates, 118
- taxify_clear_cache, 119
- taxify_data_dir, 119
- taxify_data_dir(), 120, 122
- taxify_download, 120
- taxify_download(), 122
- taxify_download_enrichment, 120
- taxify_download_enrichment(), 122
- taxify_download_vtr, 121
- taxify_download_vtr(), 120
- taxify_example_data, 122
- taxify_example_data(), 119
- taxify_load_register, 123
- taxify_long, 123
- taxify_long(), 33
- taxify_refresh_manifest, 125
- taxify_regions, 125
- taxify_regions(), 100, 109, 115
- taxify_register_coverage, 126
- tools::R_user_dir(), 119
- trait_info, 126
- trait_info(), 95, 96, 111