We thank Mr. Astagneau for their comments, and for taking the time to review the submitted manuscript. Responses to each submitted comment are provided below, and we plan to make changes to the manuscript in response, as indicated in the author comments.

Original review comments are provided in *italics*, and author responses to the comments are in **bold** font in the sections below.

0.1 Comments

This manuscript presents a new R package which aims at helping modellers in their use of the Raven hydrologic framework. Most of the package features consist in functions for data wrangling to feed Raven and functions for simulation analyses. Rationales behind the implementation of RavenR are presented. Examples of the RavenR functionalities are introduced using a formerly built perceptual model of the Liard river basin.

Several authors have advocated for the use of flexible structures for systematic testing of multiple working hypotheses in hydrological modelling. The use of such structures inherently results in higher complexity for modellers hence a challenge for reproducibility of methods and results. I think that any attempt at improving the use of these flexible structures is therefore relevant to the community of hydrological modellers. Furthermore, an extensive documentation is introduced to use the RavenR package, lots of interesting functionalities ranging from data preparation to simulation analysis are implemented and feedbacks between users and developers are encouraged to maintain and improve the package.

However, to be able to thoroughly evaluate the added value of using RavenR, I would have needed some experience with the Raven hydrologic framework. As it is not objectively possible in the time required to write a review, the following comments can only be seen as a way to improve the readability of the paper for non-Raven users and broaden the possible reach to the hydrological community.

0.2 General comments

1. *Two similar flexible hydrological frameworks need to be cited in this work (either in the introduction or in Sect. 2): DECIPHeR (Coxon et al., 2019) and SuperflexPy (Dal Molin et al., 2020). A short description of the main differences between Raven/RavenR and these frameworks might further demonstrate the added value of using RavenR.*

   Thank you for these additional citations to include. We agree that a short description of these differences would be well placed in the manuscript, and we plan to include this in the introduction or section 2 as suggested.

2. *To improve understanding by new users of Raven (or even new hydrological modellers), I suggest adding a short description of the main choices that were made in the Raven hydrological framework and RavenR in terms of programming languages. The Raven hydrologic framework is coded in a compiled programming language, probably for computational speed and flexibility purposes. To improve its usability, the RavenR package was created. However, some hydrological models are coded in a compiled programming language and interfaced by R using packages (e.g. hydromad; Andrews and Guillaume, 2018). Why is the Raven workflow (in terms of programming languages) more suited for flexible modelling?*

   This is worth emphasizing in the manuscript, and will help to partially address the discussion by RC2 as well. In short, the Raven framework is compiled in C++ for speed, and has many design features that put flexibility of the hydrologic model as the core consideration. The RavenR package is designed to improve the workflow with Raven, and uses the tools in R that are computationally slower to use but are perfectly suitable for analysis, and with the benefit of easier development. Due to the size, complexity, and rapid rate of development of the Raven source code, implementing and maintaining the Raven model within RavenR with the Rcpp library or a similar approach would be a massive undertaking with many technical challenges. Keeping the two separate allows for better code management and development of both software packages.

3. *Section 3 is probably the most important section of this paper if we want to use the RavenR package and the Raven hydrologic framework. The steps of the hydrological workflow are presented in Table 1 and the*

related R code and model files are provided to understand the functionalities of RavenR. However, I found some parts of this section a bit difficult to understand, especially since in the provided R script, the model run command line appears before input file processing.

The Rmarkdown file provided with the manuscript (RavenR_use_cases.Rmd) is intended to highlight and demonstrate certain functionalities of the RavenR package, rather than provide a complete sequential set of steps to develop all model input files and analyze all outputs. Thus, the model run call is made in the file prior to some of the sections to demonstrate input file processing and other tasks. This can be made more clear in the manuscript for section 3.

4. *The authors state line 195 that step 4 and 5 will not be presented but it is not clear why. They are important steps of the hydrological workflow especially when performing uncertainty analyses. An explanation of why this is not relevant given the objectives of the paper is needed.*

   The use cases focus on a subset of the available tools within RavenR, with an emphasis on how the package can be used to reduce the modeller's effort in working with Raven. A use case on running the model was not deemed to be required as it is a relatively straightforward command, and is shown in the Rmarkdown file provided in the repository - this can be mentioned in the manuscript.

5. *Although it is probably relevant to introduce the notion of locked or protected HRUs in Sect. 3.2.4, hydrological modellers with less experience with Raven might need a simpler use case of model discretization first. If the authors want to keep this section as it is, I suggest adding a simpler example in the future vignettes of the package.*

   This is a good comment to provide a simpler example for new users. We will consider updating the manuscript and use case file to show a first case where no locked or protected HRUs are provided, add this result to the plot, and update the discussion accordingly to highlight the difference. At a minimum, we will direct users to the function example in the package, which does not require the definition of locked or protected HRUs to run.

6. *Sect 3.3 may be too long and its purpose not very clear since the evaluation of what the authors call "model realism" does not lead to questioning the hypotheses behind the Liard basin model. I think this section should be limited to a presentation of the possible analyses of model simulation enabled by RavenR. Possible cuts: l 376 to l 381; l 383 to "Overall" l 386; from "A similar check" l 396 to l 402; from "The model" l 407 to "bias in estimation" l 408; from "The hydrograph" l 430 to "peak" l 433; from "The plot" l 446 to l 448; from "The results" l 452 to l 453; from "The plot shows" l 460 to "measurements" l 464.*

   We thank you for this recommendation, and agree that section 3.3 should be reduced to be more succinct, and perhaps reduce the number of figures and related explanations in the current version of the manuscript. We will take into account these recommendations in producing a revised version.

7. *Overall, I think that the R script provided to understand Sect. 3 could become a vignette but for a very simple use case that would include parameter estimation procedures and questioning of modelling hypotheses. Building a simple model from data preparation to output analysis using a catchment from the Camel dataset (Addor et al., 2017) would allow very different modellers to use the Raven hydrologic framework.*

   The RavenR software is not necessarily intended to build every required or desired input file from scratch in R, but to provide tools to make this process more efficient for users. The manuscript is to demonstrate some of these useful tools. We do not think this manuscript is the best place for a complete tutorial on building model files from scratch, however, we can point to the existing RavenR package vignette and the Raven User's Manual as useful resources in this respect. We also agree that an accompanying vignette for the package that walks a user through these steps would be a very useful addition, and we plan to include this in future versions of the software.

0.3 Minor comments

1. *I think that lines 60 to 70 could be moved just after line 44 for better links between the paragraphs of the introduction.*

Thank you for this suggestion, we agree and will update this in the revised manuscript.

2. *Please add the references of Python, R and C++.*

   This will be updated in the revised manuscript.

3. *Line 128, "3) running raven" should be moved before "2) reading output files".*

   Thank you, this will be updated in the revised manuscript.

4. *Line 349/350: please remove "providing. . . for the right reasons? (Kirchner; Euser et al., 2013)", as it is not the place to provide insights into a scientific question that was not presented in the introduction.*

   This will be updated in the revised manuscript, and done in tandem with providing better definition for model realism and reality checks.

5. *Please define "model realism" and "reality checks" in Sect. 3.3.1, as they are vague concepts, especially when no other data than streamflow are available for model validation.*

   Clarification on the use of these terms will be provided in the manuscript, along with a note about limiting these checks to the available data (i.e. streamflow observations).

6. *Line 365: I do not think that the term "observed baseflow" can be used to refer to the results of baseflow separation techniques that rely only on streamflow time series.*

   Thank you for this point, the term for "observed baseflow" will be updated in the revised manuscript to ensure this distinction is clear.

7. *Lines 414 to 418 should not appear in Sect 3.3.*

   These lines will be removed from section 3.3, and consider instead making a note in section 3.2.2 about how forcing data may be read in from Raven output as part of a workflow.

8. *Line 449: "Figure C" should be "Figure B".*

   Thank you, this will be reflected in the revised manuscript.

9. *Line 550: "Figure D" should be "Figure C".*

   Thank you, this will be reflected in the revised manuscript.

0.4 Technical comments

I noticed a few typos. As I am a non-native English speaker, the following comments might not be relevant.

1. *L1: "advances. . . have enhanced" instead of "has enhanced".*

   Thank you for this, the correction is valid and will be updated.

2. *References such as "(e.g. GR4J (Perrin et al., 2003))" should appear as "(e.g. GR4J; Perrin et al., 2003)". The latex command for this is:*
   citep[e.g. GR4J;][]citationkey.

   Thank you, this will be reflected in the revised manuscript.

3. *Line 312: "The development. . . requires" instead of "require".*

   Thank you for this, the correction is valid and will be updated.

0.5 Comments specific to the R package documentation

1. *From my understanding, the pipe operator is not mandatory to run the Raven package and is only used here for better readability. However, some R users are not familiar with the dplyr syntax. Although this is mentioned in the title of Figure 2 of the article, I would recommend adding this information in the package documentation (if not done already, I might have missed it).*

   Thank you, this is correct and will be noted in the package documentation (including the package vignette).

2. *For some functions (e.g. rvn_annual_peak), the units of the related arguments are mentioned in the detail section. It is always easier for users to find the required unit beside the related argument. I would suggest doing so in future versions of the package.*

   We agree, and will be sure to include units where possible in future updates to the package.

3. *I noticed that for some functions, time series must be provided at a daily time step. I thought that the Raven hydrologic framework could run at multiple time steps. Again, I might have missed the explanation at some point. If not, I would suggest adding a warning somewhere to use the time step required by RavenR/Raven.*

   Raven can indeed be run at any time step that is less than or equal to one day. The functions are generally setup to run for any time step but we will be sure to include warnings for those that require a daily time step in future versions of the package. Thank you for this note.