Package ‘perumammals’

January 6, 2026

Title Taxonomic Backbone and Name Validation Tools for Mammals of Peru

Version 0.0.0.1

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

Description Provides a curated taxonomic backbone of mammal species from Peru based on Pacheco et al. (2021) ``Lista actualizada de la diversidad de los mamíferos del Perú y una propuesta para su actualización" <doi:10.15381/rpb.v28i4.21019>. The package includes standardized species data, occurrence by ecoregions, endemism status, and tools for validating and matching scientific names through exact and fuzzy procedures. It is designed as a lightweight and dependable reference for ecological, environmental, biogeographic, and conservation workflows that require reliable species information for Peruvian mammals.

License MIT + file LICENSE

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

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

Depends R (>= 4.1)

Encoding UTF-8

LazyData true

Config/testthat/edition 3

RoxygenNote 7.3.3

Imports assertthat, cli, dplyr, fuzzyjoin, progress, purrr, readr, stringr, tibble, memoise

Suggests ggplot2, knitr, rmarkdown, testthat (>= 3.0.0), tidyr, ggtext

VignetteBuilder knitr

NeedsCompilation no

Author Paul E. Santos Andrade [aut, cre] (ORCID: <https://orcid.org/0000-0002-6635-0375>), Fiorella N. Gonzales Guillen [ctb] (ORCID: <https://orcid.org/0000-0001-5240-2464>)

Repository CRAN

Date/Publication 2026-01-06 11:30:02 UTC

Contents

---

found_in_peru *Quick check: Is species found in Peru?*

Description

Simplified boolean check for species presence in Peru mammals database. Useful for filtering and logical operations.

Usage

```
found_in_peru(splist, exact_only = FALSE)
```

Arguments

splist	Character vector of species names
exact_only	Logical. If TRUE, only exact matches return TRUE (default: FALSE)

Value

Logical vector (TRUE = found, FALSE = not found)

Examples

```
species <- c("Panthera onca", "Tremarctos orrnatus",
             "Tremarctos orrnatos", "Felis catus")

# Check presence (includes fuzzy matches)
found_in_peru(species)

tibble::tibble(splist = species) |>
 dplyr::mutate(endemic = found_in_peru(splist))
```

`get_ambiguous_matches` *Retrieve Ambiguous Match Information for Peru Mammals*

Description

Extracts information about ambiguous matches (multiple candidates with tied distances) from matching results. Useful for quality control and manual curation. Adapted for peru_mammals (genus and species only).

Usage

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

Arguments

`match_result` A tibble returned by matching functions.

`type` Character. Type of ambiguous matches to retrieve:

- `"genus"` (default): Ambiguous genus-level matches
- `"species"`: Ambiguous species-level matches
- `"all"`: Both types

`save_to_file` Logical. If TRUE, saves results to CSV. Default is FALSE (CRAN compliant).

`output_dir` Character. Directory to save file if save_to_file = TRUE. Defaults to `tempdir()`.

Details

During fuzzy matching, multiple candidates may have identical string distances. The matching algorithm automatically selects the first candidate, but this function allows you to review all alternatives for quality control.

Value

A tibble with ambiguous match details, or NULL if none exist. Includes original names, matched names, distances, and database metadata.

`get_common_names_peru` *Get taxonomic and common name information for Peru mammals*

Description

Returns taxonomic classification and common names for species validated against the Peru mammals database.

Usage

```
get_common_names_peru(splist, return_details = FALSE)
```

Arguments

`splist`	Character vector of species names
`return_details`	Logical. If TRUE, includes full taxonomic information (default: FALSE)

Value

If return_details = FALSE: Character vector with common names If return_details = TRUE: Tibble with taxonomic and common name information

Examples

```
species <- c("Panthera onca", "Tremarctos ornatus",
             "Puma concolor", "Myotis bakeri")

# Get common names
# Vector
get_common_names_peru(species)
# tibble
tibble::tibble(splist = species) |>
 dplyr::mutate(endemic = get_common_names_peru(splist))

# Get full taxonomic information
taxonomy <- get_common_names_peru(species, return_details = TRUE)
taxonomy
```

is_endemic_peru *Check if species are endemic to Peru*

Description

Simplified wrapper specifically for checking endemism status of mammals in Peru. Only evaluates species that are confirmed to occur in Peru.

Usage

```
is_endemic_peru(splist, return_logical = FALSE, filter_exact = FALSE)
```

Arguments

`splist`	Character vector of species names
`return_logical`	Logical. If TRUE, returns logical vector (TRUE/FALSE/NA). If FALSE, returns descriptive character vector (default: FALSE)
`filter_exact`	Logical. If TRUE, only considers exact matches (default: FALSE)

Value

If return_logical = FALSE: Character vector with endemism status If return_logical = TRUE: Logical vector (TRUE = endemic, FALSE = not endemic, NA = not found or endemism unknown)

Examples

```
species <- c("Panthera onca",
             "Atelocynus microtis",
             "Felis catus",
             "Myotis bakeri")

is_endemic_peru(species)
# Descriptive output
tibble::tibble(splist = species) |>
  dplyr::mutate(endemic = is_endemic_peru(splist))
```

is_peru_mammal *Check if species are Peru mammals*

Description

Main wrapper function that validates species names against the Peru mammals database with various output options for match quality, endemism status, and detailed information.

Usage

```
is_peru_mammal(
  splist,
  return_details = FALSE,
  match_type = "status",
  filter_exact = FALSE
)
```

Arguments

`splist` Character vector of species names to check

`return_details` Logical. If TRUE, returns full validation tibble. If FALSE, returns simplified status vector (default: FALSE)

`match_type` Character. Type of information to return when return_details = FALSE:

- "status": Returns "Found" or "Not found" (default)
- "match_quality": Returns match quality ("Exact", "Fuzzy", or "Not found")
- "endemic": Returns endemism status ("Endemic", "Not endemic", or "Not found")

`filter_exact` Logical. If TRUE, only returns exact matches (genus_dist = 0 AND species_dist = 0). Fuzzy matches are treated as "Not found" (default: FALSE)

Details

This function wraps `validate_peru_mammals()` to provide flexible output formats for different use cases:

- Basic presence/absence checking
- Match quality assessment (exact vs fuzzy)
- Endemism status queries

The function handles taxonomic matching with fuzzy string matching to accommodate minor spelling variations while maintaining data quality.

When filter_exact = TRUE, only matches with zero edit distance in both genus and species names are considered valid matches. All fields related to fuzzy matches are set to NA or "—" to maintain consistency.

Value

If return_details = FALSE: Character vector with requested information. If return_details = TRUE: Tibble with complete validation information.

Examples

```
species <- c(
  "Panthera onca",       # Exact match
  "Pantera onca",        # Fuzzy match (genus misspelled)
  "Tremarctos orrnatus", # Fuzzy match (species misspelled)
  "Felis domesticus",     # Not in Peru
  "Myotis bakeri"
)

# Check if species are found (includes fuzzy matches)
is_peru_mammal(species)

# Check with exact matches only
is_peru_mammal(species, filter_exact = TRUE)

# Check match quality
is_peru_mammal(species, match_type = "match_quality")

# Check endemism
is_peru_mammal(species, match_type = "endemic")

# Get detailed information
is_peru_mammal(species, return_details = TRUE)

# Get detailed information with exact matches only
is_peru_mammal(species, return_details = TRUE, filter_exact = TRUE)
```

---

match_quality_peru *Get match quality for Peru mammal names*

Description

Returns the quality of taxonomic name matching (exact vs fuzzy) for species validated against the Peru mammals database.

Usage

```
match_quality_peru(splist, return_details = FALSE)
```

Arguments

splist	Character vector of species names
return_details	Logical. If TRUE, includes distance metrics and matching information (default: FALSE)

Details

Match quality categories:

- "Exact": Perfect match with no spelling differences (genus_dist = 0, species_dist = 0)
- "Fuzzy": Match found with minor spelling variations (genus_dist > 0 or species_dist > 0)
- "Not found": No match in database

The function uses string distance metrics to quantify matching quality:

- genus_dist: Edit distance for genus name
- species_dist: Edit distance for species epithet

Value

If return_details = FALSE: Character vector with match quality If return_details = TRUE: Tibble with detailed matching information

Examples

```
species <- c(
  "Panthera onca",      # Exact
  "Tremarctos orrnatus", # Fuzzy (spelling error)
  "Felis domesticus",   # Not found
  "Myotis bakeri"
)

# Simple quality check
match_quality_peru(species)

# Detailed information with edit distances
details <- match_quality_peru(species, return_details = TRUE)
details
```

`peru_mammals` *Mammal species of Peru based on Pacheco et al. (2021)*

Description

A backbone of the terrestrial and marine mammal species known for Peru, compiled from Pacheco et al. (2021) "Lista actualizada de la diversidad de los mamíferos del Perú y una propuesta para su actualización".

Usage

```
data("peru_mammals")
```

Format

A tibble with 573 rows and 12 variables:

pm_id Character. Internal stable identifier for the species, combining the original numeric id and an abbreviation of the genus. Intended for internal linking between tables.

order Character. Taxonomic order (e.g. *Didelphimorphia*, *Rodentia*, *Chiroptera*).

family Character. Taxonomic family.

genus Character. Genus name.

species Character. Specific epithet.

scientific_name Character. Binomial scientific name (*Genus species*), without authorship. This is the main field used for name validation.

scientific_name_full Character. Full scientific name including authorship and year, as provided in the original annex.

author Character. Authorship and year of the species name.

common_name Character. Common name in Spanish, when available.

endemic Logical. `TRUE` if the species is considered endemic to Peru in Pacheco et al. (2021), `FALSE` otherwise.

ecoregions Character. Comma-separated codes of Peruvian ecoregions where the species occurs, using the abbreviations defined by Pacheco et al. (2021) (e.g. `"YUN, SB, SP"`). See `peru_mammals_ecoregions_meta` for code definitions.

reference Character. Bibliographic notes or specific references supporting the presence or taxonomy of the species.

Details

Each row corresponds to a single species as listed in the original annex of the paper. This dataset is the main taxonomic backbone used by the **perumammals** package.

Source

Pacheco, V., Cadenillas, R., Zeballos, H., Hurtado, C. M., Ruelas, D., & Pari, A. (2021). Lista actualizada de la diversidad de los mamíferos del Perú y una propuesta para su actualización.

`peru_mammals_backbone` *Summary information on the* **perumammals** *taxonomic backbone*

Description

A one-row tibble with metadata about the taxonomic backbone used in **perumammals**, including its bibliographic source, year, number of species and the date when the internal data objects were created.

Usage

```
data("peru_mammals_backbone")
```

Format

A tibble with 1 row and 4 variables:

source Character. Short bibliographic reference to the backbone source (Pacheco et al. 2021).

source_year Integer. Publication year of the backbone source (2021).

n_species Integer. Number of species included in the backbone (as rows in `peru_mammals`).

created_at Date. Date when the backbone data objects were generated (in the package build process).

Details

This object is intended for internal bookkeeping and for functions that report the origin and version of the backbone.

See Also

`peru_mammals`

---

peru_mammals_ecoregions *Mammal species by Peruvian ecoregion*

Description

A long-format table linking each mammal species to the Peruvian ecoregions where it occurs, based on Pacheco et al. (2021).

Usage

```
data("peru_mammals_ecoregions")
```

Format

A tibble with one row per species–ecoregion combination and 3 variables:

pm_id Character. Internal species identifier, matching `peru_mammals`.

scientific_name Character. Binomial scientific name (*Genus species*).

ecoregion_code Character. Abbreviation of the ecoregion where the species occurs (e.g. `"YUN"`, `"SB"`, `"COS"`). See `peru_mammals_ecoregions_meta` for code definitions.

Details

Each row corresponds to a single combination of species and ecoregion. This dataset is derived from the `ecoregions` field of `peru_mammals`.

Source

Pacheco et al. (2021).

See Also

`peru_mammals`, `peru_mammals_ecoregions_meta`

`peru_mammals_ecoregions_meta`

Metadata for Peruvian mammal ecoregions

Description

Definitions of the ecoregion codes used in `peru_mammals` and `peru_mammals_ecoregions`. The codes follow the abbreviations used by Pacheco et al. (2021), based on Peruvian ecoregion schemes.

Usage

```
data("peru_mammals_ecoregions_meta")
```

Format

A tibble with one row per ecoregion code and 2 variables:

ecoregion_code Character. Ecoregion abbreviation. The codes used in the dataset are:

- `"OCE"` – Oceánica
- `"BPP"` – Bosque Pluvial del Pacífico
- `"BSE"` – Bosque Seco Ecuatorial
- `"COS"` – Costa
- `"VOC"` – Vertiente Occidental
- `"PAR"` – Páramo
- `"PUN"` – Puna
- `"YUN"` – Yungas
- `"SB"` – Selva Baja
- `"SP"` – Sabana de Palmera

ecoregion_label Character. Human-readable label/description of the ecoregion in Spanish.

Source

Pacheco et al. (2021).

See Also

`peru_mammals`, `peru_mammals_ecoregions`

`pm_backbone_info` *Display taxonomic backbone metadata for Peruvian mammals*

Description

Displays summary information about the taxonomic backbone used in **perumammals**. The backbone is based on the taxonomic checklist published by Pacheco et al. (2021), which was digitised from the original PDF publication into a structured tibble format.

Usage

```
pm_backbone_info()
```

Value

Invisibly returns a tibble with one row containing the backbone metadata. The same structure as `peru_mammals_backbone`. Called primarily for its side effect of printing the summary information.

References

Pacheco Torres, V. R., Diaz, S., Graham Angeles, L. A., Flores-Quispe, M., Calizaya-Mamani, G., Ruelas, D., & Sánchez-Vendizú, P. (2021). Lista actualizada de la diversidad de los mamíferos del Perú y una propuesta para su actualización. *Revista Peruana De Biología*, **28**(4), e21019. doi:10.15381/rpb.v28i4.21019

See Also

`peru_mammals_backbone` for the complete backbone data.

Examples

```
# Display backbone information
pm_backbone_info()

# Access the data invisibly returned
backbone_data <- pm_backbone_info()
backbone_data$n_species
```

pm_by_ecoregion *List species by ecoregion*

Description

Convenience wrapper to list species occurring in one or more Peruvian ecoregions. This function uses pm_species() internally and therefore supports the same taxonomic and endemism filters.

Usage

```
pm_by_ecoregion(
  ecoregion,
  order = NULL,
  family = NULL,
  genus = NULL,
  endemic = NULL
)
```

Arguments

ecoregion	Character vector with one or more ecoregion codes (e.g. "YUN", "SB", "COS"). At least one code must be provided. Invalid codes will generate a warning.
order	Optional character vector with one or more taxonomic orders to keep. If NULL (default), no filter is applied by order.
family	Optional character vector with one or more families to keep. If NULL (default), no filter is applied by family.
genus	Optional character vector with one or more genera to keep. If NULL (default), no filter is applied by genus.
endemic	Optional logical. If TRUE, only endemic species are returned; if FALSE, only non-endemic species are returned; if NULL (default), no filter is applied by endemism.

Value

A tibble with a subset of rows from peru_mammals corresponding to species present in at least one of the requested ecoregions. Returns an empty tibble if no species match the criteria.

See Also

pm_list_ecoregions() to see available ecoregion codes, pm_species() for the underlying function.

Examples

```
# All species in Yungas
pm_by_ecoregion("YUN")

# Endemic species in Selva Baja (SB)
```

```
pm_by_ecoregion("SB", endemic = TRUE)

# Rodents in Costa and Vertiente Occidental
pm_by_ecoregion(c("COS", "VOC"), order = "Rodentia")

# Bats in multiple ecoregions
pm_by_ecoregion(c("YUN", "SB"), order = "Chiroptera")
pm_by_ecoregion(c("YUN", "SB"), order = "Chiroptera",
endemic =  TRUE)
```

`pm_ecoregion_summary` *Summary of species richness by ecoregion*

Description

Computes a summary of species richness and endemism for each ecoregion in the Peruvian mammal backbone.

Usage

```
pm_ecoregion_summary(sort_by = c("code", "species", "endemic", "label"))
```

Arguments

`sort_by` Character string indicating how to sort the results. Options are:

- `"code"` (default) – sort alphabetically by ecoregion code.
- `"species"` – sort by number of species (descending).
- `"endemic"` – sort by number of endemic species (descending).
- `"label"` – sort alphabetically by ecoregion label.

Details

The summary is based on the long-format table `peru_mammals_ecoregions` and joins metadata from `peru_mammals_ecoregions_meta` and endemism information from `peru_mammals`.

Value

A tibble with one row per ecoregion and the following columns:

- `ecoregion_code` – ecoregion abbreviation.
- `ecoregion_label` – ecoregion description in Spanish.
- `n_species` – total number of species recorded in the ecoregion.
- `n_endemic` – number of endemic species recorded in the ecoregion.
- `pct_endemic` – percentage of endemic species in the ecoregion.

See Also

pm_list_ecoregions() for ecoregion metadata, pm_by_ecoregion() to list species by ecoregion.

Examples

```
# Get summary for all ecoregions (sorted by code)
pm_ecoregion_summary()

# Sort by species richness
pm_ecoregion_summary(sort_by = "species")

# Sort by number of endemic species
pm_ecoregion_summary(sort_by = "endemic")

# Find ecoregion with highest species richness
eco_summary <- pm_ecoregion_summary(sort_by = "species")
eco_summary[1, ]

# Ecoregions with more than 100 species
eco_summary <- pm_ecoregion_summary()
subset(eco_summary, n_species > 100)

# Compare richness between lowland and highland ecoregions
eco_summary <- pm_ecoregion_summary(sort_by = "species")
lowland <- eco_summary[eco_summary$ecoregion_code %in% c("SB", "SP"), ]
highland <- eco_summary[eco_summary$ecoregion_code %in% c("PUN", "PAR"), ]
```

pm_endemics *List endemic mammal species of Peru*

Description

Returns endemic species from the Peruvian mammal backbone, with optional filters by order, family and/or ecoregion.

Usage

```
pm_endemics(order = NULL, family = NULL, genus = NULL, ecoregion = NULL)
```

Arguments

order	Optional character vector with one or more taxonomic orders to keep. If NULL (default), no filter is applied by order.
family	Optional character vector with one or more families to keep. If NULL (default), no filter is applied by family.
genus	Optional character vector with one or more genera to keep. If NULL (default), no filter is applied by genus.

`ecoregion` Optional character vector with one or more ecoregion codes (e.g. "YUN", "SB", "COS"). If supplied, only species occurring in at least one of the given ecoregions are returned.

Details

This is a convenience wrapper around `pm_species()` with `endemic = TRUE`.

Value

A tibble with endemic species (subset of `peru_mammals`).

Examples

```
# All endemic species
pm_endemics()

# Endemic rodents
pm_endemics(order = "Rodentia")

# Endemic species in Yungas (YUN)
pm_endemics(ecoregion = "YUN")
```

`pm_list_ecoregions` *Display ecoregion metadata for Peruvian mammals*

Description

Displays summary information about the ecoregions used in the Peruvian mammal backbone. Ecoregions follow the Brack-Egg (1986) classification system used in Peruvian biogeography to describe the distribution of mammal species across different ecological regions.

Usage

```
pm_list_ecoregions(include_endemic = FALSE)
```

Arguments

`include_endemic` Logical. If `TRUE`, includes columns showing the number and percentage of endemic species per ecoregion. Default is `FALSE`.

Details

The ecoregion classification follows Brack-Egg (1986), a widely-used biogeographic framework for Peru that recognizes 10 distinct ecological regions based on climate, vegetation, and elevation. This classification is used in Pacheco et al. (2021) to document the distribution patterns of Peruvian mammals.

The function prints a formatted summary to the console and invisibly returns the complete data for further analysis.

Value

A tibble with one row per ecoregion, arranged in descending order by species richness, with the following columns:

ecoregion_code Abbreviated ecoregion code (e.g., "SB", "YUN")

ecoregion_label Full ecoregion name in Spanish

n_species Total number of mammal species recorded in the ecoregion

pct_species Percentage of Peru's total mammal diversity (0-100)

n_endemic (Only if `include_endemic = TRUE`) Number of endemic species in the ecoregion

pct_endemic (Only if `include_endemic = TRUE`) Percentage of endemic species relative to total species in the ecoregion (0-100)

References

Brack-Egg, A. (1986). Ecología de un país complejo. In J. Mejía Baca (Ed.), Gran Geografía del Perú: Naturaleza y Hombre (Vol. 2, pp. 175-319). Barcelona: Manfer-Mejía Baca.

See Also

`peru_mammals_ecoregions_meta` for the complete ecoregion metadata, `peru_mammals_ecoregions` for species-ecoregion associations, `pm_by_ecoregion()` to filter species by ecoregion, `pm_ecoregion_summary()` for species richness summaries by ecoregion.

Examples

```
# Display ecoregion information
pm_list_ecoregions()

# Include endemic species information
 pm_list_ecoregions(include_endemic = TRUE)

# Access the data for further analysis
ecoregion_data <- pm_list_ecoregions()

# Ecoregions with highest species richness
ecoregion_data
```

`pm_list_endemic` *List endemic mammal species by taxonomic order*

Description

Summarises the diversity of endemic mammal species in Peru, grouped by taxonomic order. Provides counts of families, genera, and species that are endemic to Peru within each order. Optionally includes endemism rates relative to total species richness.

Usage

```
pm_list_endemic(include_rate = FALSE)
```

Arguments

`include_rate`	Logical. If `TRUE`, includes additional columns showing total species richness and endemism rate for each order. Default is `FALSE`.

Details

This function focuses exclusively on species that are endemic to Peru (i.e., species found nowhere else in the world). Orders without any endemic species are not included in the output.

When `include_rate = FALSE` (default), results are sorted by the number of endemic species in descending order, highlighting which orders have the highest endemic diversity.

When `include_rate = TRUE`, results are sorted by total species richness in descending order, and include endemism rates to show what proportion of each order's diversity is endemic to Peru. A summary row labeled "Total" is appended to show overall statistics.

Value

A tibble with one row per order containing endemic species, arranged in descending order by number of endemic species, with the following columns:

order Taxonomic order

n_families Number of families with endemic species in the order

n_genera Number of genera with endemic species in the order

n_endemic Number of endemic species in the order

n_species (Only if `include_rate = TRUE`) Total number of species in the order

endemic_rate (Only if `include_rate = TRUE`) Proportion of endemic species (0-1)

endemic_pct (Only if `include_rate = TRUE`) Percentage of endemic species (0-100)

Examples

```
# Summary of endemic species by order
pm_list_endemic()

# Include endemism rates
pm_list_endemic(include_rate = TRUE)
```

`pm_list_families` *List taxonomic families in the Peruvian mammal backbone*

Description

Summarises the number of genera, species and endemic species per family. Optionally filters the output to one or more taxonomic orders.

Usage

```
pm_list_families(order = NULL)
```

Arguments

`order`	Optional character vector specifying one or more taxonomic orders to include. If `NULL` (default), all orders are included. Order names are case-sensitive (e.g., "Rodentia", "Chiroptera").

Value

A tibble with one row per family, arranged by order and family name, with the following columns:

order Taxonomic order

family Family name

n_genera Number of genera in the family

n_species Number of species in the family

n_endemic Number of endemic species to Peru in the family

Examples

```
# All families
pm_list_families()

# Only families within Rodentia
pm_list_families(order = "Rodentia")

# Multiple orders
pm_list_families(order = c("Rodentia", "Chiroptera"))
```

`pm_list_genera` *List genera in the Peruvian mammal backbone*

Description

Summarises the number of species and endemic species per genus. Optionally restricts the output to one or more orders and/or families. Genera with missing values are excluded from the results.

Usage

```
pm_list_genera(order = NULL, family = NULL)
```

Arguments

`order` Optional character vector with one or more taxonomic orders to keep. If `NULL` (default), no filter is applied by order. Invalid order names will generate a warning.

`family` Optional character vector with one or more families to keep. If `NULL` (default), no filter is applied by family. Invalid family names will generate a warning.

Details

The function validates input parameters and warns if invalid order or family names are provided. It also warns if the filters result in an empty dataset.

Value

A tibble with one row per genus and the following columns:

- `order` – taxonomic order.
- `family` – family name.
- `genus` – genus name.
- `n_species` – number of species in the genus.
- `n_endemic` – number of endemic species in the genus.

Returns an empty tibble with the same structure if no records match the specified filters.

Examples

```
# All genera
pm_list_genera()

# Genera within Chiroptera (bats)
pm_list_genera(order = "Chiroptera")

# Multiple orders
pm_list_genera(order = c("Didelphimorphia", "Chiroptera"))
```

```
# Genera within a specific family
bat_genera <- pm_list_genera(family = "Phyllostomidae")

# Count total endemic species in a family
sum(bat_genera$n_endemic)

# Combination of filters
pm_list_genera(order = "Chiroptera", family = "Phyllostomidae")
```

pm_list_orders *List taxonomic orders in the Peruvian mammal backbone*

Description

Summarises the number of families, genera, species and endemic species per order in `peru_mammals`.

Usage

```
pm_list_orders()
```

Value

A tibble with one row per order and the following columns:

- `order` – taxonomic order.
- `n_families` – number of families in the order.
- `n_genera` – number of genera in the order.
- `n_species` – number of species in the order.
- `n_endemic` – number of endemic species in the order.

Examples

```
pm_list_orders()
```

pm_species *Filter mammal species from the Peruvian backbone*

Description

Convenience wrapper around peru_mammals to subset species by taxonomic group, endemism and/or ecoregion.

Usage

```
pm_species(
  order = NULL,
  family = NULL,
  genus = NULL,
  endemic = NULL,
  ecoregion = NULL
)
```

Arguments

order	Optional character vector with one or more taxonomic orders to keep. If NULL (default), no filter is applied by order.
family	Optional character vector with one or more families to keep. If NULL (default), no filter is applied by family.
genus	Optional character vector with one or more genera to keep. If NULL (default), no filter is applied by genus.
endemic	Optional logical. If TRUE, only endemic species are returned; if FALSE, only non-endemic species are returned; if NULL (default), no filter is applied by endemism.
ecoregion	Optional character vector with one or more ecoregion codes (e.g. "YUN", "SB", "COS"). If supplied, only species occurring in at least one of the given ecoregions are returned.

Value

A tibble with a subset of rows from peru_mammals.

Examples

```
# All species
pm_species()

# Only Rodentia
pm_species(order = "Rodentia")

# Endemic bats (Chiroptera)
pm_species(order = "Chiroptera", endemic = TRUE)
```

```
# Species present in Yungas (YUN) and Selva Baja (SB)
pm_species(ecoregion = c("YUN", "SB"))
```

validate_peru_mammals *Match Species Names Against Peru Mammals Database*

Description

Matches given species names against the official list of mammal species of Peru (Pacheco et al. 2021). Uses a hierarchical matching strategy that includes direct matching, genus-level matching, and fuzzy matching to maximize successful matches while maintaining accuracy.

Peru Mammals Database:

- 575 mammal species
- Binomial nomenclature only (no infraspecific taxa)
- Includes 6 undescribed species ("sp." cases)
- Fields: genus, species, scientific_name, common_name, family, order, endemic

Usage

```
validate_peru_mammals(splist, quiet = TRUE)
```

Arguments

`splist`	A character vector containing the species names to be matched. Names can be in any format (uppercase, lowercase, with underscores, etc.). Duplicate names are preserved in the output.
`quiet`	Logical, default TRUE. If FALSE, prints informative messages about the matching progress.

Details

Matching Strategy: The function implements a hierarchical matching pipeline:

1. **Node 1 - Direct Match:** Exact matching of binomial names (genus + species)
2. **Node 2 - Genus Match:** Exact matching at genus level
3. **Node 3 - Fuzzy Genus:** Fuzzy matching for genus with typos (max distance = 1)
4. **Node 4 - Fuzzy Species:** Fuzzy matching for species within matched genus

Special Cases:

- Handles "sp." cases: "Akodon sp. Ancash", "Oligoryzomys sp. B", etc.
- Case-insensitive matching
- Removes common qualifiers (CF., AFF.)

- Standardizes spacing and formatting

Rank System:

- **Rank 1:** Genus level only (e.g., "Panthera")
- **Rank 2:** Binomial (genus + species, e.g., "Panthera onca")

Ambiguous Matches: When multiple candidates have identical fuzzy match scores, a warning is issued and the first match is selected. Use `get_ambiguous_matches()` to examine these cases.

Input Requirements:

Species names must be provided as binomials (Genus species) WITHOUT:

- Author information: Panthera onca Linnaeus"
- Infraspecific taxa: "Panthera onca onca"
- Parenthetical authors: "Panthera onca (Linnaeus, 1758)"

Valid formats:

- Standard binomial: "Panthera onca"
- Undescribed species: "Akodon sp. Ancash"
- Case-insensitive: "PANTHERA ONCA" or "panthera onca"

Names with 3+ elements will be automatically rejected with a warning.

Value

A tibble with the following columns:

sorter Integer. Original position in input vector

Orig.Name Character. Original input name (standardized)

Matched.Name Character. Matched name from database or "—"

Match.Level Character. Quality of match ("Exact rank", "No match", etc.)

matched Logical. Whether a match was found

Rank Integer. Input taxonomic rank (1 or 2)

Matched.Rank Integer. Matched taxonomic rank (1 or 2)

Comp.Rank Logical. Whether ranks match exactly

valid_rank Logical. Whether match is valid at correct rank

Orig.Genus Character. Input genus (uppercase)

Orig.Species Character. Input species (uppercase)

Author Character. Taxonomic authority if provided

Matched.Genus Character. Matched genus (uppercase)

Matched.Species Character. Matched species (uppercase)

genus_dist Integer. Edit distance for genus (0=exact, >0=fuzzy, NA=no match)

species_dist Integer. Edit distance for species (0=exact, >0=fuzzy, NA=no match or genus-only)

scientific_name Character. Scientific name from peru_mammals

common_name Character. Common name in Spanish

family Character. Family

order Character. Order

endemic Logical. Endemic to Peru?

Attributes: The output includes metadata accessible via `attr()`:

- `target_database`: "peru_mammals"
- `matching_date`: Date of matching
- `n_input`: Number of input names
- `n_matched`: Number of successful matches
- `match_rate`: Percentage of successful matches
- `n_fuzzy_genus`: Number of fuzzy genus matches
- `n_fuzzy_species`: Number of fuzzy species matches
- `ambiguous_genera`: Ambiguous genus matches (if any)
- `ambiguous_species`: Ambiguous species matches (if any)

See Also

`get_ambiguous_matches` to retrieve ambiguous match details

Examples

```
# Basic usage
species_list <- c("Panthera onca", "Tremarctos ornatus", "Puma concolor")
results <- validate_peru_mammals(species_list)

# Check results
table(results$matched)
table(results$Match.Level)

# View matched species
results |>
  dplyr::filter(matched) |>
  dplyr::select(Orig.Name, Matched.Name, common_name, endemic)

# With typos (fuzzy matching)
typos <- c("Pumma concolor", "Tremarctos ornatu")  # Spelling errors
results_fuzzy <- validate_peru_mammals(typos, quiet = FALSE)

# Check for ambiguous matches
get_ambiguous_matches(results_fuzzy, type = "genus")

# Access metadata
attr(results, "match_rate")
attr(results, "n_fuzzy_genus")
```

```
# With special "sp." cases
sp_cases <- c("Akodon sp. Ancash", "Oligoryzomys sp. B")
results_sp <- validate_peru_mammals(sp_cases)
# Should match exactly
```

Index