Code Review Checklist
Most important before paper
Important every session
Unclear

Usability	Comments (at least Yes / No)
Is the code easy to install? Are there setup instructions and a list of requirements?	No, the readme could describe how to install the code, what the different files are for, etc. The requirements are listed, it could be described more clearly how to use them/install them
Is there an example script or a full pipeline that is easy to run and understand?	No, there is no pipeline and it is not clear which script to run first, or what order to use
Data preparation	
Are data loading and analysis implemented as separate steps? Ideal: have a data loader class	Data loading is not clearly implemented as a separate step; it is also not clear how to prepare the data (e.g. use preprocessing, other scripts?, in what order?)
Is ALL data used available in the cluster	All data seems to be available on the github repository
Analysis & Plotting	
Are the different steps of the analysis clearly identified in the README?	The readme could include a lot more detail; how to install, what the order of the scripts is, what data is provided, what the output is
Does the analysis code reflect what is described in the paper? If applicable	It is not very easy to draw parallels between the methods section in the preprint, and the scripts available in the repo
Is it clear what code is used to create each of the figures or panels in the paper?	No, it is quite difficult to infer which figure can be created using which script - just adding 'Used for Figure A.b' to the script files would clarify a lot
Code quality	

Project in periodically updated in github, gitignore, README	Initial commit; so probably not regularly updated?
Project structure: folders: data, notebooks, scripts, figures	Yes, it is structured. Maybe add a functions folder?
Is the code well organized (functions, classes, modules, settings, ... as applicable)?	The functions are not explained, there are no comments in the scripts
Are all functions and classes documented?	Not really
Are some values hardcoded?	Some values are hard-coded, e.g. analysis_simu -> task = 1
Can any of the code be replaced by existing packages?	
Are there any obvious optimisations that will improve performance?	
Is there any redundant code that should be removed?	We don't see any obviously redundant code
Does the code agree with basic style guidelines (e.g. PEP8 for Python)	
Variables names are self explanatory (eg no a, b, c etc)	Variable names could be a little more explanatory
Are there any passwords in the repo or exposed in the code?	Not as far as we can see
Is any identifying information unwillingly exposed?	No