Using **plantR** to Manage Taxonomy

Renato A. F. de Lima*

24 abril 2025

Contents

1 Introduction **1**

2 Installing plantR **2**

3 A practical example **2**
3.1 Preparing names using `fixSpecies()` 3
3.1.1 Internal functions 4
3.2 Validating taxon names using `prepSpecies()` 4
3.2.1 Internal functions 6
3.3 Validating family names using `prepFamily()` 6

4 Brief code summary **7**

5 Citation **7**

6 Bug report and suggestions **7**

1 Introduction

One of the most important steps in managing and using information from biological datasets is the management of taxonomic nomenclature. Taxon names need to be correctly spelled and we should use their currently accepted names based on a given taxonomic backbone (CITE 10 SIMPLE RULES). However, there are many small corrections and standardizations that need to be made before one can cross-validate taxonomic nomenclature, which can become quite burdensome as the size of the dataset increases.

plantR provides tools to format, spell-check and validate taxon names at the family and species levels and using different taxonomic backbones. The default backbone used by **plantR** is the Flora e Funga do Brasil. However, any

*Universidade de São Paulo, https://github.com/LimaRAF

taxonomic backbone can be used, as long as it has a specific content and format. The companion R package **plantRdata** provides other backbones already in this specific format from the World Flora Online, the World Checklist of Vascular Plants and GBIF.

To simplify the process, the management of taxonomic nomenclature in **plantR** can be simply done using the wrapper function `formatTax()` (see the brief code summary below). But first, we will explain in detail each step and function to manage taxonomy within **plantR**. These functions can be applied individually, as we will show below, but the management of taxonomic nomenclature is potentialised when they are executed in a specific order.

2 Installing plantR

The package can be installed and loaded from GitHub with:

```
install.packages("remotes")
library("remotes")
install_github("LimaRAF/plantR")
library("plantR")
```

You can also download the development (and probably the most up-to-date) version of the package with:

```
install_github("LimaRAF/plantR", ref = "dev")
library("plantR")
```

3 A practical example

Let´s create a list of names with many common issues in biological datasets. This list includes issues related to format, casing, spelling, synonyms, typos, names with/without authors, incomplete identifications, etc.

```
names <- c(
  "Lindsaea sp.", "Lindsaeaceae sp.",
  "Lindsaea lancea", "Lindsaea lancia", "Lindsaea pumila",
  "Lindsaea lancea (L.) Bedd.",
  "lindsaea lancea", "Lindsaea Lancea", "LINDSAEA LANCEA",
  "Lindsaea lancea var. Falcata",
  "Lindsaea lancea var falcata",
  "Lindsaea Aff. lancea",
  "Lindsaea Aff.lancea",
  "Lindsaea aff. lancea (L.) Bedd.",
  "Lindsaea ×improvisa K.U.Kramer",
  "Parablechnum C.Presl",
  "Blechnum spannagelii Rosenst.",
  "Blechnum cf. spannagelii", "Blechnumcf.spannagelii",
```

```
  "Blechnum austrobrasilianum de la Sota",
  "Casearia sylvestris var. angustifolia",
  "Casearia sylvestris var. angustifolia Uittien",
  "Casearia sylvestris angustifolia Uittien",
  "Casearia sylvestris Sw. var. sylvestris",
  "Blechnaceae1",
  "Blechnum sp.2", "Blechnum sp. 2", "Blechnum sp 2", "Blechnum sp",
  "indet", "Indeterminada1"
)
```

3.1 Preparing names using `fixSpecies()`

This function accepts both a vector of names or a data frame. In the latter case, the data frame should contain the taxon names in a column called, by default, 'scientificName'. The name of this column can be defined by the user through the argument `tax.name`. And if taxon names and authorities are available in separate columns, which is advised, please make sure that author names are stored in a column called 'scientificNameAuthorship' or alter this column name using the argument `author.name`.

Here we provide to the function the vector of names defined above:

```
names_fixed <- fixSpecies(names)
head(names_fixed[,-c(2,4)], 7)
```

```
#>               scientificName scientificName.new scientificNameStatus
#> 1               Lindsaea sp.       Lindsaea sp.                indet
#> 2           Lindsaeaceae sp.   Lindsaeaceae sp.      family_as_genus
#> 3            Lindsaea lancea    Lindsaea lancea          possibly_ok
#> 4            Lindsaea lancia    Lindsaea lancia          possibly_ok
#> 5            Lindsaea pumila    Lindsaea pumila          possibly_ok
#> 6 Lindsaea lancea (L.) Bedd.    Lindsaea lancea       name_w_authors
#> 7            lindsaea lancea    Lindsaea lancea    name_w_wrong_case
```

The output of `fixSpecies()` is a data frame that contains the columns necessary for the name validation step below. For each name, it is returned a new column containing a suggestion of a more standardised name (i.e. 'scientificName.new'), with the isolation of the name modifiers (e.g. cf. or aff.). Also, if the name contains author names (and if those author names are in a detectable format), they are split into two different columns ('scientificName.new' and 'scientificNameAuthorship.new').

The new column 'scientificNameStatus' stores all possible flags detected in the original name. These flags include the detection and standardization of open nomenclature (e.g. aff., cf.), infra-specific levels (e.g. var., subsp., f.), hybrids, incomplete identifications and undeterminations. It also flags and solves issues related to name casing and the notation of morphotypes.

3.1.1 Internal functions

Besides the flagging of particular cases of taxon names, the function `fixSpecies()` contains different internal functions that standardise the:

- notation of open nomenclature abbreviations and name modifiers (function `fixAnnotation()`);
- notation of incomplete identifications and undeterminations (function `fixIndet()`);
- the casing of taxon names (function `fixCase()`);
- the separation between taxon names and taxon name authorships (function `fixAuthors()`).

These internal functions may also be useful in itself depending on the user's goals. Below, some examples of their isolate use:

```
plantR:::fixAnnotation(c("Lindsaea lancea var falcata", "Lindsaea Aff.lancea"))
#> [1] "Lindsaea lancea var. falcata" "Lindsaea aff. lancea"
plantR:::fixIndet(c("Indet1", "Blechnum sp. 2", "Blechnum sp 2", "Blechnum sp"))
#> [1] "Indet. sp.1"   "Blechnum sp.2" "Blechnum sp.2" "Blechnum sp."
plantR:::fixCase(c("lindsaea lancea", "Lindsaea Lancea", "LINDSAEA LANCEA"))
#>   lindsaea lancea   Lindsaea Lancea   LINDSAEA LANCEA
#> "Lindsaea lancea" "Lindsaea lancea" "Lindsaea lancea"
plantR:::fixAuthors(c("Lindsaea lancea (L.) Bedd.", "Parablechnum C.Presl"))
#>                    orig.name        tax.name tax.author
#> 1 Lindsaea lancea (L.) Bedd. Lindsaea lancea (L.) Bedd.
#> 2       Parablechnum C.Presl    Parablechnum    C.Presl
```

3.2 Validating taxon names using `prepSpecies()`

After the standardization of notation and format of taxon names, it is possible to validate them against a taxonomic backbone, aiming at finding synonyms, orthographic variants and/or typos.

In **plantR** this validation of the taxonomic nomenclature is done using the function `prepSpecies()`:

```
names_valid <- prepSpecies(names_fixed,
                           tax.names = c("scientificName.new",
                                         "scientificNameAuthorship.new"))
head(names_valid[,-c(2,3,4,9,11)], 7)
#>               scientificName scientificNameStatus suggestedFamily   suggestedName suggest
#> 1               Lindsaea sp.                indet    Lindsaeaceae        Lindsaea
#> 2          Lindsaeaceae sp.      family_as_genus    Lindsaeaceae    Lindsaeaceae
#> 3           Lindsaea lancea          possibly_ok    Lindsaeaceae Lindsaea lancea
#> 4           Lindsaea lancia          possibly_ok    Lindsaeaceae Lindsaea lancea
#> 5           Lindsaea pumila          possibly_ok    Lindsaeaceae Lindsaea lancea
#> 6 Lindsaea lancea (L.) Bedd.       name_w_authors    Lindsaeaceae Lindsaea lancea
#> 7           lindsaea lancea    name_w_wrong_case    Lindsaeaceae Lindsaea lancea
```

```
#>           scientificNameFull
#> 1         Lindsaea Pic.Serm.
#> 2       Lindsaeaceae C.Presl
#> 3 Lindsaea lancea (L.) Bedd.
#> 4 Lindsaea lancea (L.) Bedd.
#> 5 Lindsaea lancea (L.) Bedd.
#> 6 Lindsaea lancea (L.) Bedd.
#> 7 Lindsaea lancea (L.) Bedd.
```

The output of the function is optimized to ease the user´s interpretation and decision-making. The output column 'tax.notes' contains the important comments related to the name validation, while the argument `drop.cols` controls which columns should be returned (set `drop.cols = ""` to get all columns). By default, synonyms and orthographic variants are replaced by the name accepted in the reference backbone.

In addition, `prepSpecies()` has arguments to control the minimum similarity allowed between names in fuzzy matching (argument `sug.dist`) and the editing and cleaning of names before matching (arguments `clean.indet` and `clean.names`).

The function was conceived to validate thousands of names at once. So, to speed up the validation process, users can validate names by their initial letter (argument `split.letters`) or parallelise the computation (arguments `parallel` and `cores`).

By default, `prepSpecies()` uses the internal backbone from the Flora e Funga do Brasil. But any backbone in the right format can be provided by the user via the argument `db` of `prepSpecies()`. The **plantRdata** package, available only on GitHub at this link, provides preformatted objects with backbones from different sources.

```
# using the World Flora Online
names_valid_wfo <- prepSpecies(names_fixed,
                               tax.names = c("scientificName.new",
                                         "scientificNameAuthorship.new"),
                               db = plantRdata::wfoNames)
# using the World Checklist of Vascular Plants
names_valid_wcvp <- prepSpecies(names_fixed,
                                tax.names = c("scientificName.new",
                                         "scientificNameAuthorship.new"),
                                db = plantRdata::wcvpNames)
# Comparing the results
names_bfo_wfo_wcvp <- cbind.data.frame(names_valid$scientificName.new,
                                  names_valid$scientificNameFull,
                                  names_valid_wfo$scientificNameFull,
                                  names_valid_wcvp$scientificNameFull)
diff <- names_valid$scientificNameFull != names_valid_wfo$scientificNameFull
```

```
diff[is.na(diff)] <- FALSE
head(names_bfo_wfo_wcvp[diff, ], 3)
#>   names_valid$scientificName.new names_valid$scientificNameFull names_valid_wfo$scientifi
#> 1                  Lindsaea sp.             Lindsaea Pic.Serm.            Lindsaea Dryan
#> 2               Lindsaeaceae sp.           Lindsaeaceae C.Presl           Lindsaeaceae M.
#> 5                Lindsaea pumila     Lindsaea lancea (L.) Bedd.        Asplenium dielerec
#>   names_valid_wcvp$scientificNameFull
#> 1             Lindsaea Dryand. ex Sm.
#> 2                   Lindsaeaceae sp.
#> 5         Asplenium dielerectum Viane
```

Note that the computing speed when using larger backbones (over a million names) is very different than when using smaller ones (a few hundred thousand names). This is because the name-matching process is done pairwise between the input and reference names. But please try to play with the arguments `split.letters` and `parallel` and `cores` to investigate how fast it can be.

Note as well that each taxonomic backbone provides a different opinion on each name. So it is up to the user to select the best backbone for its own purposes.

3.2.1 Internal functions

The most important internal function used by `prepSpecies()` is `nameMatching()`, which is the function that actually performs the exact and fuzzy matching between the input names against the reference names from the taxonomic backbone selected. As mentioned above, the name matching can be parallelized and/or performed separately by initial letters to speed up computational time for larger datasets. But basically, it compares a set of input and reference names trying to find exact or fuzzy matches:

```
input_names <- c("Casearia silvestris", "Casearia decandra")
ref_names <- c("Casearia aculeata", "Casearia arborea",
               "Casearia decandra", "Casearia sylvestris")
nameMatching(input_names, ref_names)
#> [1] 4 3
```

3.3 Validating family names using `prepFamily()`

plantR contains an internal dictionary of valid family names which can be used via the function `prepFamily()`. Currently, valid family names are based on the APG IV (angiosperms) and the PPG I (ferns and lycophytes). And if the family info is missing it can also be found based on genus information.

```
names_fam_valid <- prepFamily(names_valid,
                              fam.name = "suggestedFamily",
                              spp.name = "scientificName.new")
```

It was not the case in this example, but `prepFamily()` returns warnings in the case of conflicts between the original family names and the family name

suggested by the APG IV/PPG I family list. In this example, `prepFamily()` just added the missing family names based on the names of the genera, in the new column called 'family.new'.

4 Brief code summary

In this tutorial, we described many steps and details related to the functions that manage taxonomy within **plantR**. Here we provide the main codes that you actually need to include in your scripts, assuming that you already created the vector of taxon names called `names`:

```
names_fixed <- fixSpecies(names)
names_valid <- prepSpecies(names_fixed,
                           tax.names = c("scientificName.new",
                                         "scientificNameAuthorship.new"))
names_valid <- prepFamily(names_valid,
                          fam.name = "suggestedFamily",
                          spp.name = "scientificName.new")
```

Or, as mentioned above, even simpler using the wrapper `formatTax()`:

```
names_df <- data.frame(scientificName = names)
names_df_valid <- formatTax(names_df)
```

5 Citation

If you use **plantR**, please cite it as:

Lima, R.A.F., Sánchez-Tapia, A., Mortara, S.R., ter Steege, H., Siqueira, M.F. (2021). *plantR*: An R package and workflow for managing species records from biological collections. Methods in Ecology and Evolution 14(2): 332-339. https://doi.org/10.1101/2021.04.06.437754

And please also cite the taxonomic backbones that you used!

6 Bug report and suggestions

The plantR project is hosted on GitHub. Please report any bugs and suggestions for improvements to the package here.