

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?	Looks easy to install (tutorial in the last part). Miss the requirements.
Is there an example script or a full pipeline that is easy to run and understand easily explained in the README?	A full pipeline to execute and plots the figures
<u>Data preparation</u>	
Are data loading and analysis implemented as separate steps? Ideal: have a data loader class	Preproc, proc, analysis and second level analysis are made separately. Loader is not a class and data are loaded directly from the script.
Is ALL data used available also in the cluster	Data are available in clusters
<u>Analysis & Plotting</u>	
Are the different steps of the analysis clearly identified in the README?	Yes
Does the analysis code reflect what is described in the paper? If applicable	Yes, and the protocol is propose a notebook to walk through the analysis
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	One shot upload...
Project structure: folders: data, notebooks, scripts, figures	Some folders well describe. 13 script out of folders but with explicit name
Is the code well organized (functions, classes, modules, settings, ... as applicable)?	Settings and utilities are present, with uppercase for global variable (path, parameter). Some scripts with function inside.
Are all functions and classes documented?	Well documented for function and header of the script
Are some values hardcoded?	Don't have time to read all scripts...

	But it seems no
Can any of the code be replaced by existing packages/functions?	Can't have time to reach all functions, seems ok with right packages used for fmri, statistics and modelling.
Are there any obvious optimisations that will improve performance?	
Is there any redundant code that should be removed/refactored?	No
Consistent, readable coding style (bonus points if. PEP8 for Python)	Very readable, I like the inference part and the model part.
Variables names are self explanatory (eg no a, b, c etc)	Yes
Are there any passwords in the repo or exposed in the code?	No
Is any identifying information unwillingly exposed?	