

Code Curation Checklist

- Are source code files named appropriately, with an extension that matches the programming language?
- Are hardware and software dependencies adequately documented?
 - The hardware, operating system, and package versions used to run the code should be documented.
- Does the documentation describe how to use the code?
 - The README should give an example of how to use the code, including expected outputs or behaviors.
- Are important functions and components documented with comments in the source code?
 - Critical pathways should include inline comments describing what the code does (e.g., expected inputs and outputs).
- Are file paths in the source code portable? Would running the code on another computer or operating system cause the paths to “break”?
 - Relative file paths are preferred to absolute file paths. Ideally, paths are constructed using software tools that account for differences between operating system path delimiters.
- Are files referenced in the source code included in the submission?
 - If not, the documentation should include instructions on obtaining the missing files.
- Is there a README and is it appropriately descriptive. Contents may include:
 - Description: the code’s purpose and behavior, including any usage examples.
 - Contributors: name, institutional affiliations, and contact
 - Technical Requirements: notes on hardware and software dependencies
 - Known Limitation & Bugs
 - License
 - Preferred Citation
 - Additional Acknowledgements
- Does the code include a license and is it appropriate?
 - The code should be licensed with an open source software license, preferably a permissive one (Apache, BSD-3, MIT).
- Are files in the submission organized in a consistent and logical manner?
 - The directory structure should help to differentiate, for example, source code files and data files.