



The Open Source Ambassador

README Files for Code

A Guide for Writing README Files for Code

The ASAP Open Science Team

Dana Lewis, PhD

Devin Snyder, PhD

Robert Thibault, PhD

The ASAP CRN Single Cell Multi(omics) Working Group

Anthony Russo, MSc

Oliver Tam, PhD

Sonia Garcia Ruiz, PhD

Oke Avwenagha, PhD

Xianjun Dong, PhD

README Files for Code

The purpose of this document is to provide guidance on how to write README files for code.

The [ASAP Open Science Policy](#) requires that all code generated as part of an ASAP funded project be deposited in a publicly accessible repository, assigned a persistent identifier, and that the identifier be cited in all publications that use the code.

Throughout this document, we use the term *code* broadly to encompass all scripts, software, packages, libraries, macros, pipelines, algorithms, executables, batch files, and any other code that manipulates data in any way, including but not limited to cleaning data, preprocessing data, analyzing data, and producing figures, tables, and results. Code can be written specifically to clean and analyze the data for a particular study or with flexibility so that other researchers can easily use the code (e.g., software). This guide focuses on the former.

The ASAP Open Science Policy also requires that a README file be included in each repository where code is deposited. README files must include enough information that someone who was not part of the study team can access the code, identify which scripts execute which functions (e.g., clean data, produce figures), and understand the scripts well enough to reuse them.

Table of Contents

- [What is a README file?](#)
- [What should be included in a README file?](#)
- [How do I make a README file?](#)
- [Why does ASAP require a README file for code?](#)

What is a README file?

A README file is often the first thing someone will see when viewing code in a code repository such as GitHub. README files should be written to assume limited prior knowledge by the reader and should clearly describe the purpose of the code, including what data it processes, how it cleans and analyzes that data, any specific steps required to run the code successfully, and the outputs it produces.

The README file should be saved in the root directory of the repository where the project is deposited. This action can be easily completed by selecting one of the checkboxes available during the repository set-up or by uploading a README.md file into the repository at a later time.

What should be included in a README file?

There is not a single standard structure for README files for code. To view examples of README files, check out this [curated list](#). Regardless of the specific headers you use for your README file, it should include information on all of the items listed below.

1. Project Title

2. Project Overview

- Outline the purpose of the code and the project it relates to.
- Describe the type(s) of data used, supported file types, and output(s) of the code.

3. Documentation and Working Example

- Provide an example of how to run the code, including a text with a list of commands with relevant parameters (i.e., a shell script) and example outputs.

TIP: If it is not possible to include the data from your study within the GitHub repository where your code is deposited (e.g., it includes protected health information), we recommend providing dummy data (i.e., simulated/fake data) that can be used to ensure the code is running as expected.

- This section may also include links to a “Getting Started Guide” or tutorial and FAQs.

4. System Requirements & Dependencies

- List the system requirements to run the code (e.g., operating system, version). Including the CPU and memory needed can ensure users leverage large clusters when needed.
- List all software, libraries, and packages (including version numbers) which your code requires to run correctly. Where feasible, consider including code that checks for dependencies and an appropriate computational environment. If you have only tested the code in one version of the software, you should state “no other versions have been tested.”

TIP: We recommend sharing system requirements and dependencies in a systematic manner by running certain functions (e.g., [renv](#) or *sessionInfo* in R) and including the output in the relevant GitHub repository, or by creating and depositing a dockerfile, among other methods.

- If data are required to run your code but are not contained within the GitHub repository, explain where a user can access the data and where they need to save the data in order for the code to run properly.

5. Installation

- If dependencies are not automatically installed when running your code, present the code needed to install these dependencies.
- Provide the command to clone the repository to the user’s local machine. For example:
 - \$ git clone https://github.com/user_name/software_name.git
 - \$ cd software_name_folder
 - \$ make

TIP: Try to run the code on a different machine. This process can help detect all dependencies and system requirements.

6. Folder Structure

- Outline the content of the folders and files in your repository.

7. Limitations

- List any known limitations of your code. For example, *“This software has only been tested using input datasets containing a maximum of 3 million rows. Inputting longer datasets might produce unexpected results.”*

8. Citation

- Use the [Citation File Format generator](#) to create a citation for your code.
- If the code was written for a particular study, include a link to the relevant publication and/or preprint.

9. License

- While some people include a license in their README file, we recommend issuing a license for your GitHub repository by including a file entitled LICENSE.txt. This file will automatically be created when selecting a license within your GitHub repository.
- For more information about licensing code, see the [Licensing Guide](#).

10. Acknowledgements

- Acknowledge those who contributed to the GitHub repository and related outputs.
- Acknowledge ASAP with the following language, editing the text in square brackets as appropriate:
“This research was funded [in whole or in part] by Aligning Science Across Parkinson’s [Grant number(s)] through the Michael J. Fox Foundation for Parkinson’s Research (MJFF).”

11. Additional Items to Support Collaboration

- When preparing code you expect others to use for purposes beyond a single study you conducted, we recommend including additional items including:
 - How to contribute to the project / GitHub repository
 - A Code of Conduct
 - Templates for reporting *issues*, such as Bug Reports and Feature Requests.
 - A Change Log

How do I make a README file?

A README file can be written in any text file format using any text editor. Markdown (.md file extension) is the most common format for README files shared on GitHub. You can create a README from scratch or use a [template](#). You can also use a [web editor](#) to create a README file.

Why does ASAP require a README file for code?

README files have many benefits, including:

- Helping others understand your code and what it does
- Increasing collaboration and innovation in software development
- Fostering wider adoption of code and software
- Helping code and analysis pipelines be open and reusable
- Increasing reproducibility by providing an unambiguous record of the analysis pipeline
- Identifying and resolving programming errors