

Code Review Checklist

Most important before paper

Important every session

Optional

<u>Usability</u>	<u>Comments (at least Yes / No)</u>
Is the code easy to install/run? Are there setup instructions and a list of requirements?	Yes there is a manifest
Is there an example script or a full pipeline that is easy to run and understand?	No, we don't see the entry point of the pipeline
<u>Data preparation</u>	
Are data loading and analysis implemented as separate steps? Ideal: have a data loader class	Yes there is a preprocessing.jl
Is ALL data used available in the cluster	No (at least the data in the repo is not compatible with the code)
<u>Analysis & Plotting</u>	
Are the different steps of the analysis clearly identified in the README?	No, the readme can give some indications on how the code is organized
Does the analysis code reflect what is described in the paper? If applicable	It should be better documented
Is it clear what code is used to create each of the figures or panels in the paper?	No. You could add the figures on the repo.
<u>Code quality</u>	
Project in periodically updated in github, gitignore, README	Lacks a .gitignore
Project structure: folders: data, notebooks, scripts, figures	Yes (lacks the figures folder)
Is the code well organized (functions, classes, modules, settings, ... as applicable)?	Yes
Are all functions and classes documented?	No
Are some values hardcoded?	Yes some paths are hardcoded and not consistent with the data structure provided in the repo
Can any of the code be replaced by existing packages/functions?	It seems so, but we are not good enough in julia to answer
Are there any obvious optimisations that will improve performance?	Same as above

Is there any redundant code that should be removed/refactored?	There are two BMS functions and they are unused.
Consistent, readable coding style (bonus points if. PEP8 for Python)	No, you could use a formatter to make the code more readable. (Shorter lines)
Variables names are self explanatory (eg no a, b, c etc)	No. A lot of variables are named tmp (71), g, x, y, c, ...
Are there any passwords in the repo or exposed in the code?	no
Is any identifying information unwillingly exposed?	Yes example: include("/Users/sami/PhD/Model_Tasks_Data/Data/WMM/pipeline/utils.jl")