

# Package: taxastand (via r-universe)

August 22, 2024

**Title** Taxonomic Name Standardization

**Version** 1.0.0

**Description** Matches species names to a taxonomic standard. Resolves synonyms consistently and reproducibly.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**SystemRequirements** parsenames  
(<<https://github.com/camwebb/taxon-tools>>), matchnames  
(<<https://github.com/camwebb/taxon-tools>>)

**Imports** assertr, assertthat, digest, dplyr, fs, glue, magrittr, processx, tibble, tidyr

**Roxygen** list( markdown = TRUE, roclets = c("`collate", ``namespace", ``rd", ``roxygenals::global\_roclet"))

**RoxygenNote** 7.1.2

**Depends** R (>= 4.1.0)

**Suggests** rmarkdown, knitr, roxygenals (>= 0.2.1), testthat (>= 3.0.0), babelwhale

**Config/testthat/edition** 3

**Remotes** anthonymnorth/roxygenals

**VignetteBuilder** knitr

**Repository** <https://joelnitta.r-universe.dev>

**RemoteUrl** <https://github.com/joelnitta/taxastand>

**RemoteRef** HEAD

**RemoteSha** 3894436b120367976fb6331ced1c19d001c251f7

## Contents

filmy_taxonomy . . . . .	2
ts_match_names . . . . .	2

ts_parse_names . . . . .	5
ts_resolve_names . . . . .	6
ts_tt_installed . . . . .	8
ts_write_names . . . . .	9

## Index 10

---

filmy_taxonomy	<i>Taxonomy of filmy ferns (family Hymenophyllaceae)</i>
----------------	--

---

### Description

A dataset containing taxonomic names and associated metadata for the fern family Hymenophyllaceae. Downloaded from the [Catalog of Life](#), Version 1.5. All columns formatted according to [Darwin Core standard](#). Only includes taxa at the species or infraspecies level.

### Usage

```
filmy_taxonomy
```

### Format

A data frame with 2729 rows and 31 variables.

### Source

<http://www.catalogueoflife.org/>

---

ts_match_names	<i>Match taxonomic names to a reference</i>
----------------	---

---

### Description

Allows for orthographic differences between query and reference by using fuzzy matching on parsed taxonomic names. Requires [taxon-tools](#) to be installed.

### Usage

```
ts_match_names(
  query,
  reference,
  max_dist = 10,
  match_no_auth = FALSE,
  match_canon = FALSE,
  collapse_infra = FALSE,
  collapse_infra_exclude = NULL,
  simple = FALSE,
```

```

docker = getOption("ts_docker", default = FALSE),
tbl_out = getOption("ts_tbl_out", default = FALSE)
)

```

## Arguments

query	Character vector or dataframe; taxonomic names to be queried. If a character vector, missing values not allowed and all values must be unique. If a dataframe, should be taxonomic names parsed with <a href="#">ts_parse_names()</a> .
reference	Character vector or dataframe; taxonomic names to use as reference. If a character vector, missing values not allowed and all values must be unique. If a dataframe, should be taxonomic names parsed with <a href="#">ts_parse_names()</a> .
max_dist	Max Levenshtein distance to allow during fuzzy matching (total insertions, deletions and substitutions). Default: 10.
match_no_auth	Logical; If no author is given in the query and the name (without author) occurs only once in the reference, accept the name in the reference as a match. Default: to not allow such a match (FALSE).
match_canon	Logical; Allow a "canonical name" match if only the genus, species epithet, and infraspecific epithet (if present) match exactly. Default: to not allow such a match (FALSE).
collapse_infra	Logical; if the specific epithet and infraspecific epithet are the same, drop the infraspecific rank and epithet from the query.
collapse_infra_exclude	Character vector; taxonomic names to exclude from collapsing with <code>collapse_infra</code> . Any names used must match those in query exactly, or they won't be excluded.
simple	Logical; return the output in a simplified format with only the query name, matched reference name, and match type. Default: FALSE.
docker	Logical; if TRUE, docker will be used to run taxon-tools (so that taxon-tools need not be installed).
tbl_out	Logical vector of length 1; should a tibble be returned? If FALSE (default), output will be a data.frame. This argument can be controlled via the option <code>ts_tbl_out</code> ; see Examples.

## Details

taxon-tools matches names in two steps:

1. Scientific names are parsed into their component parts (genus, species, variety, author, etc).
2. Names are fuzzily matched following taxonomic rules using the component parts.

For more information on rules used for matching, [see taxon-tools manual](#).

Parsing is fairly fast (much faster than matching) but can take some time if the number of names is very large. If multiple queries will be made (e.g., to the same large reference database), it is recommended to first parse the names using [ts\\_parse\\_names\(\)](#), and use the results as input to query and/or reference.

collapse\_infra is useful in situations where the reference database does not use names that have the same specific epithet and infraspecific epithet. For example, reference name "Blechnum lunare" and query "Blechnum lunare var. lunare". In this case, if collapse\_infra is TRUE, "Blechnum lunare" will be queried instead of "Blechnum lunare var. lunare". Note that the match\_type will be "exact" even though the literal query and the matched name are different (see example below).

## Value

Dataframe with the following columns (if simple is FALSE):

- query: Query name
- reference: Matched reference name
- match\_type: Type of match (for a summary of match types, [see taxon-tools manual](#))
- id\_query: Unique ID of query
- id\_ref: Unique ID of reference
- genus\_hybrid\_sign\_query: Genus hybrid sign in query
- genus\_name\_query: Genus name of query
- species\_hybrid\_sign\_query: Species hybrid sign in query
- specific\_epithet\_query: Specific epithet of query
- infraspecific\_rank\_query: Infraspecific rank of query
- infraspecific\_epithet\_query: Infraspecific epithet of query
- author\_query: Taxonomic author of query
- genus\_hybrid\_sign\_ref: Genus hybrid sign in reference
- genus\_name\_ref: Genus name of reference
- species\_hybrid\_sign\_ref: Species hybrid sign in reference
- specific\_epithet\_ref: Specific epithet of reference
- infraspecific\_rank\_ref: Infraspecific rank of reference
- infraspecific\_epithet\_ref: Infraspecific epithet of reference
- author\_ref: Taxonomic author of reference

If simple is TRUE, only return the first three columns above.

## Examples

```
if(ts_tt_installed()) {
  ts_match_names(
    "Crepidomanes minutus",
    c("Crepidomanes minutus", "Hymenophyllum polyanthos"),
    simple = TRUE
  )

  # If you always want tibble output without specifying `tbl_out = TRUE`
  # every time, set the option:
  options(ts_tbl_out = TRUE)
  ts_match_names(
```

```

    "Crepidomanes minutus",
    c("Crepidomanes minutum", "Hymenophyllum polyanthos")
  )

# Example using collapse_infra argument
ts_match_names(
  c("Crepidomanes minutus", "Blechnum lunare var. lunare",
    "Blechnum lunare", "Bar foo var. foo", "Bar foo"),
  c("Crepidomanes minutum", "Hymenophyllum polyanthos", "Blechnum lunare",
    "Bar foo"),
  collapse_infra = TRUE,
  collapse_infra_exclude = "Bar foo var. foo",
  simple = TRUE
)
}

```

---

ts_parse_names	<i>Parse taxonomic names</i>
----------------	------------------------------

---

## Description

Requires [taxon-tools](#) or docker to be installed.

## Usage

```

ts_parse_names(
  taxa,
  tbl_out = getOption("ts_tbl_out", default = FALSE),
  quiet = FALSE,
  docker = getOption("ts_docker", default = FALSE)
)

```

## Arguments

taxa	Character vector; taxon names to be parsed by taxon-tools parsenames. Missing values not allowed. Must all be unique.
tbl_out	Logical vector of length 1; should a tibble be returned? If FALSE (default), output will be a data.frame. This argument can be controlled via the option ts_tbl_out; see Examples.
quiet	Logical; if TRUE, suppress warning messages that would normally be issued
docker	Logical; if TRUE, docker will be used to run taxon-tools (so that taxon-tools need not be installed).

## Details

Parses scientific names into their component parts (genus, species, variety, author, etc).

**Value**

A dataframe including the following columns.

- id: A unique ID number assigned to the input name
- name: The input name
- genus\_hybrid\_sign: Hybrid sign for genus
- genus\_name: Genus name
- species\_hybrid\_sign: Hybrid sign for species
- specific\_epithet: Specific epithet (name)
- infraspecific\_rank: Infraspecific rank
- infraspecific\_epithet: Infraspecific epithet (name)
- author: Name of taxon

**Examples**

```
# Using local taxon-tools installation
if (ts_tt_installed()) {

  ts_parse_names("Foogenus x barspecies var. foosubsp (L.) F. Bar")
  ts_parse_names(
    "Foogenus x barspecies var. foosubsp (L.) F. Bar", tbl_out = TRUE)

  # If you always want tibble output without specifying `tbl_out = TRUE`
  # every time, set the option:
  options(ts_tbl_out = TRUE)
  ts_parse_names("Foogenus x barspecies var. foosubsp (L.) F. Bar")
  ts_parse_names("Crepidomanes minutum (Blume) K. Iwats.")

}

# Using docker
if (babelwhale::test_docker_installation()) {

  ts_parse_names(
    "Foogenus x barspecies var. foosubsp (L.) F. Bar",
    docker = TRUE)

}
```

---

ts\_resolve\_names

*Resolve synonyms in taxonomic names*


---

**Description**

After matching taxonomic names to a reference, some may match synonyms. This function resolves synonyms to their accepted names.

**Usage**

```
ts_resolve_names(
  query,
  ref_taxonomy,
  max_dist = 10,
  match_no_auth = FALSE,
  match_canon = FALSE,
  collapse_infra = FALSE,
  collapse_infra_exclude = NULL,
  docker = getOption("ts_docker", default = FALSE),
  tbl_out = getOption("ts_tbl_out", default = FALSE)
)
```

**Arguments**

query	Character vector or dataframe; taxonomic names to be resolved. If a character vector, missing values not allowed and all values must be unique. If a dataframe, should be taxonomic names matched with <a href="#">ts_match_names()</a> .
ref_taxonomy	Dataframe; reference taxonomic data adhering to the <b>Darwin Core standard</b> with the following columns: <ul style="list-style-type: none"> <li>• taxonID: <b>Unique identifier for each taxon.</b></li> <li>• acceptedNameUsageID: If the taxon is a synonym, the <b>unique identifier for the accepted name</b></li> <li>• taxonomicStatus: <b>The status of the use of the scientificName as a label for the taxon.</b></li> <li>• scientificName: <b>The full scientific name of the taxon</b>, with authorship and date information if known.</li> </ul>
max_dist	Max Levenshtein distance to allow during fuzzy matching (total insertions, deletions and substitutions). Default: 10.
match_no_auth	Logical; If no author is given in the query and the name (without author) occurs only once in the reference, accept the name in the reference as a match. Default: to not allow such a match (FALSE).
match_canon	Logical; Allow a "canonical name" match if only the genus, species epithet, and infraspecific epithet (if present) match exactly. Default: to not allow such a match (FALSE).
collapse_infra	Logical; if the specific epithet and infraspecific epithet are the same, drop the infraspecific rank and epithet from the query. For more information, see <a href="#">ts_match_names()</a> .
collapse_infra_exclude	Character vector; taxonomic names to exclude collapsing with collapse_infra. Any names used must match those in query exactly, or they won't be excluded.
docker	Logical; if TRUE, docker will be used to run taxon-tools (so that taxon-tools need not be installed).
tbl_out	Logical vector of length 1; should a tibble be returned? If FALSE (default), output will be a data.frame. This argument can be controlled via the option ts_tbl_out; see Examples.

## Details

query can take as input either a character vector of taxonomic names, or the output of `ts_match_names()`. If the former, it will run `ts_match_names()` to match the query to `ref_taxonomy`, then resolve synonyms. If the latter, the scientific names in `ref_taxonomy` should be the same used as reference with `ts_match_names()` (this is not checked).

`ref_taxonomy` must be taxonomic data adhering to the **Darwin Core standard**. Darwin Core includes many terms, but only four (`taxonID`, `acceptedNameUsageID`, `taxonomicStatus`, and `scientificName`) are required for this function.

## Value

Dataframe; results of resolving synonyms in matched taxonomic names. Includes the following columns:

- `query`: Query name
- `resolved_name`: Accepted name after resolving synonyms
- `matched_name`: Name matched to query
- `resolved_status`: Taxonomic status of the resolved name (same as `taxonomicStatus` in `ref_taxonomy`)
- `matched_status`: Taxonomic status of the matched name (same as `taxonomicStatus` in `ref_taxonomy`)
- `match_type`: Type of match (for a summary of match types, see [taxon-tools manual](#))

Names that could not be matched or resolve to multiple, different synonyms have NA for `resolved_name`.

## Examples

```
if (ts_tt_installed()) {
  # Load reference taxonomy in Darwin Core format
  data(filmy_taxonomy)

  ts_resolve_names("Gonocormus minutum", filmy_taxonomy)
  # If you always want tibble output without specifying `tbl_out = TRUE`
  # every time, set the option:
  options(ts_tbl_out = TRUE)
  ts_resolve_names("Gonocormus minutum", filmy_taxonomy)
}
```

---

<code>ts_tt_installed</code>	<i>Test if <a href="https://github.com/camwebb/taxon-toolstaxon-tools">Rhrefhttps://github.com/camwebb/taxon-toolstaxon-tools</a> is installed</i>
------------------------------	--

---

## Description

Test if **taxon-tools** is installed



**Usage**

```
ts_tt_installed()
```

**Value**

TRUE if **taxon-tools** is installed, or FALSE if not.

**Examples**

```
ts_tt_installed()
```

---

ts_write_names	<i>Write out parsed names to a text file</i>
----------------	--

---

**Description**

Write out parsed names to a text file

**Usage**

```
ts_write_names(df, path)
```

**Arguments**

df	Dataframe with parsed names
path	Path to write dataframe

Writes out parsed names in a format that can be used by **taxon-tools** (each part of the scientific name is separated by the pipe symbol (|), with one name per line).

**Value**

Path to parsed names

**Examples**

```
if (ts_tt_installed()) {  
  parsed_names <- ts_parse_names(  
    "Foogenus x barspecies var. foosubsp (L.) F. Bar")  
  temp_file <- tempfile()  
  ts_write_names(parsed_names, temp_file)  
  readLines(temp_file)  
  file.remove(temp_file)  
}
```

# Index

## \* datasets

filmy\_taxonomy, 2

filmy\_taxonomy, 2

ts\_match\_names, 2, 7, 8

ts\_parse\_names, 3, 5

ts\_resolve\_names, 6

ts\_tt\_installed, 8

ts\_write\_names, 9