Geosci. Model Dev. Discuss., 8, C846–C849, 2015
www.geosci-model-dev-discuss.net/8/C846/2015/

Geoscientific Model Development Discussions

Interactive comment on "An open and extensible framework for spatially explicit land use change modelling in R: the lulccR package (0.1.0)" *by* S. Moulds et al.

E. Pebesma (Referee)

edzer.pebesma@uni-muenster.de

Received and published: 20 May 2015

Review of: **An open and extensible framework for spatially-explicit land use change modelling in R: the lulccR package (0.1.0)**, by S. Moulds et al.; MS No.: gmd-2015-48. Reviewer: Edzer Pebesma; 20 May 2015.

I congratulate the authors with a very readable paper on land use change modelling and how this can be done transparently and reproducibly with a package they developed for R. Not only does it describe and introduce the software well, it also gives a very extensive literature review to modern environmental modelling paradigms as well

as land use change modelling approaches. As I am not an expert in the area of land use change modelling, in this review I will focus on the software development side, and whether the paper reaches the aim of empowering land use change modellers, and inviting them to take the modelling process in their own hands.

For a paper introducing a software framework, it is extremely extensive on describing land use change modelling, but extremely thin on describing the software. The class diagram is offered in a UML diagram that gives only main classes and core functions; methods are not even mentioned. Will the land use modeller be helped by this, and be invited to understand it, use it and extend it? The least the authors need to do is explaining the arrow types and symbols in the UML. A table with all methods and key function offered would also be helpful, as these are the things a user will need first.

The paper also needs to be more explicit about which users it wants to attract, and serve. Should the package users be fluent with the packages sp, raster, caret, rgdal, and maybe more? Or should more novice R users also feel invited? The current examples, which should be the package advertisement, contain constructs like `raster::extract`, `obs@maps[[1]]`, save a plot object to `p` to later plot it with `print(p)` – all constructs that will scare novice users, and that may not be necessary.

After the first action (p 11, l 11), I was surprised that (i) `pie` is a list that is available in the package (and not e.g. a data object loaded by `data(pie)`), (ii) that the object created is nothing but a RasterStack, the categories, their names, and a set of times. Why not create a RasterStack that holds all this information? In that case, `ObsLulcMaps` could simply extend `RasterStack`, and would get all its methods (like `plot`!!) for free.

In figure 1, it is not clear from the caption what t0, t1 and t2 refer to, and why either LULCC (t1-t0) or LULC (t1) can be input.

I am not very fond of R packages with an R in them. In this case, it leads to expressions like "the lulccR R package", "the lulccR package for R", or "the lulccR package [...]

written in the R programming language." (abstract), which are all odd. I would suggest to rename it into `lulcc`, and start sentences now starting with *lulccR* with "The *lulcc* R package ..."

The package is currently found on GitHub. Why has it not been submitted to CRAN? Submitting to CRAN offers easier accesibility (and allows to remove the now very odd lines 9-10 on page 11), quality control, and archiving by a third party: the current version 0.1.0 may change any moment. Also, a version number should be removed from the title, but it would be appropriate to have the paper correspond to a 1.0 version on CRAN, indicating the author's opinion that it is mature enough to be published. Did the package get any use by others that the authors can report on, e.g. in papers published?

The classes provided by lulccR seem to be useful, but why does the package not come with methods that users expect? After creating `obs`, I tried `plot(obs)`, `summary(obs)`, but none of them were worked. Now, users not only need to type the long `AgreementBudget.plot`, and `FigureOfMerit.plot`, but they also need to memorize it, instead of simply using `plot`. This needs to be simplified; similar to saving the plot, then printing it, or only saving it (p 22, l 17,19).

The discussion about memory footprint and the caching that `raster` does is relevant, but it would also be good to mention which dimensions a model still can have on e.g. a 4 Gb RAM machine. Programmers often forget how large and cheap RAM is, these days.

Smaller items:

1. P 10, l 22: "a raster object belonging to the raster package": rephrase.
2. p 7 l 17: add to this sentence: "be expressed programmatically and be communicated as such with reasonable effort".
3. p 11, l8,9: omit; put installation instructions in the Code availability section.

4. Title: version should not be necessary. Suggest: *An open and extensible framework for spatially explicit land use change modelling: the* `lulcc` *R package*
5. Under R 3.2.0, the source package passes R CMD check with only one (easy to resolve) NOTE.
6. page 14, l2: underscores are escaped; correct.
7. page 15, l 12: avoid R comments, but use regular text.
8. Why is Performance.plot not simply called plot?
9. How can the current script of the paper be run from within lulccR? Explain in "Code availability".
10. are all figures reproduced by the scripts in this document? Or is there a demo script in lulcc that reproduces all figures in this paper?
11. p 22, l 17 and 19: these expressions do not show a plot – why not call the method "plot", and show the plot instead of saving the plotting object here?

Interactive comment on Geosci. Model Dev. Discuss., 8, 3359, 2015.