Package: wcvpmatch (via r-universe)

November 22, 2026

Title Taxonomic Name Reconciliation Against the 'WCVP' Backbone

Version 0.0.3

Author Paul Efren Santos Andrade [aut, cre, cph]

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

Description Standardizes and reconciles scientific plant names against a World Checklist of Vascular Plants ('WCVP')-style taxonomic backbone. The package parses names into taxonomic components and applies staged exact and fuzzy matching for binomial and trinomial inputs, including infraspecific rank-aware checks. It also returns accepted-name context and row-level matching flags to support reproducible, auditable preprocessing for downstream biodiversity, spatial, and trait analyses. A user-supplied backbone can be passed through 'target_df'; when the optional companion package 'wcvpdata' is installed, its default checklist can also be used.

License MIT + file LICENSE

Encoding UTF-8

Imports assertthat, cli, dplyr, fozziejoin, lifecycle, magrittr, memoise, purrr, stringdist, stringr, tibble, tidyr

Depends R (>= 4.1.0)

LazyData true

LazyDataCompression xz

Suggests knitr, pkgload, rlang (>= 1.0.0), rmarkdown, testthat (>= 3.0.0), wcvpdata (>= 0.7.0)

VignetteBuilder knitr

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

Config/testthat/edition 3

Config/Needs/website pkgdown

URL https://paulesantos.github.io/wcvpmatch/, https://github.com/PaulESantos/wcvpmatch

BugReports https://github.com/PaulESantos/wcvpmatch/issues

Config/roxygen2/version 8.1.1

NeedsCompilation no

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

Date/Publication 2026-11-22 16:27:44 UTC

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

RemoteRef HEAD

RemoteSha 08e48577144b66927c8eb1124450275088db5583

Contents

classify_spnames 2
fia 4
wcvp_distribution 4
wcvp_matching 6
wcvp_setup_info 8
wcvp_synonyms 9

Index **11**

`classify_spnames` *Classify Scientific Plant Names into Taxonomic Components*

Description

[Stable]

Parse and classify scientific plant names into taxonomic components: genus, specific epithet, infraspecific rank, infraspecific epithet, and author.

Output is aligned to a backbone convention:

- `Orig.Genus` in Title Case (first letter uppercase, rest lowercase).
- `Orig.Species` and `Orig.Infraspecies` epithets in lowercase.
- `Infra.Rank` in lowercase (`subsp.`, `var.`, `subvar.`, `f.`, `subf.`).
- `Author` is recovered from the input and preserved in its original casing/punctuation (no forced uppercasing).
- `Orig.Name` is reconstructed as: genus + species + (rank + infra) + author.

Robustness rules:

- `cf.` / `aff.` are removed from parsing but preserved as flags (`has_cf`, `has_aff`).
- Hybrid markers (`x`/`\u00D7`) as standalone tokens are removed with `had_hybrid = TRUE`.
- `sp.` / `spp.` triggers genus-only classification (`Rank = 1`, `Orig.Species = NA`) and sets `is_sp`/`is_spp`.

- If an infraspecific rank is present but the infraspecific epithet is missing, sets `rank_missing_infra = TRUE` and keeps `Infra.Rank` while `Orig.Infraspecies = NA`.
- If rank appears "late" (after author-like tokens), parsing is best-effort and `rank_late = TRUE`.
- If there is no explicit rank and a third token exists, the function can infer an unranked infraspecific epithet when the third token looks epithet-like (all lowercase), and does not look like the start of an author. In that case `implied_infra = TRUE`, `Orig.Infraspecies` is filled, `Infra.Rank = NA`, and `Rank = 3`.

Usage

```
classify_spnames(splist)
```

Arguments

`splist`	Character vector. Scientific plant names.

Value

A tibble with one row per input name and standardized columns/flags:

sorter Numeric index of original order.

Input.Name Original input string as provided by user.

Orig.Name Reconstructed standardized name aligned to backbone + original-cased author.

Orig.Genus Genus in Title Case.

Orig.Species Specific epithet in lowercase, or `NA` for genus-only (`sp./spp.`).

Author Recovered author string (original casing/punctuation) or `""`.

Orig.Infraspecies Infraspecific epithet in lowercase (ranked or implied), or `NA`.

Infra.Rank Infraspecific rank in lowercase (`subsp.`, `var.`, `subvar.`, `f.`, `subf.`), or `NA`.

Rank Numeric level: `1` genus-only, `2` genus+species, `3` includes infraspecific epithet.

has_cf,has_aff,is_sp,is_spp,had_hybrid,rank_late,rank_missing_infra,had_na_author,implied_infra Logical flags.

Examples

```
library(wcvpmatch)
classify_spnames(c("Opuntia sp.", "Rosa canina subsp. coriifolia (Fr.) Leffler"))
classify_spnames(c("Cydonia japonica tricolor")) # implied unranked infra epithet
```

fia — *Cleaned Master Tree Species List from FIA*

Description

A cleaned dataset containing tree species recorded by the Forest Inventory and Analysis (FIA) program of the U.S. Forest Service. This dataset is used in the examples and README of the `wcvpmatch` package. The data was downloaded in November 2022 from the official webpage of the Forest Inventory and Analysis National Program, available at the following link, and was originally used during the development of the `treemendous` package. For `wcvpmatch`, the variable names have been standardized to `Orig.Genus` and `Orig.Species`.

Usage

```
fia
```

Format

A data frame with 2169 rows and 2 variables:

Orig.Genus Genus name of the species binomial

Orig.Species Specific epithet of the species binomial

wcvp_distribution — *Retrieve Tabular WCVP Distribution by Taxonomic Rank*

Description

[Stable]

Usage

```
wcvp_distribution(
  taxon,
  taxon_rank = c("species", "genus", "family", "order", "higher"),
  native = TRUE,
  introduced = TRUE,
  extinct = TRUE,
  location_doubtful = TRUE,
  wcvp_names = NULL,
  wcvp_distributions = NULL,
  prefilter_genus = TRUE,
  fallback_to_genus = TRUE,
  summarise_by_input = FALSE,
  max_dist = NULL,
  method = "osa",
  output = c("standard", "full", "spatial", "summary")
)
```

Arguments

taxon	Character vector of taxa to query.
taxon_rank	Character scalar. One of "species", "genus", "family", "order", or "higher". The last two require corresponding order or higher columns in wcvp_names.
native	Logical. Include native occurrences? Defaults to TRUE.
introduced	Logical. Include introduced occurrences? Defaults to TRUE.
extinct	Logical. Include extinct occurrences? Defaults to TRUE.
location_doubtful	Logical. Include doubtful occurrences? Defaults to TRUE.
wcvp_names	Optional WCVP names table. If NULL, the function loads wcvpdata::wcvp_matching_names().
wcvp_distributions	Optional WCVP distribution table. If NULL, the function loads wcvpdata::wcvp_distribution().
prefilter_genus	Logical. Forwarded to wcvp_matching() for species-level queries. Ignored for all other taxonomic ranks.
fallback_to_genus	Logical. If TRUE and taxon_rank = "species", inputs without species-level distribution are retried at genus level.
summarise_by_input	Logical. If TRUE, return one row per input taxon with collapsed distribution fields. In this mode, area_codes, areas, regions, continents, and distribution are returned as character strings separated by " - " rather than list-columns.
max_dist	Maximum string distance. If NULL, species queries default to 2 and genus/family queries to 0.
method	String distance method passed to fozziejoin.
output	Output layout: "standard" (the default analytical taxon-area table), "full" (the complete audit table), "spatial" (the compact taxon-area table for a later spatial join), or "summary" (one row per submitted taxon). summarise_by_input = TRUE is retained as a backwards-compatible alias for output = "summary".

Details

Queries distribution records by matching a taxon name against the WCVP names table and then resolving the corresponding rows in the WCVP distribution table. The function is designed around wcvpdata::wcvp_matching_names() and wcvpdata::wcvp_distribution(), but custom tables with the same schema can also be supplied.

Matching is performed with fozziejoin, using compact lookup tables and length-based prefiltering to keep the candidate set small. Species queries are resolved in two stages: genus candidates are matched first, then species names are searched only within those candidate genera.

If species-level matches resolve to synonyms and the names table contains accepted_plant_name_id, distribution is recovered from the accepted taxon. For queries above species, accepted names are preferred to avoid double counting synonym records.

The result deliberately contains no geometry and does not require sf. area_code_l3 is the stable WGSrpd level-3 key intended for a later join to a user-supplied spatial object.

Default names and distribution tables are cached for the current R session. Exact species queries use a direct lookup; lowercase strings and species keys over the full backbone are only built when a fuzzy query needs them.

Value

A non-spatial tibble. The default `output = "standard"` returns one row per matched query-area combination with the submitted and resolved taxa, geographic hierarchy, and four occurrence flags (12 columns). `output = "full"` additionally returns matching provenance and identifiers. `output = "spatial"` returns only the taxon-area fields needed for a later spatial join. `output = "summary"` returns one row per input taxon with collapsed text fields such as `distribution`, `areas`, `area_codes`, `regions`, `continents`, and `n_areas`.

Examples

```
library(wcvpmatch)

wcvp_distribution("Opuntia ficus-indica", taxon_rank = "species")
wcvp_distribution("Opuntia", taxon_rank = "genus")
wcvp_distribution("Cactaceae", taxon_rank = "family")
# When `order` is present in a custom names table:
# \dontrun{wcvp_distribution("Caryophyllales", taxon_rank = "order", wcvp_names = custom_names)}
```

---

wcvp_matching *Match Scientific Names Against WCVP*

Description

[Stable]

Runs a matching pipeline with exact and partial matching for binomial and trinomial names, including infraspecific rank validation.

With the default `wcvpdata` backbone and `prefilter_genus = TRUE`, a matching-schema version 1 Arrow dataset is queried for candidate genera before records are loaded into R. Accepted-name metadata is retrieved by identifier, including accepted taxa in other genera. The genus catalog and most recently selected target are reused within the R session. Older data schemas and `prefilter_genus = FALSE` use the complete backbone. Prepared custom backbones are reused, but custom data are not loaded selectively.

Usage

```
wcvp_matching(
  df,
  target_df = NULL,
  prefilter_genus = TRUE,
  allow_duplicates = FALSE,
```

```
  max_dist = 1,
  method = "osa",
  add_name_distance = FALSE,
  name_distance_method = "osa",
  profile = FALSE,
  output_name_style = c("snake_case", "legacy"),
  output = c("standard", "full")
)
```

Arguments

df	Input tibble/data.frame with either Genus/Species or Orig.Genus/Orig.Species. For trinomials, include Infra.Rank and Infraspecies (or Orig.Infra.Rank/Orig.Infraspecies).
target_df	Optional custom target table. If NULL, data are read from the optional wcvpdata checklist when available; otherwise pass target_df explicitly.
prefilter_genus	Logical. If TRUE (default), retain exact and fuzzy candidate genera. For a version 1 default wcvpdata backbone, only these genera are loaded from Arrow. If FALSE, load the complete default backbone. A supplied target_df is filtered in memory.
allow_duplicates	Logical. If TRUE, duplicated taxon keys are deduplicated internally for matching and then expanded back to original rows. Output includes input_index for traceability to the original input.
max_dist	Maximum distance used in all fuzzy matching stages (genus, species, infraspecies).
method	A string indicating the fuzzy matching method (passed to fozziejoin). Supported methods: • "levenshtein": Levenshtein edit distance (default). • "osa": Optimal string alignment. • "damerau_levensthein" or "dl": Damerau-Levenshtein distance. • "hamming": Hamming distance (equal-length strings only). • "lcs": Longest common subsequence. • "qgram": Q-gram similarity (requires q). • "cosine": Cosine similarity (requires q). • "jaccard": Jaccard similarity (requires q). • "jaro": Jaro similarity. • "jaro_winkler" or "jw": Jaro-Winkler similarity. • "soundex": Soundex codes based on the National Archives standard.
add_name_distance	Logical. If TRUE, add matched_dist as pairwise distance between input name (Input.Name fallback Orig.Name) and matched_taxon_name.
name_distance_method	Method passed to stringdist::stringdist when add_name_distance = TRUE (for example "osa").
profile	Logical. If TRUE, attach a timing table in the "timings" attribute of the returned tibble, with elapsed seconds per pipeline stage.

output_name_style
: Naming style for output columns:
  - "snake_case" returns standardized lower snake_case names.
  - "legacy" keeps the historical mixed naming convention.

output
: Output layout. "standard" (the default) returns the parsed, matched, and accepted-name fields needed for routine reconciliation. "full" additionally returns internal matching-stage flags, fuzzy distances, and parsing diagnostics.

Value

A tibble with parsed input fields and matched/accepted taxonomic context. The standard output contains input_index, input and matched name components, authorship, matched and accepted IDs/names, status, match_ambiguity, and matched. Use output = "full" for matching diagnostics.

Examples

```
library(wcvpmatch)
# Match a single name
wcvp_matching(data.frame(Genus = "Opuntia", Species = "yanganucensis"))

# Match multiple names with snake_case output
names <- c("Aniba heterotepala", "Anthurium quipuscoae")
df <- classify_spnames(names)
wcvp_matching(df, output_name_style = "snake_case")

# Attach per-stage timings for profiling
out <- wcvp_matching(df, output_name_style = "snake_case", profile = TRUE)
attr(out, "timings")
```

---

wcvp_setup_info *Check Default Backbone Setup*

Description

[Stable]

Usage

```
wcvp_setup_info(inform = TRUE)
```

Arguments

inform
: Logical. If TRUE (default), print a short setup message.

Details

Reports whether the optional companion package `wcvpdata` is available for use as the default WCVP backbone and, if not, explains how to install it from `r-universe`.

Value

Invisibly returns a named list with setup status fields: `default_backbone_available`, `wcvpdata_installed`, `wcvpdata_has_backbone`, `wcvpdata_version`, `repository`, and `install_command`.

Examples

```
library(wcvpmatch)
wcvp_setup_info()
```

---

wcvp_synonyms *Retrieve Synonyms Resolved Through the WCVP Backbone*

Description

[Stable]

Usage

```
wcvp_synonyms(
  taxon,
  target_df = NULL,
  prefilter_genus = TRUE,
  max_dist = 2,
  method = "osa",
  include_accepted = FALSE,
  output = c("compact", "full")
)
```

Arguments

`taxon`	Character vector of species names to resolve.
`target_df`	Optional WCVP-like names table. If `NULL`, the optional `wcvpdata` checklist is used.
`prefilter_genus`	Logical. Passed to `wcvp_matching()`.
`max_dist`	Maximum string distance used while resolving `taxon`.
`method`	String distance method passed to `wcvp_matching()`.
`include_accepted`	Logical. Include the accepted name as a row with `name_role = "accepted"` before its synonyms? Defaults to `FALSE`.
`output`	Output layout. `"compact"` (the default) returns the accepted name and its synonym fields. `"full"` additionally returns matching diagnostics, WCVP identifiers, basionym ID, and retrieval status.

Details

Resolves submitted species names with `wcvp_matching()` and retrieves every WCVP record with `taxon_status = "Synonym"` that points to the resolved accepted name. This means that an accepted name, a synonym, or a minor spelling variant can all be used as the query.

Exact submitted names use a direct lookup before invoking the fuzzy matching pipeline. The accepted-name lookup and normalized default backbone are cached for the current R session and shared with the other package functions.

Value

A tibble with one row per input name and retrieved synonym. The default compact result contains `submitted_name`, `accepted_taxon_name`, `accepted_taxon_authors`, `name_role`, `synonym_name`, `synonym_authors`, and `homotypic_synonym`. Inputs without a match or without recorded synonyms are retained with missing synonym fields.

Examples

```
wcvp_synonyms("Nopalea cochenillifera")
```

Index

∗ **datasets**
fia, 4

classify_spnames, 2

fia, 4

wcvp_distribution, 4
wcvp_matching, 6
wcvp_matching(), *9*, *10*
wcvp_setup_info, 8
wcvp_synonyms, 9