

Geosci. Model Dev. Discuss., referee comment RC2
<https://doi.org/10.5194/gmd-2021-336-RC2>, 2022
© Author(s) 2022. This work is distributed under
the Creative Commons Attribution 4.0 License.

Comment on gmd-2021-336

Anonymous Referee #2

Referee comment on "RavenR v2.1.4: an open-source R package to support flexible hydrologic modelling" by Robert Chlumsky et al., Geosci. Model Dev. Discuss., <https://doi.org/10.5194/gmd-2021-336-RC2>, 2022

This manuscript provides a description of a set of R functions to process in- and output files and the contained data, for the purpose of running the hydrological modelling software Raven, which itself is available as a C++ executable.

The manuscript is generally well-presented and well written, and I have very few specific comments. However, I make the following more general observations:

- I am actually not sure whether the manuscript fits any of the designated manuscript type. I suppose that it is classified as a development paper because of the ample references to reproducibility in the manuscript. However, the original model is available as an open source code as well and therefore perfectly reproducible (at least in the sense that it is described on the about GMD web page). So at most, it is an enhancement of the usability of a specific model rather than its reproducibility.

- The other reviewer has made some very useful comments on the presentation, with which I fully agree. Overall, I think that the manuscript is too wordy and can be reduced substantially. Specifically, the authors seem to be at pains to convince the reader about the importance of open source software, accessibility, and good practices in model development. I don't think that the GMD readership needs such advocacy. It distracts from the core message and makes the manuscript unnecessarily long and somewhat tedious to read. (For example, the section L.137 - 146 is quite trivial and may be deleted entirely, but also many other sections can be streamlined).

- The technical implementation of the package is quite straightforward, and does not make optimal use of advanced functionality of R. Specifically:

- The fact that the model needs to be run separately is not very elegant. It would be ideal if the Raven model itself is distributed with the package as a dynamic library, and can be loaded as such by the R process. This would avoid the need for separate installation of the model, as well as the slightly clunky way that the executable is called by the `rvn_run()` function. It would also help with the next point.
- The fact that the scripts writes the input files to disk, which are then subsequently read by the executable (and vice versa for the output files) is inelegant at least, and probably also inefficient as well. If the model itself were implemented as a dynamic library then the in- and output data could be passed in memory to the model, which would greatly enhance performance in use cases such as monte carlo simulation.
- The package makes relative limited use of the object oriented nature of R. It does use relevant classes such as `xts` and `lubridate`, but does not define any classes itself. This results in a very long list of functions, essentially one function for every step in the analysis. It would be much more elegant (and efficient) to define a set of classes (e.g., one for each in- and output file, by extending classes such as `xts`) and then use method dispatch to read and write them, as well as any other standard processing such as aggregation. This would reduce the need for a long list of different functions to a few `read()` and `write()` commands, and allow for method dispatch on existing `xts` functions.

Lastly, while the examples in the manuscript are generally easily reproducible, some of the examples in the online documentation are not, for example because they include idiosyncratic path statements. I strongly recommend the authors to read through R guidelines such as the ones below, and cross-check that all the code adheres to these good practices:

<https://www.tidyverse.org/blog/2017/12/workflow-vs-script/>

<https://www.carlboettiger.info/2013/06/13/what-I-look-for-in-software-papers.html>

To conclude, I believe that this is certainly a useful piece of software, however for me the manuscript reads too much like a manual instead of a scientific paper, even of the type that GMD aims at. I think that there is scope for streamlining, and ideally going a bit beyond simply presenting a wrapper, towards exploring how even something as simple as a wrapper can incorporate state-of-the-art software design concepts. This does not mean that the software needs to be entirely implemented according to the recommendations above. But some attempt, or at least a discussion as to why this may be scientifically non-trivial, would lift the scientific value of the manuscript in my opinion.

