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?	Yes everything is well documented
Is there an example script or a full pipeline that is easy to run and understand?	No full pipeline script but the order of the scripts is clearly stated in the readme. The main scripts that need to be run by the user could be clearer categorized.
Data preparation	
Are data loading and analysis implemented as separate steps? Ideal: have a data loader class	Yes as far as we can tell
Is ALL data used available in the cluster	No it's on his server
Analysis & Plotting	
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	Unclear
Is it clear what code is used to create each of the figures or panels in the paper?	No
Code quality	
Project in periodically updated in github, gitignore, README	Only 4 commits + last one in February
Project structure: folders: data, notebooks, scripts, figures	There is a basic structure but it would be better if only the main scripts were outside of the folder structure. Without the readme it is difficult to know which scripts are the main scripts and which ones are helper scripts
Is the code well organized (functions, classes, modules, settings, … as applicable)?	There a clear functions, but sometimes they are in a separate helper scripts and sometimes they

	are part of a main or subscript.There is no settings file which could be useful.
Are all functions and classes documented?	Yes. Input and output could be clearer.
Are some values hardcoded?	Paths are sometimes hardcoded within the scripts
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)	
Variables names are self explanatory (eg no a, b, c etc)	yes
Are there any passwords in the repo or exposed in the code?	
Is any identifying information unwillingly exposed?	