

How to expand TR8

Gionata Bocci
Pisa (ITALY)
boccigionata@gmail.com

October 31, 2014

This document briefly explains how users can include their functions for downloading traits in to the TR8 package.

Beware: this document assumes that the reader has a good knowledge of the R programming language and has experience in writing R packages.

The best way to proceed is to clone the github repository (at <https://github.com/GioBo/TR8>) and try to work on that.

Suppose that the user has created a function called `funct_tr` which retrieves the following traits

- Height
- Dispersal type
- Clonality

from a database called `New_db` which is available on the Internet.

1 Adapt your function to TR8

The function should be structured in such a way that:

- it accepts the following parameters

species_list the list of species names provided by the user (without authors' names);

TRAITS the list of traits chosen by the user among those made available by `New_db` (either passing them to the `tr8` function or choosing them through the GUI);

rest a number which establishes the `Sys.sleep` pause before each new query to `New_db` (TR8 uses `Sys.sleep` in order not to overload remote servers: please always conform to this policy in writing extensions to TR8).

- within the function an object of class `results` is created (the definition of this S4 class is defined inside `TR8`)

```
> res<-new("results")
```

- if `TRAIT` is `NULL`, then `res@results` should be set to `NULL` (it means that these traits were not required by the user)

```
> res@results<-NULL
```

- **OR**, if `TRAIT` is not `NULL` then the selected traits for the requested species should be retrieved from `New_db` and a dataframe `species*trait` is created (let's call it `datatraits`); `NA` must be used for data which are not available (thus a plant species which is not present in `New_db` should have all `NA`s in such a dataframe)

- the dataframe `species*traits` is included in the `@result` slot of the object

```
> res@results<-datatraits
```

- a string representing the correct bibliographic citation is put in the slot `@bibliography`

```
> res@bibliography<-"Di Sarli, C. and Troilo, A., 2014.  
+   TRAITs: A new web traitbase for the flora of Argentina.  
+   http://www.pichuco.edu"
```

- `funct_tr` returns the whole object

```
> return(res)
```

2 Adjust `tr8()` function

Then a call to `funct_tr` should be put in `TR8/R/tr8.R` inside the `tr8()` definition (eg. after the call to `it_flowering`) and its results saved into an object:

```
> it_flowering<-get_italian_flowering(species_list,  
+                                     TRAITs=traits_list$Pignatti,rest=rest)  
> my_exp<-funct_tr(species_list,TRAITs=traits_list$New_db,rest=rest)
```

please note that the variables passed as parameters are created within the `tr8` function, thus the call to `funct_tr` should be just like it has been shown here (do not change names).

Then the `my_exp` name should be added to those already existing in in the following cycle (toward the end of `tr8()`):

```

>     for(i in c(eco_traits,biolflor_traits,
+               leda_traits,pignatti_traits,it_flowering,amf_traits)){
+         ## merge the dataframes only if they contain data
+         ...
+     }

```

which should become:

```

>     for(i in c(my_exp,eco_traits,biolflor_traits,
+               leda_traits,pignatti_traits,it_flowering,amf_traits)){
+         ## merge the dataframes only if they contain data
+         ...
+     }

```

3 Adjust other variables

The available traits should be added to the `TR8/data/column_list.R` list; this is a list of vectors in the form of

```
[[code_for_trait]] == c("short_code", "shortdescription", "Traitbase")
```

thus, in this case, we should add there the following lines:

```

> column_list<-list(
+   ## already existing traits
+   ## ...
+   ## ...
+   "height"=c("height","height of a species","New_db"),
+   "dispersal_type"=c("disp_type","Typology of dispersal","New_db"),
+   "clonality"=c("clonality","Type of clonal species","New_db")
+ )

```

This dataframe is used by `tr8` to build the GUI and to pass traits to all the sub-functions which actually retrieve data from remote servers.

Convert `column_list` to a dataframe (retaining the short code, the short description and the database columns) and save it as `available_tr8.Rd` in `TR8/data`:

```

> library(plyr)
> tp<-ldply(column_list)[2:4]
> names(tp)<-c("short_code","description","db")
> save(tp,file="TR8/man/available_tr8.Rd")

```

4 A last remark

This workflow is working for `TR8` version 0.9: the inner work of the package is likely to be changed in future versions, thus users are invited to always check the latest version of the documentation.