Mohammed et al. response for egusphere-2023-328

This paper presents novel open-source software packages and web-based environmental modeling applications for Earth observation data accessing, reformation, and presenting quantitative data products. This software is very useful for the scientists and stakeholders to further provide support for environmental modeling . This software can help to lower the technical barriers and leverage the distributed computing resources for environmental modeling. The user manuals are described very well and easily to follow. And the paper is well written.

Dear Referee,
We would like to extend our thanks to you for your valuable and constructive feedback on our paper. In the revised manuscript, we have addressed all the comments raised. Specifically, we have revised Table 1 by adding more information to help the readers in comparing NASAaccess with existing tools. We have also added some information related to NASAaccess framework retrieval time. We think that our revised manuscript has been adjusted to add strength and support to our tool. Finally, we have addressed all the minor edits as requested to enhance reading the paper easily. Please find below our detailed response to the comments listed.

However, I have some suggestions/questions:

1. **There are some similar functionality between the NASAaccess and other open-sources mentioned in Table 1. What is the gap between NASAaccess and other software in the Table? What benefit can we obtain by using NASAaccess compared to other software?**

The main benefits for NASAaccess framework can be summarized as: 1) an-open source tool, 2) modular - which means the framework could be replicated, customized, and implemented anywhere, 3) seamless earth-observation remote sensing and climate data ingestion into other modeling frameworks – NASAaccess gives ready formatted ascii data required to drive various hydrological models, and 4) lowering the technical barrier for leveraging and visualizing a wide array of satellite-based earth observations. The above-mentioned points have been discussed in section 4 of the manuscript. In the revised manuscript we modified Table 1 by adding three columns (Visualization Capability; Data Retrieval Format; Source Code Availability) to illustrate the differences between NASAaccess and some of the current NASA GES DISC tools and services for accessing and visualizing earth observation remote sensing data as requested.

2. **Line 130 - 135, once the data were generated and downloaded, what kind of data format it would be? And could you show some examples?This could be important information for the end-user to further use the processed data.**

Once the data was generated and downloaded, it would be in gridded ascii format suitable for ingestion by various hydrological models. We have presented some examples in section 3 showing the gridded ascii format layout. Yes, we do agree that this is important information for end-users since most existing tools do not give ascii output formatted to be ingested with other hydrological models.

3. **Line 175-176. The authors mentioned the .netrc file for storing the credentials information. What is the".urs_cookies" file for?**

This ".urs_cookies" file will be used to persist sessions across individual cURL calls, making it more efficient. The revised manuscript has been updated with this comment response.

4. **Line 175-177 is the same contents as 185-190**

Yes. The repeated content has been removed in the revised manuscript.

5. **I am not sure if I understand correctly. The raw data is saved in NASA' server with PostgreSQL database? The users can download the data which is not saved in the PostgresQL as mentioned in Line 114 . Then why the database is required to be downloaded by the database server as stated in Line 200 - 205. Indeed, it would be great if the authors can add a data flow and data information , i.e. where is the raw data, what are formats of the data, how the data is processed, where is the processed data stored,what is the processed data format etc.**

The Tethys framework comes with PostgresQL database to store data as a standard configuration. The NASAaccess web-based application does not create tables and store them in the PostgresQL database associated with the Tethys application standard practice. Rather, we have designed the NASAaccess web-based application to let the user download the data when it is ready immediately rather than writing the data in tables. This would save time in executing jobs using the NASAaccess web-based application. Line 113-114 in the revised manuscript says that the NASAaccess web-based application fetches and retrieves data without saving it in the PostgresQL associated with the Tethys framework. It is also important to point out that the Tethys Platform allows the user to choose if the app that one is going to develop has a PostgreSQL database integrated with it. In the case of the NASAaccess web application, no PostgreSQL database is integrated with it.

6. **The tool seems to be supported by R and Python. But the examples in the manuscript are only given to R. I would like to suggest authors to add examples that use Python. Or you can point to the document link where there is any example for Python**

Yes, NASAaccess conda package supports R and Python because of the nature of conda which is language-agnostic binary package manager. All the examples given in section 3 can be replicated using Python by calling the Rscript executable in any conda environment that has the conda package NASAaccess installed. Lines 488 – 492 in the revised manuscript show an example that uses Python with the NASAaccess tool.