crestr (v1.3.0)

cheat sheet

last update: 06/06/2023

How to install?

The latest development version is currently only available on GitHub. It can be installed using the **devtools** package:

```
> install.packages('devtools')
```

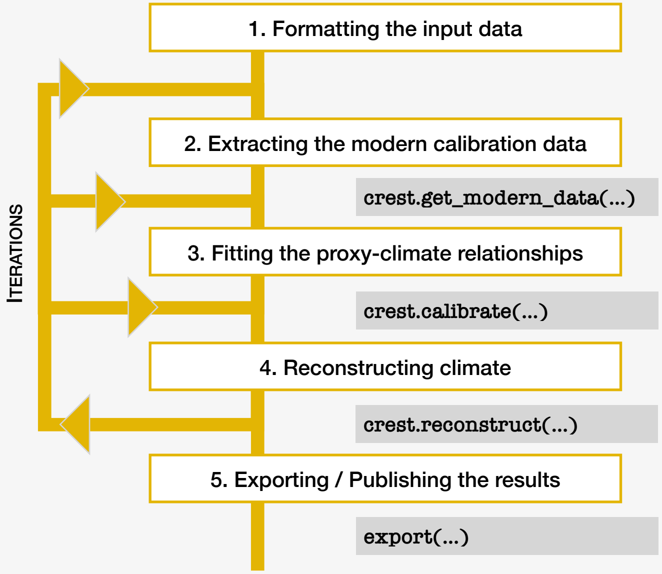
```
> devtools::install_github('mchevalier2/crestr')
```

R will most likely ask you to install missing packages or update existing packages. You should install all the missing packages. Updates are not enforced, but this might create compatibility issues. Finally you can load the package with:

```
> library(crestr)
```

The whole game

A CREST analysis can be divided in five successives steps. It is, however, generally not a linear process and you will be expected to iterate through the different steps to maximise the quality of your reconstructions. Each step is described in the following sections.


1. Formatting the input data
2. Extracting the modern calibration data
crest.get_modern_data(...)
3. Fitting the proxy-climate relationships
crest.calibrate(...)
4. Reconstructing climate
crest.reconstruct(...)
5. Exporting / Publishing the results
export(...)
ITERATIONS

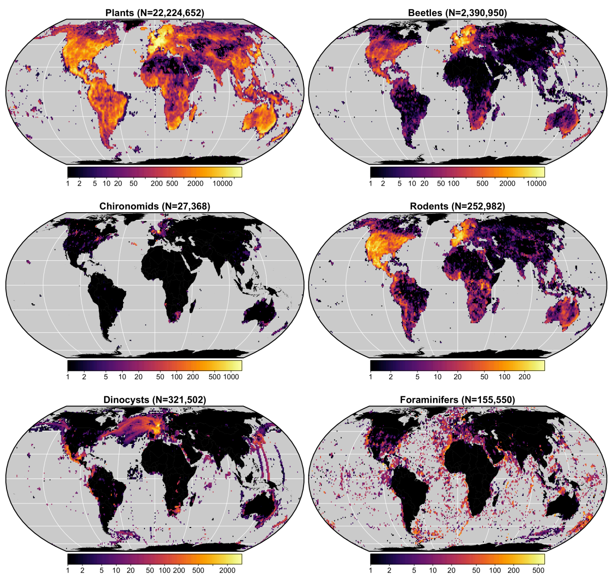
Join the *crestr* mailing-list

Become a member of the CREST community and receive news about package updates, novel/innovative applications, and early access to *crestr* training workshop. Sign up at the following link: https://mchevalier2.github.io/crestr/articles/community.html

The *gbif4crest_02* calibration data

crestr can be used with a curated calibrated dataset called *gbif4crest_02* that contains the modern distribution data for 6 palaeoecological proxies (see maps). The selection of the proxy is done with the parameter **proxyType** (1: plants, 2: beetles, 3: chironomids, 4: forams, 5: diatoms, 6: rodents).

Number of occurrence data per grid cell for six common biological climate proxies



This cloud-based dataset can be accessed via different routes:

1. As illustrated throughout this cheat sheet, it can be remotely accessed using the **crest.get_modern_data()** function (see Step 2). This online service can be slow and is unfortunately not accessible from every country.
2. Alternatively, the whole database can be downloaded locally. Once unzipped the file is, however, rather large (~21Gb).

```
> dbDownload(dbname='gbif4crest_02', out='path/to/db')
```

3. A solution combining the advantages of both routes consists in creating local snapshot of the online database from geographical constraints. The snapshot can be generated directly from the online database or after downloading the database locally

```
> dbSubset(dbname='gbif4crest_02', taxatype, xnx, xmx,
           ymn, ymx, out='path/to/dbsubset')
```

NOTE: Options 2 and 3 create a .sqlite3 file on your computer to be used in every function requiring database access by replacing **dbname = "gbif4crest_02"** by **dbname="path/to/db.sqlite3"**.

Each grid cell with observation is assigned a range of climate parameters that can be used for reconstructions (see **accClimateVariables()**) and a range of descriptors to refine the definition of the calibration dataset (see **accBasinNames()**, **accCountryNames()**, and **accRealmNames()**).

Step 1: Formatting the input data

crestr requires simple data formats, *i.e.* R data frames. While different objectives will require different files, most uses will be based on the analysis of fossil data based on a corresponding PSE file to resolve the taxonomy. These input files are as follow:

- The fossil data to inform the reconstructions can be expressed as raw counts, percentages (recommended), proportions, or any form of weighting scheme to be employed (advanced users), and in the following structure:

Age/Depth	Taxon_1	Taxon_2	Taxon_3	…	Taxon_n
10	…	…	…	…	…
20	25	35	5	…	12
30	…	…	…	…	…
40	…	…	…	…	…

- The Proxy-Species Equivalency (PSE) file that links individual species of the *gbif4crest_02* dataset to the observed taxa. The classification can be done at several taxonomical level depending on your data. Use **createPSE()** to generate a template to fill in.

Level	Family	Genus	Species	ProxyName
1	Family_1			Taxon_1
2	Family_2	Genus_1		Taxon_2
2	Family_2	Genus_2		Taxon_2
…	…	…	…	…
3	Family_3	Genus_3	Species_1	Taxon_3

- By default, the package gets calibration data for the embedded *gbif4crest_02* database. Still, using this database is optional and other calibration data can used. However, they must be formatted as illustrated below and used with the **crest.set_modern_data()** function (described here). This file replaces the PSE file.

Species	Taxa	Lon	Lat	clim_1	…	clim_n
Species1	Taxon_1	…	…	…	…	…
Species2	Taxon_1	…	…	…	…	…
…	…	…	…	…	…	…
SpeciesX	Taxon2	…	…	…	…	…

Step 2: Extracting the calibration data

This first function is designed to extract all the necessary calibration data from the *gbif4crest_02* database and format them. The specificities of these data (*e.g.* spatial extent, diversity, climate variables) must be determined by the characteristics of the studied record. The selection can be done based on many parameters, such as geographical coordinates, elevation, countries and biomes.

```
> recon <— crest.get_modern_data( df = fossil_data,
      pse = PSE, climate= c('bio1', 'bio12'), dbname =
      "gbif4crest_02", xmn, xnx, ymn, ymx, elev_min,
      countries, biomes, ... )
```

This function will create a **crestObj**, here called **recon**, that will become the main parameter of most functions that will be subsequently used. Details about all the function parameters available [here].

Step 3: Fitting proxy-climate relationships

Once a calibration dataset adapted to the analysed fossil record has been formatted, possibly after a few iterations, the proxy-climate relationships can be fitted as follow:

```
> recon <— crest.calibrate( recon, shape=c('normal',
          'lognormal', bin_width=c(2, 100), ... )
```

A **shape** parameter must be defined for each variable (a normal/ lognormal shape is recommended for temperature- / precipitation-like variables). The **bin_width** parameter is less critical and should be selected so that it divides the range of the climate space in 15-25 bins of equal width. Details about all the function available [here].

Step 4: Reconstructing climate

Finally, climate parameters can be reconstructed with:

```
> recon <— crest.reconstruct( recon, taxWeight=
          'normalisation', presenceThreshold=0)
```

The **taxWeight** parameter can take different values depending on the data: 'originalData', 'presence/absence', 'percentages' or 'normalisation'. The normalisation is recommended for proxies with different production rates (*e.g.* pollen data). The **presenceThreshold** parameter excludes all the observations lower than its value (*e.g.* to reduce the noise in the input data). Details about all the function parameters available [here].

Since CREST is an indicator species method, the influence of every taxa on the results can be measured using a leave-one-out (LOO) analysis, in which new reconstructions are performed by excluding one taxon at the time. The anomaly between the 'true' and the 'loo' reconstruction is an estimate of the taxon's influence on the results.

```
> recon <— loo( recon )
```

Combined with the graphical diagnostic tools provided in *crestr* and ecological considerations, individual taxa can be included or excluded from the analysis using **includeTaxa()** and **excludeTaxa()**. You can select different sets of taxa for each climate variable you reconstruct.

Step 5: Exporting and citing the results

Once you have satisfactory results, you can export them with:

```
> export( recon, loc='path/to/folder', as.csv=FALSE,
          fullUncertainties=TRUE, loo=TRUE,
          weights=TRUE, pdfs=FALSE )
```

This function will create a .RData file to reload the **crestObj** in R later (using the **load()** function) and a spreadsheet with the results. All the metadata of the study will be exported in the spreadsheet. In particular, this will include a list of references to cite if the results are published. The same information can be obtained from:

```
> cite_crest( recon )
```

Your analysis relies on several datasets, methods and software. Please cite all the suggested references in any 'official' document.

Interacting with a crestObj

The data stored in a **crestObj** can be read, plotted and modified in numerous ways. While users can directly modify these objects (see the results of **str(recon)** or this [page] for an overview of its structure), it is recommended to use the functions of the package. Details about all the parameters and examples available [here].

Modify the data:

```
> includeTaxa( recon, taxa=c('taxon1', 'taxon2'), climate='bio1')
> excludeTaxa( recon, taxa=c('taxon3'), climate='bio12')
```

Summarise the data:

```
> getSpeciesDiversity( recon )
> taxonComposition( recon, taxanames='Taxon1')
> pdf_ranges( recon, orderby='bio1')
```

Plot the data:

```
> plot( recon )
> plot_climateSpace( recon, ... )
> plot_combinedPDFs( recon, ... )
> plot_diagram( recon, ...)
> plot_loo( recon, ...)
> plot_scatterPDFs( recon, ...)
> plot_taxaCharacteristics( recon, ...)
> plot_violingPDFs( recon, ...)
```

Exporting the data outside of R:

```
> export( recon )
> copy_crest( recon, climate='bio1', uncertainties=TRUE )
> crest.simplify( recon )
> createPSE( recon)
```

Additional reading

Online documentation: https://mchevalier2.github.io/crestr/index.html

About the method: Chevalier, M., Cheddadi, R., Chase, B.M., 2014. CREST (Climate REconstruction SofTware): a probability density function (PDF)-based quantitative climate reconstruction method. https://doi.org/10.5194/cp-10-2081-2014

About the calibration dataset: Chevalier, M., 2019. Enabling possibilities to quantify past climate from fossil assemblages at a global scale. https://doi.org/10.1016/j.gloplacha.2019.01.016

About the package: Chevalier, M., 2022. crestr: an R package to perform probabilistic climate reconstructions from palaeoecological datasets. https://doi.org/10.5194/cp-18-821-2022

About the taxon selection: Chevalier, M., Chase, B.M., Quick, L.J., Scott, L., 2021. An atlas of southern African pollen types and their climatic affinities, in: Quaternary Vegetation Dynamics – The African Pollen Database. CRC Press, London, pp. 239–258. https://doi.org/10.1201/9781003162766-15

A practice dataset

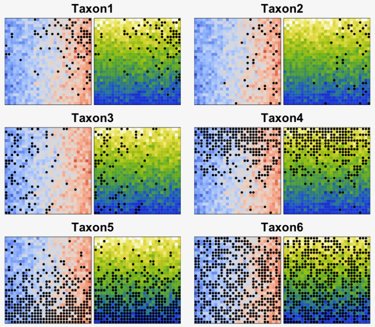
crestr comes with an example dataset to familiarise yourself with the environment. **I highly recommend following the tutorial before using your own data.** All the necessary R commands with explanations are accessible [here].

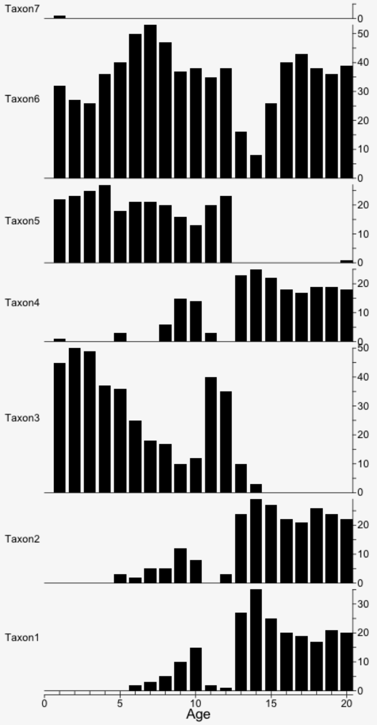
The tutorial presents the key steps of a reconstruction with artificial data with controlled settings. The six taxa have predefined climate niches and their temporal variations were generated to illustrate climate changes from cold & dry to warm & wet conditions.

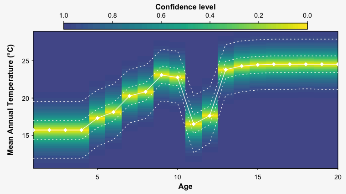
The dataset is included in the R package and can be loaded with: **data(crest_ex)** [the fossil data represented as a barplot diagram on the right] and **data(crest_ex_pse)** [the PSE file]. The distribution data are stored in the *gbif4crest_02* database and can be accessed by using **dbname='crest_example'**.

The tutorial takes you through the five steps (preparing the data, acquiring the calibration data, fitting the proxy-climate relationships, reconstructing climate and exporting the results) and illustrates how the different plots can inform the iterative process. In particular, it illustrates the process of identifying climate-sensitive taxa and how to select a unique set of climate-sensitive taxa for each variable.

Once successfully completed, format your data as required and start your analysis!







Contact

crestr is constantly evolving as more functionalities are added and bugs fixed. It is thus recommended to regularly update the package. If you experience any problem or if you would like new functionalities to be added, contact me at chevalier.manuel@gmail.com.