

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?	More or less, I never manage to install (ssm)
Is there an example script or a full pipeline that is easy to run and understand easily explained in the README?	Yes, there are notebooks.
<u>Data preparation</u>	
Are data loading and analysis implemented as separate steps? Ideal: have a data loader class	Yes
Is ALL data used available also in the cluster	no
<u>Analysis & Plotting</u>	
Are the different steps of the analysis clearly identified in the README?	The readme can be more detailed and indicate a sequence of steps and the notebook corresponding to each step.
Does the analysis code reflect what is described in the paper? If applicable	
Is it clear what code is used to create each of the figures or panels in the paper?	Yes
<u>Code quality</u>	
Project in periodically updated in github, gitignore, README	Yes
Project structure: folders: data, notebooks, scripts, figures	Yes
Is the code well organized (functions, classes, modules, settings, ... as applicable)?	Yes
Are all functions and classes documented?	Yes: the functions in _scripts contain doc strings and the jupyter notebooks contain headers and comments
Are some values hardcoded?	No (constants are set at the beginning of notebooks, as they should be)
Can any of the code be replaced by existing packages/functions?	

Are there any obvious optimisations that will improve performance?	
Is there any redundant code that should be removed/refactored?	
Consistent, readable coding style (bonus points if. PEP8 for Python)	Yes
Variables names are self explanatory (eg no a, b, c etc)	
Are there any passwords in the repo or exposed in the code?	No
Is any identifying information unwillingly exposed?	

NOTES on code:

- imports with * are discouraged because they can re-import and overwrite variables and functions that have the same name in different modules.
 - Also we can't tell which function was imported from which package
 - Import with **import xxxx as xx**
- use pickles only for local caching or make sure that they only contain base python (and maybe numpy) objects
 - saving pandas and other objects leads to incompatibilities (can't load the pickle unless you revert to old pandas version)
 - note that it is inefficient for storing large amounts of data (takes more space than for e.g., .npy files)
-