An introduction to MAD*traits*

William D. Pearse, Maxwell J.Farrell, Konrad C. Hafen, Mallory A. Hagadorn, Spe

June 11, 2019

Contents

1 Preamble

Currently, installing you can install MAD*traits* by running `devtools::install_github("willpearse/`
Once the MAD*traits* manuscript is accepted somewhere, a new version of `suppdata` and a version of `MADtraits` will be uploaded to CRAN, and you will be able to install by simply running `install.packages("MADtraits")`.

You can get a listing of the functions in the package by typing `library(help=MADtraits)`. If you find any bugs, or have any feature requests for the package, please use the online tracker. Indeed, please contribute to the package using at its GitHub site—help is always welcome!

While *MADtraits* contains much novel code, it relies heavily on the *R* ecosystem. In the development of *MADtraits* we wrote a great deal of code to help with *suppdata* as well.

2 Using MAD*traits*

MAD*traits* is Not A DataBase. Instead, MAD*traits* is a set of code that will download and collate data to which you already have access, and build it into a database for you. The distinction is important: it means that we are not, through MAD*traits*, distributing other people's data, and so you must cite the original data authors when using the data (indeed, we would rather you did so at the expense of citing MAD*traits*, if you are forced to choose).

While that might sound like a lot, getting the data is actually pretty simple.

```
library(MADtraits)                                  # 1
data <- MADtraits(cache="~/Code/MADtraits/cache")   # 2

## Downloading/loading data
## '.' --> 1%; '|' --> 10% complete
## |.........|.........|.........|.........|.........|.........|.........|.........|.

clean.data <- clean.MADtraits(data)                 # 3
clean.data

## A Trait DataBase containing:
##             Species Traits Data-points:
## Numeric       51305   1129      3214840
## Categorical   91326    226       701084
## Total        115554   1355      3915924
## Meta-data present. Units present.
```

The first line loads the MAD*traits* library into your `R` session. The second line does the work of downloading all the data (currently over 110 datasets!) and sorting them into a single `R` object for you to work with. There's something *very important* to note about that second line: the use of the `cache`. By specifying an existing folder for MAD*traits* to use as a cache, MAD*traits* downloads all the datasets to that folder for you. You don't have to use the same location that I use (`/Code/MADtraits/cache`), but by using a location of some sort you ensure that you only have to download all the data once. You want to do this, because it can take a very long time to download allthe data. The third line performs a bit of 'cleaning' on the data—neatening up variable and species names that are obviously intended to be the same thing but are called slightly different things in different datasets. MAD*traits* doesn't do this by default so that you can, if you want, check to see whether the decisions we make are the same that you might make (and, if it's your own data you're downloading, so that you can disagree with us!). However, you will almost certainly find that you want to use the 'cleaned' version of MAD*traits*. If you disagree with some of our decisions, let us know by following the instructions in the section "*Contributing data and/or code to MAD*traits".

```
str(clean.data)

## List of 2
```

```
##  $ numeric    :'data.frame': 3214840 obs. of  6 variables:
##   ..$ species : chr [1:3214840] "abudefduf_vaigiensis" "acanthocybium_solandri" "a
##   ..$ metadata: chr [1:3214840] "id:133;Super_class:osteichthyen;Order:Perciformes
##   ..$ variable: chr [1:3214840] "common_length" "common_length" "common_length" "c
##   ..$ value   : num [1:3214840] 15 150 21.5 32.5 135 125 125 125 175 6.5 ...
##   ..$ units   : chr [1:3214840] "cm" "cm" "cm" "cm" ...
##   ..$ dataset : chr [1:3214840] ".albouy.2015" ".albouy.2015" ".albouy.2015" ".alb
##  $ categorical:'data.frame': 701084 obs. of  6 variables:
##   ..$ species : chr [1:701084] "abudefduf_vaigiensis" "acanthocybium_solandri" "ac
##   ..$ metadata: chr [1:701084] "id:133;Super_class:osteichthyen;Order:Perciformes;
##   ..$ variable: chr [1:701084] "iucn_red_list_category" "iucn_red_list_category" "
##   ..$ value   : chr [1:701084] "np" "LC" "LC" "np" ...
##   ..$ units   : chr [1:701084] NA NA NA NA ...
##   ..$ dataset : chr [1:701084] ".albouy.2015" ".albouy.2015" ".albouy.2015" ".albo
##  - attr(*, "class")= chr "MADtraits"
```

Internally, MAD*traits* stores your data in two `data.frames`: one devoted to the numerical (continuous) data, and the other to the categorical (discrete) data. You can see those two kinds of data in the code snippet above, which shows the `structure` of the data. MAD*traits* stores information about the species, variable, value (of the trait), and meta-data associated with that data (in the format `type of metadata:value`) in a reasonably straightforward way that you can just look at for yourself.

```
subset.data <- clean.data[
    c("quercus_robur","quercus_ilex","quercus_rubra"),
    c("specific_leaf_area","seed_mass")
]
data.frame <- as.data.frame(subset.data)

## Error in tapply(value, list(species, variable), cat.func, ...):  object
'variable' not found

head(data.frame)

##
## 1 function (..., row.names = NULL, check.rows = FALSE, check.names = TRUE,
## 2     fix.empty.names = TRUE, stringsAsFactors = default.stringsAsFactors())
```

```
## 3 {
## 4     data.row.names <- if (check.rows && is.null(row.names))
## 5         function(current, new, i) {
## 6             if (is.character(current))
```

That format of data isn't the most useful for working with the data, so MAD*traits* comes with a convenience function to summarise your data into a `data.frame` where each row represents a single species, and each column a single trait value. If you look at the help file for `as.data.frame.MADtraits` you'll see that it's possible to summarise your data using whatever kind of summary function you wish (it doesn't have to be the mean of the numeric data and the modal value for the categorical data). If you're familiar with functions like `reshape`, you have probably guessed that you can work with the data as stored in the `MADtraits` object itself as well—this is totally fine. This is, in fact, the reason we haven't written a convenience function to work with the meta-data stored within `MADtraits`: we can't think of a good way to summraise meta-data collected over lots of different datasets into a single, useful value that all users would like, and even if we could it would only work for the default values of `as.data.frame.MADtraits`. If you have any better ideas, please let us know!

Notice that, in the snippet above, we subset our data, using the `[species_names , trait_names]` syntax, down to some species and traits that we were interested in. This is important, because there are over three-and-a-half million datapoints within MAD*traits*. If you turn the entire dataset into a `data.frame`, it'll be too big for you to do very much that's useful with it. You can use the `species` and `traits` functions to figure out what's in MAD*traits* if you want.

```
clean.data <- convert.MADtraits.units(subset.data)

## Warning in if (!is_supported_unit(origin)) {:  the condition has length
> 1 and only the first element will be used
## Warning in if (!is_supported_unit(origin)) {:  the condition has length
> 1 and only the first element will be used
## Warning in if (!is_supported_unit(origin)) {:  the condition has length
> 1 and only the first element will be used
## Warning in if (!is_supported_unit(origin)) {:  the condition has length
> 1 and only the first element will be used

clean.data <- lookup.MADtraits.names(subset.data)
```

If you've worked with large datasets before, you've probably noticed that unit and species names can vary across them. The convenience functions `convert.MADtraits.units` and `lookup.MADtraits.names` help with 'cleaning up' units and taxonomic names within MAD*traits*. Give these a try—in particular, you will likely not want to work with data whose units you haven't made the same throughout as otherwise you'll be multiplying apples by oranges!

3 Contributing data and/or code to MAD*traits*

3.1 Contributing data

Have you just published some data? Or know of some published data that others should be using? Great! To get that data into MAD*traits*, you need to do three things: write a function that loads that data, get the citation information for that data, and make a pull request (or send an email) with all of the above.

3.1.1 Write a function that loads that data

```
# Give the function the right name
.pearse.2014 <- function(...){
    # Load the data using suppdata
    data <- read.csv(
        suppdata("10.6084/m9.figshare.979288", 4),
        sep = ",", na.strings = c("","NA")
    )

    # Get the data into the right format
    species <- rep(c("Carcinus_maenas"), nrow(data))
    data <- data.frame(species, data)
    metadata <- data[,c(2:3,8:15)]
    data <- data[,-c(2:3,8:15)]
    units <- c(NA, "mm","#","#")

    # Return the output from .df.melt
    return(.df.melt(data, "species", units=units, metadata=metadata))
}
```

This is the hardest part (sorry!) and it has four parts. The first is giving the function that will download the data the correct name: a `.`, then the first author's surname (`-` should be written as a `_`), a `.`, and then the year of publication. If that function name is already taken, either because there are multiple datasets to be loaded or multiple publications by that author that year, then add `a`, `b`, `c`, etc., after the function name (*e.g.*, `.pearse.2014a`).

The second part is loading the data. Please, please, please use `suppdata` from `suppdata`—all it requires is the DOI of the journal where the paper was published, and either the name of the supplement or its number where the data is published, to download that data. I maintain that function in `suppdata`, so if it's not possible to download data from the journal you prefer *send me an email (`will.pearse@usu.edu`) or make an issue on the MAD*traits *GitHub (see below) and I will fix this for you.*

The third part is getting the data into the right format. You need to make a `data.frame` that contains all the data (the traits), a column with species names, and nothing else. You should also make a separate `data.frame` that contains the meta-data for your dataset (and nothing else), with a column for each separate piece of meta-data. Finally, you should make a vector that contains the units