Manipulating the *gbif4crest* calibration data

Manuel Chevalier

2023-02-22

Note: If you have your own calibration data, this vignette is of little interest to you. Refer to the documentation of `crest.set_modern_data()`.

Introduction

`crestr` comes with a built-in global calibration dataset for several proxies (see details in other vignettes andd the didicated webpage). This dataset can be accessed in two different ways: - The data can be accessed using *crestr* built-in functions, which connects to a cloud-based database. This solution requires an active internet connection and can be slow. The database in hosted on an Amazon server, which is not accessible from some countries or if your internet connection is not secure. - The entire calibration dataset can be downloaded from [here]. However, it is a large file once unzipped (>23Gb). The data extraction phase of the package is, however, MUCH faster.

The two options can be easily selected from the package. But let's begin by loading `crestr`.

```
library(crestr)
```

In this example, we will work with a reduced dataset, composed of three pollen taxa:

```
PSE
#>   Level     Family     Genus Species ProxyName
#> 1     1  Ericaceae      <NA>    <NA> Ericaceae
#> 2     2 Asteraceae Artemisia    <NA> Artemisia
#> 3     2   Oleaceae      Olea    <NA>      Olea
```

Using the *gbif4crest* database

Independently of how you want to access the calibration data, you will use the `crest.get_modern_data()` if you want to use the *gbif4crest* data.

If you want to access the cloud-based data, you will do the following:

```
reconstr <- crest.get_modern_data(  pse = PSE,
                                    taxaType = 1,
                                    climate = c("bio1", "bio12"),
                                    # The name of the online database
                                    dbname = "gbif4crest_02",
                                    verbose = TRUE
)
```

If you have downloaded and unzipped the *gbif4crest* dataset on your computer, you can use the following:

```
reconstr <- crest.get_modern_data(  pse = PSE,
                                    taxaType = 1,
                                    climate = c("bio1", "bio12"),
                                    # The full path to the local database
                                    # Or place the database in the working
                                    #   directory.
                                    dbname = "path/to/gbif4crest_02.sqlite3",
                                    verbose = TRUE
)
```

> Note: The local calibration database file can be saved in your working directory or anywhere else and renamed in any way. The file extension '.sqlite3' is, however, the only difference between the online and the offline options. It is thus crucial to call the file database.sqlite3 to orient the package in the right direction.

Local subset of the calibration dataset

A third option exists (`crestr` > v1.3.0) that combines the strength of the two above approaches: subsetting the online database and save it locally. This way, you do not need to save a huge file on your computer and you only need an active internet connection to create the subset. Such subsets can be created using the `dbSubset()` function.

```
dbSubset( taxaType = 1,
          xmn = 0, xmx = 50, ymn = -45, ymx = 0,
          out = 'gbif4crest_southernAfrica.sqlite3',
          verbose = TRUE
      )
```

This will create a local snapshot of the database in your working directory (unless you give it a different path with `out`). You can then use this restricted dataset with `crest.get_modern_data()` and all other functions as if it was the "full" dataset. This restricted database is only 45Mb and the storage problem is therefore avoided.

```
reconstr <- crest.get_modern_data(  pse = PSE,
                                    taxaType = 1,
                                    climate = c("bio1", "bio12"),
                                    # The full path to the local database
                                    # Or place the database in the working
                                    #   directory.
                                    dbname = "gbif4crest_southernAfrica.sqlite3",
                                    verbose = TRUE
)
```