How to be a good member of a scientific software community [Article v0.1]

Alan Grossfield[1*]

[1]University of Rochester Medical Center, Department of Biochemistry and Biophysics

This LiveCoMS document is maintained online on GitHub at https://github.com/GrossfieldLab/article_templates; to provide feedback, suggestions, or help improve it, please visit the GitHub repository and participate via the issue tracker.

This version dated June 11, 2021

Abstract

Software is ubiquitous in modern science — almost any project, in almost any discipline, requires some code to work. However, many (or even most) scientists are not programmers, and must rely on programs written and maintained by others. As a result, a crucial but often neglected part of a scientist's training is learning how to use new tools, and how to exist as part of a community of users. This article will discuss key behaviors that can make the experience quicker, more efficient, and more pleasant for the user and developer alike.

***For correspondence:**
alan_grossfield@urmc.rochester.edu (AG)

1 Introduction

reveal the hidden curriculum – what's expected of a software consumer, how to ask for help, how to contribute

2 Prerequisites

Training articles should clearly state the target audience and knowledge prerequisites. Key prerequisites should be noted in the article abstract to permit readers to rapidly ascertain an articles suitability.

2.1 Background knowledge

Although the authors may imagine a particular career-level (e.g., undergraduate or graduate), given the diversity of disciplinary curricula, it is more important to specify precisely any knowledge prerequisites (e.g., vector calculus, basic thermodynamics).

2.2 Software/system requirements

If a particular software or programming environment plays a central role in the article, that should be specified.

3 Content and links

A training article may on additional files and materials; clearly indicate where and how these are available, with links, and how they are being archived for the long-term and maintained so they stay current. You will likely want to reference your GitHub repository as a central point to access all of this information, and then the GitHub repository may link out to other content as needed.

4 Checklists

5 Author Contributions

The initial version of this paper was written by Alan Grossfield.

For a more detailed description of author contributions, see the GitHub issue tracking and changelog at https://github.com/GrossfieldLab/article_templates.

GOOD COMMUNITY MEMBER

- ☐ Tries to solve problem themselves first
- ☐ Asks for help in the right place
- ☐ Writes informative bug reports
- ☐ Cites and acknowledges software appropriately
- ☐ Contributes to the community
- ☐ Treats fellow members and developers with courtesy and respect

POOR COMMUNITY MEMBER

- ☐ Doesn't read the manual or search the internet before asking for help
- ☐ Doesn't use the correct venue to ask for help
- ☐ Writes vague or unhelpful bug reports, or doesn't respond to questions
- ☐ Is rude or demanding when requesting support
- ☐ Treats fellow community members disrespectfully

A GOOD COMMUNITY

- ☐ Helps users solve their problems
- ☐ Is friendly and supportive when responding to questions
- ☐ Encourages respectful treatment of all community members, and calls out disrespectful behavior

6 Other Contributions

For a more detailed description of contributions from the community and others, see the GitHub issue tracking and changelog at https://github.com/GrossfieldLab/article_templates.

7 Potentially Conflicting Interests

Alan Grossfield serves as a consultant to two companies, Moderna Therapeutics and Atelerix Life Sciences.

8 Funding Information

This work supported in part by NIH R21GM138970 to AG.

Author Information

ORCID:

Alan Grossfield: 0000-0002-5877-2789