Hydrol. Earth Syst. Sci. Discuss.,
https://doi.org/10.5194/hess-2019-50-AC3, 2019

Hydrology and Earth System Sciences
Discussions

Interactive comment on "Using R in hydrology: a review of recent developments and future directions" *by* Louise J. Slater et al.

Louise J. Slater et al.

louise.slater@ouce.ox.ac.uk

Received and published: 13 April 2019

We would like to thank Michael Stoelze for this constructive review and for his positive comments on our manuscript. We look forward to implementing these suggestions which will certainly improve the paper. Below we provide Michael's comments verbatim in italic black text.

First of all, I really support the idea of the paper and I think the presented review is fully in the scope of HESS. I like how the relationship between hydrology and R is presented and the authors gave a broad overview of historical and recent developments in the RHydro community and sketch a interesting roadmap for the future of R in hydrology.

The paper is from my perspective also a good example for a joint effort from young scientists in our hydrological community. However, since the first two reviews are full of praise I try to be a little bit more critical putting together some thoughts that hopefully improve the paper. My major "overall" suggestion is to revise the paper going a little bit away from just having a list of available packages or presenting potential possibilities in the R-Hydro universe towards a) more "best practices" for certain challenges and b) more comparisons of similar packages. However, I absolutely recommend to publish this paper in HESS. See more specific comments below.

Best regards, Michael

Thank you for these positive comments on our manuscript. We are pleased to see this work described as a "good example of joint effort from young scientists in the community", and we appreciate the thoughtfulness in these constructive suggestions. We will certainly revise the manuscript by emphasizing best practice and comparison of package functionalities, as suggested.

Major comments

The authors give a lot of technical information about R in different section in the paper (e.g. GNU S background in sect 2.3, R script workflow in Sect. 2.1, R output format in the Introduction). As all the parts are really important to learn about the fundamental principles of R for me it would be more helpful to have the technical and historical information about R in one separate section. This might be also helpful for readers who are new to R coming from other programming languages.

We agree it would be a good idea to move some of these elements into one separate paragraph in the introduction. However we feel that creating an entire section on the history of R might be a little misplaced, so instead we suggest to include a few general references in the introductory paragraph, such as (for example) https://cran.

r-project.org/doc/manuals/r-release/R-lang.html or https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-R_003f

I miss more discussion about different philosophies how to program in R. From my perspective there is, for example, often the question whether you are doing your visualizations with ggplot2 or in base-R or doing data wrangling with dplyr or in a complete different "style" with data.table. I don't know if there are comparable examples for hydrological packages (doing more or less the same thing but with different techniques / packages). The paper could, however, be improved by presenting opposed approaches to solve problems in R. For example, data.table is known to be very fast in data wrangling but on the other side the dplyr package (and piping %>%) is known to support highly readable code. Here the trade-off between performance and communication could be discussed (i.e. Sect 4.3 and Sect. 2.1/2.2). My experience from teaching R in hydrology is that students really like the concept of piping (%>% read as "and then do that") and can do the step from base-R to tidyverse in a couple of sessions. Tidyverse and Pipes could also be mentioned in Sect. 5.3 as a good possibility to teach "highly readable code" in the classroom.

We agree that the different approaches to R programming (e.g. base-R versus the tidyverse) are definitely worth mentioning. It is often difficult to decide between different packages that offer similar options (e.g. many different packages for parallel computing). Regarding which approach is easiest for beginners, we note that while the tidyverse style might seem easier at first, as soon as the analysis becomes more complex, dplyr on his own won't suffice. For example, to handle out-of memory dplyr needs to be connected to a database, or you need to switch to bigmemory/data.table. Thus in many cases, using base-R is safer (e.g. for operational purposes, or for creating and maintaining packages) as it avoids issues with dependencies and updates. Overall, we entirely agree that it is worth discussing these issues in the manuscript and providing a more explicit comparison of the different approaches - thank you for raising this point!

Regarding Sect 2.2 it would be helpful to draft a way forward how developers of R-Hydro packages can develop better R packages with a clear description of input data types, a comprehensive documentation on package usage. For example, tidyverse is set of packages that work well together as all packages use the same data representations and API design. Is this also a valuable approach for, let's say, developer of R packages that include hydrological models? For example, is it possible to use the data retrieval packages from Table 1 in a coherent way or is the syntax between all packages different? Is there an effort to combine data retrieval packages from different regions with a consistent R syntax/usage? Using different hydrological R packages with the same syntax might be of great value for the community. A short discussion about this issue can draft a way forward in the R Hydrology community (as mentioned in Sect 4.1 where the need of good vignettes in R is highlighted).

Yes, we agree with the points made here. The existing hydrological data retrieval packages (e.g. rnrfa, dataRetrieval...) are structured differently because they were set up independently, and because the underlying hydrological and hydrometric datasets differ from country to country. It is therefore difficult to coordinate hydrological data amongst regions. As more nations develop similar hydrological data acquisition packages in the future, we believe it would be worth implementing a common syntax and data output form. For example, it would be ideal to use consistent APIs and consistent output objects across packages. We will make some recommendations for future developers of these hydrological packages in the manuscript. Additionally, it is worth mentioning that there is currently no effort to combine data retrieval packages from different regions but this is certainly worth doing. It would be worth implementing for example, (i) a meta-package with functions that convert hydrometric data from other packages to a standard format, or (ii) a meta-package for all hydrological models to be run within the same framework.

The introduction of section 3 and also Fig. 3 are really nice and well-conceived. I like the idea to split the data analysis into different parts and subsections in the paper (Sect. 3.x).

Thank you!

In general, the paper is full of valuable package, links, and corresponding citations. However, from my perspective often only the package names are given and more detailed information how the package is working or why it is especially useful/important for the hydrological community is missing. Of course, it is not possible to describe all the functionality of all the packages in detail but the authors could pick 2-3 cases where they really compare the advantages and disadvantages of using different packages for the same purpose / challenge. To be honest (and I also think that the author team is aware of that), you can relatively easy check on the internet if there is a package for a certain task and what are the technical details or data requirements of a single package. So, the great list of available packages cannot be the main message of this compelling paper! Comparing different packages among each other could really be of great value for the hydrological community and this paper is a good place to do that. A good example is Sect 3.4 describing spatial data analysis and visualization in R. In this section a lot of important R packages are mentioned but the reader gets less information which package is better or has the same functionality as other packages etc. In Sect 3.5 the information about snow functionality in the hydrological models is a good example how this can be done.

Yes, it is true that in some cases we were quite brief - there is so much to say that this paper could be written as a book! However, we agree that it is worth comparing the advantages/disadvantages of similar packages when we revise the manuscript. We will provide some information about advantages of specific packages within the different sections, similarly to the snow functionality example.

For me a major deficiency of the paper is that the topic "colours" is completely missing. Hydrology is a multi-facetted discipline and hydrologists doing a lot of visualizations for posters, presentations and papers. Hydrology is also often based on multi-dimensional analyses with different variables in space and time. To visualize hydrological processes and hydrological change colour is often the first choice to compare data sets and to investigate relationships. As we all know, a sophisticated colour choice is often a huge challenge. I encourage the authors to implement a "colour section" where a short discussion about, for example, the basics of appropriate colour gradients for graphs and maps is given (one-colour gradient, two-colours gradient, discrete vs. continuous variables). In large-scale hydrology "rainbow" colour gradients are still state-of-the-art (to generate fancy and colorful world maps) but a lot of papers and blogpost have taught us that the "rainbow" isn't perceptual uniform and not colourblind-safe.

Indeed, there is a lot to say about data visualization - we agree this would be a useful addition. It is increasingly accepted that the rainbow color scheme is a poor choice for data visualization (e.g. https://betterfigures.org/, https://www.climate-lab-book.ac.uk/2014/end-of-the-rainbow/, or a recent R-bloggers post on this topic - https://www.r-bloggers.com/at-the-end-of-the-rainbow/). We will mention that R is strong in this department and will include a short section about colours, color-blind friendly palettes, and appropriate choices in hydrology. We will also mention some useful papers on this subject, e.g. Kelleher and Wagener (2011).

Minor comments

P2L20 Give 1-2 examples of hydrological models coded in R.

Yes, we will do this.

P2L33-35 It might be helpful for the reader to compare the fundamental principles of R against other programming languages (e.g C, Java) to better describe the architure of R (e.g. in terms of object-orientantion, complilation of code

Yes, we will include a short description of R's principles (as an object-oriented programming and interpreted language) in the technical section mentioned above.

P3L4 Are there any reasons for the large number of updates in R-Hydro packages (related to Fig.1)?

We actually had a lengthy discussion about this among ourselves. One possibility is that the increase in updates in 2018 is related to documentation requirements on CRAN, as a couple of features that were okay before had to be modified (possibly due to package dependencies). Alternatively it is possible that the rise in package updates was linked to the development and uptake of R-HUB (a web service that allows developers to test and debug R-packages on different operating systems to reproduce what CRAN does). We were not entirely sure of the answer and so we refrained from mentioning this in the manuscript.

P4L11 The platform Stackoverflow (from Sect. 2.5) can also be mentioned here as there often also information on relevant R packages are given.

Yes, we will do this.

P5L31 What is meant with the "standardized format"? Here it might be also valuable to add packages like roxygen2 that help to develop, create and document R packages.

The sentence referred to is *"Relying on well-established publishing platforms such as CRAN and GitHub has promoted a standardized format for developing and dissemi-*

nating R code." Here, we were referring to the best practice in writing R code, and will clarify this in the text. We will include the roxygen2 package.

P6L27 It will be easy to find some justification for this statement (e.g. number of R topics on stackoverflow.com compared to other programming languages).

The statement referred to is *"R is still considered the most powerful language and environment for statistical computing"*. We will justify the statement in our revised manuscript.

P6L3-14 The paper would be improved by adding various possibilities to share and publish code during and after paper publication (e.g. github repository, gists or other tools and platforms). Is there a possibility to share R code with a DOI?

We agree, this is a good idea. We will mention different options and platforms for sharing R code with a DOI such as GitHub, Zenodo, Figshare, and RPubs/Plotly (also for dashboards and interactive plots). Another important data repository to mention is https://www.pangaea.de/, because it is tailored to Earth and Environmental Sciences. Some of these platforms are discussed in the journal *Geoscientific Model Development* https://www.geoscientific-model-development.net/about/code_and_data_policy.html

Sect 2.3 For me this section contributes less to the strength of the paper – what is the key message here? The specific statistical packages that are relevant for R-Hydro are given in Sect 3.6 (as mentioned). The link to CRAN is perhaps too unspecific here. Yes, R provides a vast number of statistical tools for hydrologists but then you have to explain in more detail why and what advantages R here has to offer (compared to other languages?).

Yes, there is indeed some overlap between sections. We will remove this opening section (*2.3. Providing statistical tools for hydrology*) and strengthen section 3.6 (*Packages for hydrological statistics*) instead. We will move the description of CRAN to the new suggested section that provides a background on R.

Sect 2.5 What are the rotating topics and why have they been chosen? The picture might be nice to see but the space could instead better be used to highlight the EHGU SC topics from the last years with a short information why the topics were of great interest for the community. It might be also important to highlight the possibility to raise issues for certain R packages on github.com to foster an online-documented discussion with the package developers (i.e. interaction in the community). At least a cross reference to Sect. 5.4 could be made here.

The section Michael is referring to (Section 2.5.) is the section on *scientific resources and courses*. We will keep the picture and will include a list of the topics too, explaining why they are of interest. We also agree that it is worth highlighting in this section the ability to raise issues with the developers on GitHub (provided packages are developed/published on GitHub). We will encourage publication of packages on GitHub and will mention that the bugs report link can be added in the Description file so that package authors can specify how they prefer to receive bug reports.

Table 1 I am not sure on which basis you have choose the 11 packages? Are they the most important ones? Perhaps a thematic order would improve the table (sort by data type, spatial extend, multiple proposes or not,. . .)

It is difficult to be completely exhaustive but we will do our best and will point readers to the Task View for other packages. We also agree that ordering these packages by data type (and perhaps providing a bit more information about the data) would be

worthwhile.

P11L5-9 Consider to mention the rio package here as it is really powerful. The tidyverse package is cited as a package that reads multiple file formats – I think this is not the main purpose of the tidyverse package.

Thank you for the suggestion about the `rio` package. We will indeed rephrase our description of `tidyverse`!

P11L10 Which netcdf package is the best? Or let's say, are there more information which package to use for a specific application.

We will discuss which package is most appropriate for different uses.

P12L15 For all ggplot2 users it is really important to mention that the ggplot2 3.0 version offers support to visualize sf objects directly with a specific geom (geom_sf). So, if one is familiar with ggplot2 and want to visualize spatial data (raster, shapefiles etc.) this is a really easy way to do that? As the ggplot2 community is growing and growing this might be a really important information for some readers of the paper.

This is a great idea; thank you for the suggestion.

Table 2 Should be included in Sect 3.5

Yes; we will rectify this.

Sect. 3.6 Package "skimr" might be valuable for this section.

Good idea, thank you!

P15L10 I agree, one strength of ggplot2 and faceting is the ability to generate multiple views. This is especially important to avoid overplotting or – as the authors mentioned – to split a data set in different plots using meta information of this data set (i.e. season). If you have a nice example for faceting here would be a good place to show this principle for the readers.

We do have some good examples and will include one, thank you!

P19L17 Is it true that publishing a R package on CRAN is free of quality checks? However, a short comment on the differences during package publication (CRAN vs.GitHub) might be helpful here.

In our experience, CRAN only checks for code and documentation consistency, not code logic nor quality/coverage (CRAN does not run linter checks or test coverage). The point about the differences during package publication is important and we will discuss these briefly in the manuscript.

P19L21-26 I agree but the paragraph could be more specific on how we should develop R packages. Is there a good article or blogpost describing the progress of publishing a R package? What are the requirements to generate good tutorials or readme files? Might it be useful if journals like HESS recommend to publish paper code on github?

We will include (1) a description of the process, or provide a clear, straightforward reference on publishing packages; and (2) a description of what a useful tutorial/readme should include for hydrology. Some references for package building include, for example: (i) a popular minimal description of how to write a package https://hilaryparker.

com/2014/04/29/writing-an-r-package-from-scratch/, (ii) helpful tutorials such as https://kbroman.org/pkg_primer/ or a more tidyverse oriented reference with a video: https://www.rstudio.com/resources/videos/you-can-make-a-package-in-20-minutes/. We will also suggest that journals should encourage authors to publish their code. For example, the journal *Geoscientific Model Development* already does this: https://www.geoscientific-model-development.net/about/code_and_data_policy.html.

Thank you for the helpful suggestions.

References

Kelleher, C. and Wagener, T.: Ten guidelines for effective data visualization in scientific publications, Environmental Modelling Software,26, 822–827, 2011.

Interactive comment on Hydrol. Earth Syst. Sci. Discuss., https://doi.org/10.5194/hess-2019-50, 2019.