Code Review Checklist

Most important before paper

Important every session

Optional

Usability	**Comments (at least Yes / No)**
Is the code easy to install/run? Are there setup instructions and a list of requirements?	Readme and python version listed , but no list of packages.
Is there an example script or a full pipeline that is easy to run and understand easily explained in the README?	Yes! Notebook for method (i.e. basis set) and instructions on how to run the code in the readme.
Data preparation	
Are data loading and analysis implemented as separate steps? Ideal: have a data loader class	For behavioural data, not exactly. Data is not loaded within an explicit loading function is not defined and data is loaded within the analysis loop (we think). For fMRI analysis, it is difficult to tell. Seems to be a defined function which loads data (e.g. load_cvs_scores, etc).
Is ALL data used available also in the cluster	Raw data yes, not results (we think)
Analysis & Plotting	
Are the different steps of the analysis clearly identified in the README?	Nothing about the steps. There is just a description of the repository structure and how to run the code.
Does the analysis code reflect what is described in the paper? If applicable	Sure…?
Is it clear what code is used to create each of the figures or panels in the paper?	Multiple scripts for some for of visualisation. In some files (em_plots), functions for plotting have descriptive names that can be matched to a figure. Other files that display plots are less clear what is being produced (for example, efun_plot_example)
Code quality	
Project in periodically updated in github, gitignore, README	Assume yes (one commit to the computational brain repo)

Project structure: folders: data, notebooks, scripts, figures	Yes.
Is the code well organized (functions, classes, modules, settings, ... as applicable)?	Utils folder contains the functions. Scripts are separate, and there are separate scripts for plotting.
Are all functions and classes documented?	In the utils, every function had a docstring, but limited comments within scripts.
Are some values hardcoded?	In some functions there is hard coding of parameter values, which could be separately defined for easier modification.
Can any of the code be replaced by existing packages/functions?	Not a clue…
Are there any obvious optimisations that will improve performance?	See above…
Is there any redundant code that should be removed/refactored?	Again… code base is too big to thoroughly check.
Consistent, readable coding style (bonus points if. PEP8 for Python)	Not completely clear, you can clearly define different authors' contributions and styles making some code more readable than others. No standard structure used.
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?	Only thing is authors names that are part of the project.