`macroutils`, an R package for reading, writing, converting and viewing MACRO and SOIL binary files

Julien Moeys
with contributions from Kristian Persson

Swedish University of Agricultural Sciences (SLU)
Center for Chemical Pesticides (CKB)

email: `Julien DOT Moeys AT slu DOT se`

September 23, 2015

Contents

1 Introduction

1.1 What is `macroutils`?

`macroutils` is an R[3] package (library) providing utility functions to work with binary files that are inputted or outputted by the solute transport models SOIL-NDB[2] and MACRO[1].

Main features implemented in this package:

- Read binary files (i.e. import them into R). Batch import is possible;

- Write binary files (i.e. export R tables as binary files that can be read by `SOILNDB` or `MACRO`).

- Convert binary files into text files (CSV or similar formats). Batch converstion is possible;

- Plot some of the variables from one or several binary files as a time series graph;

- View the content of one binary file in a user friendly tabular format.

- Reading, converting, plotting and viewing binary files can be achieved with a *rudimentary* graphical user interface (generally text based), driving the user through a series of options;

1.2 Install and load `macroutils`

1.2.1 Requirements

To install and use `macroutils`, you need:

- An MS Windows computer. The package will most probably also work on Unix (Linux) or Mac computers, but we can not provide support for this, as it requires to build the package from sources (sources are available upon request);

- A recent version of R, already installed on your computer (see below). Check which version of R is required on the package homepage. R is free and open-source.

- The latest version of `macroutils` (see below);

1.2.2 Install R

R[1] installer for Windows can be obtained on R website `http://www.r-project.org/`.

From R homepage, go to the `CRAN`.[2] homepage `http://cran.r-project.org/` or choose the nearest CRAN mirror `http://cran.r-project.org/mirrors.html`.

From there, choose 'Download R for Windows', and then choose 'base' or 'install R for the first time'.

Download the latest R installer, and install it (follow the instructions)[3].

1.2.3 Install the package

Windows 'binary' (i.e. installer) for the latest version of the package `macroutils` can be downloaded on the homepage of the model MACRO on CKB website (or `http://www.slu.se/ckb`, choose English (top-right), 'Area of operations', 'Models', 'Macro 5.2').

The package binary files are usually named `macroutils_x.y.z.zip`, where `x`, `y` and `z` form the version number (major milestone, minor milestone, minor revision).

Notice that you don't need to uncompress this `.zip` file! (it is an installer).

Open R Graphical User Interface, and in the `packages` menu (top menu bar), choose 'Install pacakge(s) from local zip file(s)', and select the package binary file (`macroutils_x.y.z.zip`).

---

[1]R: 'R is a language and environment for statistical computing and graphics'
[2]CRAN: 'Comprehensive R Archive Network', a network of server hosting R installation files and R packages
[3]We don't provide support for R installation!
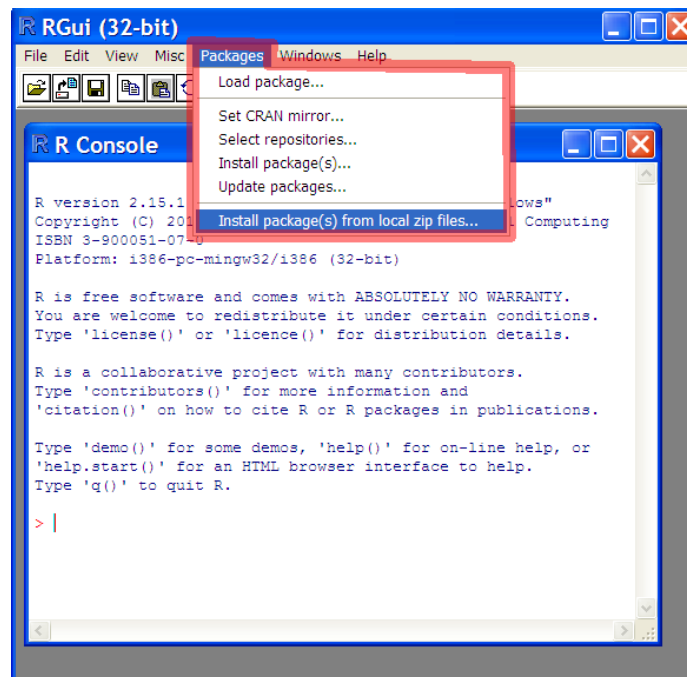
1.2.4 Load the package

After you have started R, just type:

```
library( "macroutils" )
```

to load the package. **Notice that this has to be done everytime you start a new R session**.

1.2.5 Install the package development version from GitHub

Note: If you are not familiar with R developments, you should probably avoid that method, as some complications may arise.

A - On Windows computers, install `Rtools` (see here: `cran.r-project.org/bin/windows/Rtools/`). Install the version that corresponds to your the version of R that is installed on your computer.

On other machines, read the instructions here: `http://cran.r-project.org/web/packages/devtools/README.html` (Windows users may also have a look at this page).

B - Install the package devtools.

`devtools` is a package develloped by Hadley Wickham to install development version of R packages that are hosted on GitHub.

```
install.packages("devtools")
```

C - Optionally, *upgrade* the package `devtools`.

See step 3 in that document: `http://cran.r-project.org/web/packages/devtools/README.html`

D - Install the development version of `macroutils` (GitHub version)

```
devtools::install_github("julienmoeys/macroutils/pkg/macroutils")
```

The GitHub repository of the package `macroutils` can be found here: `https://github.com/julienmoeys/macroutils`.

Notice that it is not guarantee that the development version will compile on all platforms and versions of R.

2   Working with `macroutils`

NOTES:

- The **hashtag symbol(s) #** (or **##** no matter how many) in R code examples mean(s) that the code located on the right hand-side of the hashtag symbol(s), on the same line, **are comments**. They are ignored by R.

- All the examples below are R code. It means you need to open R graphical user interface to run these commands;

- The 'R console' is the area in R GUI where you can type R commands;

2.1   Kickstart (graphical user interface)

Once the package has been installed, to view the content of some binary files, and type:

```
library( "macroutils" )
macroViewBin()
```

and follow the instructions that appear on R console.

To convert some binary files into text files, type:

```
library( "macroutils" )
macroConvertBin()
```

and follow the instructions that appear on R console.

To plot some variables from some binary files, type:

```
library( "macroutils" )
macroPlot()
```

and follow the instructions that appear on R console.

Notice that `library("macroutils")` is only needed once, and thus will not be repeated later in this tutorial!

2.2 Prepare the example files used in this tutorial

Some example bin files are provided with the package. You can view them with
the following code:

```
list.files(
    path = system.file( "bintest", package = "macroutils" ),
    full.names = FALSE, pattern = ".bin", ignore.case = TRUE )
[1] "defaultrun.bin" "Fert.bin"        "METFILE.BIN"
[4] "RAINFALL.BIN"   "soiln001.BIN"
```

For this tutorial, we will only use two of these files. We will first copy them
in the working directory, in order to avoid to alter them accidentally:

```
## File name, without the library path
filenm <- c( "METFILE.BIN", "RAINFALL.BIN" )
##
## Find the library path
path <- system.file( "bintest", package = "macroutils" )
##
## Copy these files in the working directory
file.copy(
    from       = file.path( path, filenm ),
    to         = file.path( getwd(), filenm ),
    overwrite  = FALSE  )
[1] TRUE TRUE
```

So you now have two binary files in your working directory:

```
file.exists( filenm )
[1] TRUE TRUE
```

2.3 Note on R working directory and file paths

R working directory is the place where R looks for input files or writes output
files, when no path is specified.

You can use `setwd` to change the working directory. Read its help page
(`?setwd`).

Do not forget that, contrary to windows default behaviour, R does not accept
file path with single backslash (\), as this is a special character for R. Use double
backslash instead (\\), or single slash (/), or double slashes (//). This is also
valid for file paths.

2.4 Convert binary files into text files (`macroConvertBin`)

2.4.1 Graphical user interface

Simply type:

```
macroConvertBin()
```

... and follow the instructions that appear on the screen, first as a pop-up window to choose the file(s) you want to convert, and then as a of text menu asking you some details about the convertion (symbols used as field separator and as decimal mark).

Unless you set the argument `fileOut`, the output file name(s) and extension(s) will be chosen automatically, and the file(s) will be written in the working directory.

If the file(s) already exists, you will be asked if you want to overwrite them or not.

It is possible to convert several files at a time with this function.

2.4.2 R commands (non-interactive mode)

If you want to convert `METFILE.BIN` using R commands, type:

```
macroConvertBin( file = "METFILE.BIN", gui = FALSE )
```

You can customise the file format, for example:

```
macroConvertBin(
    file   = "METFILE.BIN",
    gui    = FALSE,            # No Graphical Interface
    sep    = ";",             # Field separator
    dec    = ".",             # Decimal mark
    fileOut = "METFILE2.CSV"   # Output file
)
```

Notice that when used non-interactively, `macroConvertBin` overwrites existing output files without warning!

2.4.3 Batch conversion of multiple files

The example below should how to (a) list all the bin files in a folder, (b) convert these bin files to `CSV` files and (c) export them in textitanother folder.

```
#   Location of the directory where the files
#   are located. Here I take the default bin files in
#   the package installation directory. Just write the
#   "path/to/your/own/directory" instead
fDir <- system.file( "bintest", package = "macroutils" )
#   Location of the directory where the files should
#   be output. Here I the working directory
dirOut <- getwd()
#   List the bin files in fDir
f <- list.files( path = fDir, pattern = ".bin",
    ignore.case = TRUE, full.names = FALSE )
#   Excluding a file that raise a warning
```

```
#   (irregular time series)
f <- f[ f != "Fert.bin" ]
#   Contruct path and name for output files
fOut <- file.path( dirOut, sprintf( "%s.csv", f ) )
#   Add the path to the input files
f <- file.path( fDir, f )
#   Convert the files
macroConvertBin( file = f, fileOut = fOut, gui = FALSE,
    overwrite = TRUE )
```

Notice that if `fileOut` is not set, `macroConvertBin` will simply output the CSV files into the same folder as the input files, and just add an additional `.csv` extension to their original name.

2.5   Import binary files into R (`macroReadBin`)

2.5.1   Graphical user interface

If you want to import one or several binary files into R (as `data.frame`), simply type:

```
tbl <- macroReadBin()
```

And the binary file will be read and saved into the R object `tbl` (you can choose any name you like, if it is not an existing R function or reserved character).

If you have chosen one file, `tbl` will be a single `data.frame`. If you have chosen several file, `tbl` will be a `list` of `data.frame`s. Each table in that list can then be retrieved individually, for example:

```
tmp[[ 1 ]]
```

will fetch the first table in the list. Notice this won't work if you have chosen only one file.

Check `?data.frame` and `?"[.data.frame"` to understand what `data.frame`s are and how they can be subset (out of the scope of this tutorial).

2.5.2   R commands (non-interactive mode)

In non-interactive mode, you can type:

```
tbl <- macroReadBin( file = "RAINFALL.BIN" )
```

and then, for example, inspect this table:

```
## Class:
class( tbl )
[1] "macroTimeSeries" "data.frame"
##
## First rows:
head( tbl )
```

```
              Date rainfall
1 1987-01-01 12:00:00     2.6
2 1987-01-02 12:00:00     0.1
3 1987-01-03 12:00:00     0.1
4 1987-01-04 12:00:00     1.5
5 1987-01-05 12:00:00     3.2
6 1987-01-06 12:00:00     0.0
##
## Last rows:
tail( tbl )
                Date rainfall
726 1988-12-26 12:00:00     0.0
727 1988-12-27 12:00:00     2.1
728 1988-12-28 12:00:00     0.0
729 1988-12-29 12:00:00     0.0
730 1988-12-30 12:00:00     0.0
731 1988-12-31 12:00:00     0.0
##
## Dimentions:
dim( tbl )
[1] 731   2
##
## Column names:
colnames( tbl )
[1] "Date"    "rainfall"
```

Additional parameters can be changed via `?muPar` to modify the way binary files are processed and how column names are 'sanitised'. The default behaviour is now to replace sucessive spaces by single underscore, to remove non-alphanumeric symbols and to remove the simulation ID if it is present. For example:

```
#   Set 'removeSpace' argument
muPar( removeSpace = FALSE )
#
#   Read the binary file
tbl <- macroReadBin( file = "METFILE.BIN" )
#
#   Look at column names (notice the spaces):
colnames( tbl )
[1] "Date"
[2] "Max air temperature oC    Sutton Bonington"
[3] "Min air temperature oC    Sutton Bonington"
[4] "Solar radiation    W m2   Sutton Bonington   TRS1"
[5] "Vapour pressure    kPa    Sutton Bonington   TRS1"
[6] "Wind speed         m s1   Sutton Bonington   TRS1"
#
#   Reset default arguments
muPar( reset = TRUE )
```

2.6 Plot variables from binary files (`macroPlot`)

2.6.1 Graphical user interface

Simply type:

```
macroPlot()
```

... and follow the instructions that appear on the screen, first as a pop-up window to choose the file(s) you want to plot, and then as a of text menu asking you some details about the plot you want to produce:

- Which variables you want to plot. This is asked for each binary file you have chosen. You can choose several of them;

- If you want all the variables in one single plot, or in distinc sub-plots.

- If you want to zoom-in (or zoom-out), to view a narrower time window in the time-series.

- These questions are asked over and over, until you terminate the function (or press the `ESC` key).
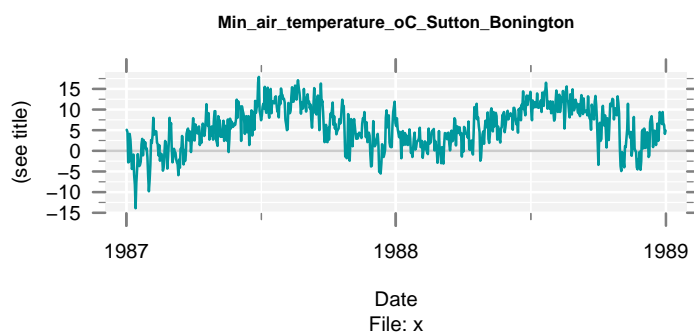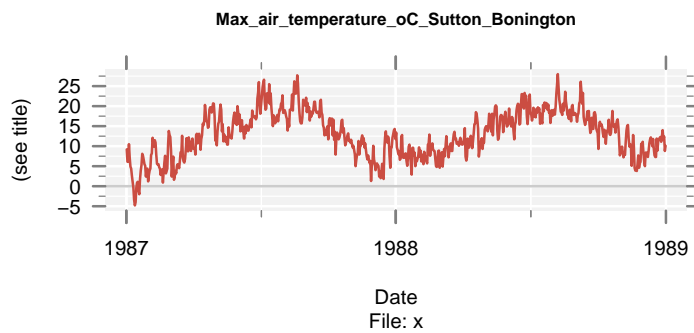
2.6.2 R commands (non-interactive mode)

If you want to use `macroPlot` in a non-interactive way, it is necessary to import some binary file(s) first:

```
tbl <- macroReadBin( file = "METFILE.BIN" )
##
## Look at it:
colnames( tbl )
[1] "Date"
[2] "Max_air_temperature_oC_Sutton_Bonington"
[3] "Min_air_temperature_oC_Sutton_Bonington"
[4] "Solar_radiation_W_m_2_Sutton_Bonington_TRS1"
[5] "Vapour_pressure_kPa_Sutton_Bonington_TRS1"
[6] "Wind_speed_m_s_1_Sutton_Bonington_TRS1"
dim( tbl )
[1] 731   6
```

Now the data can be plotted. We will choose only the first two variables here (+ the Date column):

```
macroPlot( x = tbl[, 1:3 ], gui = FALSE )
```

Max_air_temperature_oC_Sutton_Bonington

Date
File: x

Min_air_temperature_oC_Sutton_Bonington

Date
File: x

Another way to look at the same dataset:

```
macroPlot( x = tbl[, 1:3 ], gui = FALSE, subPlot = FALSE )
```

File(s) and Variable(s):
— x, Max_air_temperature_oC_Sutton_Bonington
– – x, Min_air_temperature_oC_Sutton_Bonington

Date

11
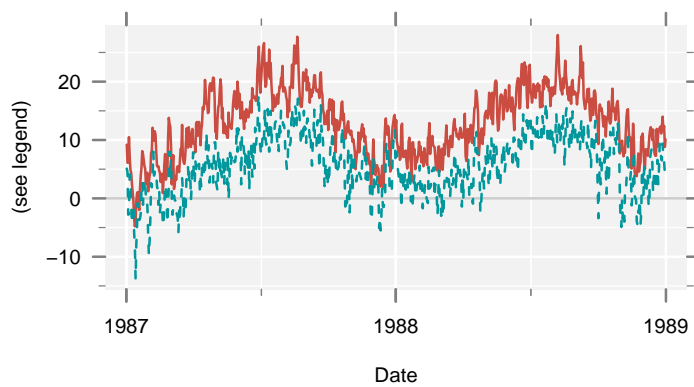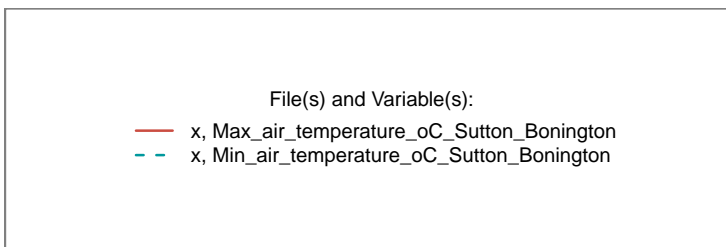
2.7 Binary files visualiser, as tabular data (`macroViewBin`)
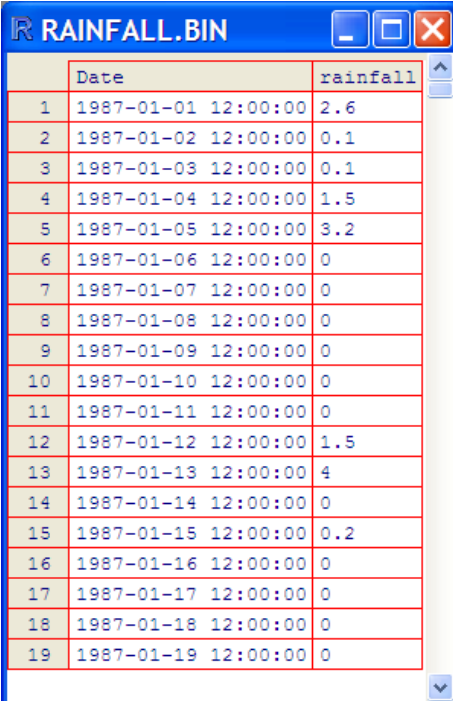
2.7.1 Graphical user interface

`macroutils` also includes a rudimentary binary file viewer. It is a wrapper around R function `View`, wich includes a file chooser.

```
macroViewBin()
```

You may also type:

```
macroViewBin( "RAINFALL.BIN" )
```

Next figure show what the viewer looks like.

	Date	rainfall
1	1987-01-01 12:00:00	2.6
2	1987-01-02 12:00:00	0.1
3	1987-01-03 12:00:00	0.1
4	1987-01-04 12:00:00	1.5
5	1987-01-05 12:00:00	3.2
6	1987-01-06 12:00:00	0
7	1987-01-07 12:00:00	0
8	1987-01-08 12:00:00	0
9	1987-01-09 12:00:00	0
10	1987-01-10 12:00:00	0
11	1987-01-11 12:00:00	0
12	1987-01-12 12:00:00	1.5
13	1987-01-13 12:00:00	4
14	1987-01-14 12:00:00	0
15	1987-01-15 12:00:00	0.2
16	1987-01-16 12:00:00	0
17	1987-01-17 12:00:00	0
18	1987-01-18 12:00:00	0
19	1987-01-19 12:00:00	0

2.8 Export R tables as binary files (`macroWriteBin`)

TO BE WRITTEN. Check `?macroWriteBin`, it presents some examples.

Notice that this function has not been tested extensively yet (i.e. we have not tried to run MACRO with files written with this function!).

2.9 Time aggregation of data imported from a binary file (`macroAggregateBin`)

TO BE WRITTEN. Check `?macroAggregateBin`, it presents some examples.

2.10 Reading the functions' help pages

The functions' in this package are documented quite extensively. You can access the help pages menu of this package by typing the command:

```
help( package = "macroutils" )
```

NB: This will open a help page in a web-browser.

To view the help page of a specific function, for example `macroReadBin`, type:

```
help( "macroReadBin" )
# OR
?macroReadBin
```

Just change the function's name to view the help page of another function.

Experienced R users may also have a look at the actual code executed by each function by typing their names, without the parenthesis. For (a short) example:

```
getMuPar
```

The `macroutils:::getMuPar` method is mostly useful for functions that are hidden from the end-user (often names starting with a `.` are hidden).

3 Miscellaneous

3.1 Credits and License

`macroutils` (and this documentation) is licensed under an Affero GNU General Public License Version 3.

This package and this document is provided at no cost, with NO responsibilities, guarantees from the author or his employer (SLU and CKB).

3.2 Bugs and improvements

Please report us any bug you may find in this package. When reporting bugs, please:

- Make sure the bug is occuring in `soilmacoutils`, and is not due to errors in your `R` code or due to a wrong usage of the package (typically a wrong object class passed to a `soilmacoutils` function).

- Read the function's documentation before reporting any problem;

- Make sure the file you are trying to read is located in the working directory (see `getwd()`), unless the file path is provided with the file name.

- Send us the binary file that cause the problem, or any binary file on which the problem occur (or a sub-part of it).

- Send us a concise **reproducible code example** that display the problem you are facing. That should use the above mentioned binary file you send to us. A reproducible code example is a code sniplet that can be executed on any computer with the same R and the same `soilmacoutils` version.

- As in the sub-section below ("Session info"), provide us you `session-Info()` and `soilmacoutils` version.

- If needed, find help from a local R expert at your workplace to provide such reproducible example.

A reproducible example is mandatory for an bug-fix demand, as we need to be able to exactly reproduce the problem ourselves to be able to fix the problem.

If you notice any error or approximation in this manual, comments are also very welcome.

Suggestions of improvements are very welcome, but keep in mid they may only be implemented if they prove useful to many users and if we have time and ressources to implement them.

3.3 Session info

Information on R Session and packages versions (that were used to build this vignette):

```
sessionInfo()
R version 3.2.0 (2015-04-16)
Platform: i386-w64-mingw32/i386 (32-bit)
Running under: Windows 7 x64 (build 7601) Service Pack 1

locale:
[1] LC_COLLATE=Swedish_Sweden.1252
[2] LC_CTYPE=Swedish_Sweden.1252
[3] LC_MONETARY=Swedish_Sweden.1252
[4] LC_NUMERIC=C
[5] LC_TIME=Swedish_Sweden.1252

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods
[7] base

other attached packages:
[1] macroutils_1.8.4

loaded via a namespace (and not attached):
[1] tools_3.2.0 tcltk_3.2.0
## packageVersion( pkg = "macroutils" )
```

References

[1] Nick Jarvis and Mats Larsbo. Macro (v5.2): Model use, calibration, and validation. *Transactions of the ASABE*, 55(4):1413–1423, 2012.

[2] Holger Johnsson, Martin Larsson, Kristina Mårtensson, and Markus Hoffmann. Soilndb: a decision support tool for assessing nitrogen leaching losses from arable land. *Environmental Modelling & Software*, 17:505–517, 2002.

[3] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.