Package ‘peruflorads43’

May 9, 2026

Type Package

Title Check Threatened Plant Species Status Against Peru's Supreme Decree 043-2006-AG

Version 0.2.3

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

Description Provides tools to match plant species names against the official threatened species list of Peru (Supreme Decree 043-2006-AG, 2006). Implements a hierarchical matching pipeline with exact, fuzzy, and suffix matching algorithms to handle naming variations and taxonomic changes. Supports both the original 2006 nomenclature and updated taxonomic names, allowing users to check protection status regardless of nomenclatural changes since the decree's publication. Threat categories follow International Union for Conservation of Nature standards (Critically Endangered, Endangered, Vulnerable, Near Threatened).

Encoding UTF-8

URL `https://github.com/PaulESantos/peruflorads43`, `https://paulesantos.github.io/peruflorads43/`

BugReports `https://github.com/PaulESantos/peruflorads43/issues`

Depends R (>= 4.1.0)

Imports assertthat (>= 0.2.1), dplyr (>= 1.1.0), fuzzyjoin (>= 0.1.6), memoise (>= 2.0.1), progress (>= 1.2.2), purrr (>= 1.0.0), readr (>= 2.1.0), stringr (>= 1.5.0), tibble (>= 3.1.0), tidyr (>= 1.3.0)

Suggests gt, scales, stringdist, ggplot2, forcats, covr, knitr, rmarkdown, testthat (>= 3.0.0), withr (>= 2.5.0)

VignetteBuilder knitr

Language en-US

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation no

Author Paul E. Santos Andrade [aut, cre] (ORCID: <https://orcid.org/0000-0002-6635-0375>)

Repository CRAN

Date/Publication 2026-02-27 08:30:07 UTC

Contents

`check_ds043` *Simplified wrapper for consolidated matching*

Description

Simplified interface for checking DS 043-2006-AG status with automatic consolidation of original and updated nomenclature.

Usage

```
check_ds043(splist, return_simple = FALSE)
```

Arguments

`splist`	Character vector of species names
`return_simple`	Logical. If TRUE, returns only "Protected" or "Not protected"

Value

Character vector with protection status

Examples

```
## Not run:
species <- c("Brassia ocanensis", "Persea americana")
check_ds043(species)

## End(Not run)
```

`comparison_table_ds043` *Create comparison table between original and updated results*

Description

Creates a side-by-side comparison table useful for understanding nomenclatural changes and their impact on DS 043-2006-AG status.

Usage

```
comparison_table_ds043(splist)
```

Arguments

`splist`	Character vector of species names

Value

Tibble with comparison

`get_ambiguous_matches` *Retrieve Ambiguous Match Information*

Description

Extracts information about ambiguous matches (multiple candidates with tied distances) from matching results. This is useful for quality control and manual curation of uncertain matches.

Usage

```
get_ambiguous_matches(
  match_result,
  type = c("genus", "species", "infraspecies", "all"),
  save_to_file = FALSE,
  output_dir = tempdir()
)
```

Arguments

`match_result`	A tibble returned by matching functions such as `matching_threatenedperu` or internal matching functions.
`type`	Character. Type of ambiguous matches to retrieve: • `"genus"` (default): Ambiguous genus-level matches • `"species"`: Ambiguous species-level matches

- `"infraspecies"`: Ambiguous infraspecies-level matches (includes level 2)
- `"all"`: All types of ambiguous matches

`save_to_file`	Logical. If TRUE, saves results to a CSV file. Default is FALSE (CRAN compliant - no automatic file writing).
`output_dir`	Character. Directory to save the file if save_to_file = TRUE. Defaults to `tempdir()` for safe file operations.

Details

During fuzzy matching, multiple candidates may have identical string distances, making the choice of match ambiguous. The matching algorithm automatically selects the first candidate, but this function allows you to:

- Review all ambiguous matches for quality control
- Export them for manual curation
- Make informed decisions about match quality

Value

A tibble with ambiguous match details, or NULL if no ambiguous matches exist. Columns depend on the match type but typically include original names, matched names, and distance metrics.

File Output

When `save_to_file = TRUE`, a timestamped CSV file is created:

- Filename format: "threatenedperu_ambiguous_[type]_[timestamp].csv"
- Location: `output_dir` (defaults to tempdir())
- Contains all ambiguous matches with metadata

`get_database_summary` *Get Database Summary Statistics*

Description

Provides summary statistics for the threatened species databases.

Usage

```
get_database_summary(type = c("both", "original", "updated"))
```

Arguments

`type`	Character string: "original", "updated", or "both" (default).

Value

A tibble with summary statistics.

Examples

```
# Get summary of both databases
summary_stats <- get_database_summary()
print(summary_stats)

# Get summary of just the original
summary_original <- get_database_summary("original")
print(summary_original)
```

---

get_threatened_database *Get Threatened Species Database*

Description

Retrieves the threatened plant species database for Peru. This function provides controlled access to the internal datasets used by the package.

Usage

```
get_threatened_database(type = c("original", "updated"))
```

Arguments

`type`	Character string specifying which database version to retrieve. Options are: • `"original"` (default): Original nomenclature from DS 043-2006-AG (2006) • `"updated"`: Updated nomenclature with current taxonomic consensus

Value

A tibble containing the threatened species database.

Database Structure

Original Database (`type = "original"`):

- ~777 species as listed in DS 043-2006-AG
- Supports quaternomial names (Rank 4)
- Includes both accepted names and synonyms
- Columns: scientific_name, genus, species, tag, infraspecies, tag_2, infraspecies_2, threat_category, accepted_name_author, taxonomic_status, accepted_name, family, protected_ds_043

Updated Database (`type = "updated"`):

- Updated nomenclature using WCVP and POWO
- Supports trinomial names (Rank 3 maximum)
- Only accepted names (synonyms resolved)
- Columns: scientific_name, genus, species, tag_acc, infraspecies, threat_category, accepted_name_author, taxonomic_status, accepted_name, family, protected_ds_043

Threat Categories

CR Critically Endangered

EN Endangered

VU Vulnerable

NT Near Threatened

Legal Context

Data based on Supreme Decree DS 043-2006-AG, Ministry of Agriculture, Peru (July 13, 2006), which establishes the official list of threatened wild flora species in Peru.

Note

This function is primarily for advanced users who need direct access to the database structure. For most use cases, use the higher-level functions: `is_threatened_peru` or `is_ds043_2006_ag`.

See Also

`is_threatened_peru` to check threat status of species `is_ds043_2006_ag` to check DS 043 protection status

Examples

```
# Get original database
db_original <- get_threatened_database(type = "original")
str(db_original)
nrow(db_original)

# Get updated database
db_updated <- get_threatened_database(type = "updated")
str(db_updated)

# Compare number of species
n_original <- nrow(db_original)
n_updated <- nrow(db_updated)
cat("Original:", n_original, "| Updated:", n_updated, "\n")

# Count by threat category
table(db_original$threat_category)

# Find critically endangered orchids
```

```
orchids <- db_original[db_original$family == "ORCHIDACEAE" &
                         db_original$threat_category == "CR", ]
head(orchids$scientific_name)
```

`is_ds043_2006_ag` *Matching for DS 043-2006-AG Species*

Description

Performs consolidated matching that searches species names in both the original DS 043-2006-AG list (2006 names) and the updated nomenclature database. This ensures that users with updated names can still identify if their species are protected under the DS 043-2006-AG, even if the nomenclature has updated.

Usage

```
is_ds043_2006_ag(splist, prioritize = "original", return_details = FALSE)
```

Arguments

`splist`	Character vector of species names to check
`prioritize`	Character. Which result to prioritize when both databases match: "original" (default) or "updated"
`return_details`	Logical. Return detailed matching information

Details

The function performs a two-stage search:

1. Searches in original DS 043-2006-AG (names as listed in 2006) 2. Searches in updated nomenclature database (current accepted names) 3. Consolidates results with clear indication of which database provided the match 4. Identifies if original names are now synonyms

This approach handles cases where: - User provides original name from 2006: Found in original database - User provides updated name: Found in updated database and linked to DS 043-2006-AG list - Name matches in both: Returns most relevant result based on priority - Original name is now a synonym: Indicated with "(synonym)" marker

Value

If return_details = FALSE: Character vector with consolidated threat status. If return_details = TRUE: Tibble with detailed reconciliation information.

Examples

```
## Not run:
# Species with nomenclatural changes
species <- c(
  "Haageocereus acranthus subsp. olowinskianus",  # Original name
  "Brassia ocanensis",                            # Updated name (was Ada)
  "Ida locusta",                                  # Updated name
  "Lycaste locusta",                              # Now a synonym
  "Persea americana"                              # Not threatened
)

# Get consolidated status
status <- is_ds043_2006_ag(species)

# Get detailed information
details <- is_ds043_2006_ag(species, return_details = TRUE)
View(details)

## End(Not run)
```

`is_threatened_peru` *Check if species are threatened listed in DS 043-2006-AG Peru*

Description

This function checks if a list of species names are threatened according to the Peruvian threatened species database. The function allows fuzzy matching for species names with a maximum distance threshold to handle potential typos or variations in species names.

Usage

```
is_threatened_peru(splist, source = "original", return_details = FALSE)
```

Arguments

`splist`	A character vector containing the list of species names to be checked for threatened status in Peru.
`source`	Character string specifying which database version to use. Options are: • `"original"` (default): Uses the original threatened species database • `"updated"`: Uses the updated database with synonyms
`return_details`	Logical. If TRUE, returns detailed matching results. If FALSE (default), returns only the threat status vector.

Value

If return_details = FALSE: A character vector indicating the threat status of each species ("Not threatened", "Threatened - CR", "Threatened - EN", "Threatened - VU", "Threatened - NT", or "Threatened - Unknown category").

If return_details = TRUE: A tibble with detailed matching results including matched names, threat categories, and matching process information.

Examples

```
# Example 1: Basic usage with valid species names
species_list <- c("Cattleya maxima", "Polylepis incana", "Fake species")

# Simple status check
threat_status <- tryCatch(
  is_threatened_peru(species_list),
  error = function(e) {
    message("Error in matching: ", e$message)
    rep("Error", length(species_list))
  }
)
print(threat_status)

# Example 2: Detailed results
detailed_results <- tryCatch(
  is_threatened_peru(species_list, return_details = TRUE),
  error = function(e) {
    message("Error in detailed matching: ", e$message)
    NULL
  }
)
if (!is.null(detailed_results)) {
  print(detailed_results)
}

# Example 3: Handling NA values gracefully
species_with_na <- c("Cattleya maxima", NA, "Polylepis incana")
status_with_na <- is_threatened_peru(species_with_na)
print(status_with_na)

# Example 4: Empty input handling
empty_result <- is_threatened_peru(character(0))
print(empty_result)  # Should return character(0)

# Example 5: Using updated database
updated_results <- tryCatch(
  is_threatened_peru(species_list, source = "updated"),
  error = function(e) {
    message("Error with updated database: ", e$message)
    rep("Error", length(species_list))
  }
)
```

```
print(updated_results)
```

`matching_threatenedperu`

Match Species Names to Threatened Plant List of Peru

Description

This function matches given species names against the internal database of threatened plant species in Peru. It uses a hierarchical matching strategy that includes direct matching, genus-level matching, fuzzy matching, and suffix matching to maximize successful matches while maintaining accuracy.

Usage

```
matching_threatenedperu(
  splist,
  source = c("original", "updated"),
  quiet = TRUE
)
```

Arguments

`splist`	A character vector containing the species names to be matched. Can include duplicate names - results will be expanded to match the input.
`source`	Character string specifying which database version to use. Options are: • `"original"` (default): Uses the original threatened species database with support for Rank 4 (quaternomial names) • `"updated"`: Uses the updated database with current nomenclature, supporting up to Rank 3 (trinomial names)
`quiet`	Logical, default TRUE. If FALSE, prints informative messages.

Details

Duplicate Handling: When the input contains duplicate names, the function automatically:

- Detects duplicates and creates a tracking column (sorters)
- Processes only unique names (efficient matching)
- Expands results to restore all original positions
- Preserves original input order via sorter column

The duplicate handling uses a ‘sorters‘ column that concatenates all original sorter values for duplicate names (e.g., "1 - 3" for a name appearing at positions 1 and 3), enabling accurate result expansion.

Matching Strategy: 1. Direct exact matching 2. Genus-level matching (exact and fuzzy) 3. Species-level matching within genus 4. Infraspecies-level matching (up to 2 levels for original database)

Rank Validation: The algorithm implements strict rank validation to prevent false positives.

Value

A tibble with detailed matching results including:

sorter	Integer. Original position in input vector
Orig.Name	Character. Original input name (standardized)
Matched.Name	Character. Matched name from database or "—"
Threat.Status	Character. IUCN threat category or "Not threatened"
Rank	Integer. Input taxonomic rank (1-4)
Matched.Rank	Integer. Matched taxonomic rank
Comp.Rank	Logical. Whether ranks match exactly
Match.Level	Character. Description of match quality
matched	Logical. Whether a match was found

See Also

`is_threatened_peru` for a simplified interface `get_ambiguous_matches` to retrieve ambiguous match details `get_threatened_database` to access the raw databases

Examples

```
## Not run:
# Basic usage
species_list <- c("Cattleya maxima", "Polylepis incana")
results <- matching_threatenedperu(species_list, source = "original")

# With duplicates
species_dup <- c("Cattleya maxima", "Polylepis incana", "Cattleya maxima")
results_dup <- matching_threatenedperu(species_dup)
nrow(results_dup) == 3  # TRUE - preserves duplicates

# Access metadata
attr(results, "match_rate")

# Check for ambiguous matches
get_ambiguous_matches(results, type = "infraspecies")

## End(Not run)
```

Index