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 9, 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

Here you would explain what problem you are tackling and briefly motivate your work.

In this particular template, we have removed most of the usage examples which occur in `sample-document.tex` to provide a minimal template you can modify; however, we retain a couple of examples illustrating more unusual features of our templates/article class, such as the checklists, and information on algorithms and pseudocode.

Keep in mind, as you prepare your manuscript, that you should plan for a representative image which will be used to highlight your article on the journal website and publications. Usually, this would be one of your figures, but it must also be uploaded separately upon article submission. We give specific guidelines for this image on the journal website in the section on article submission (see https://livecomsjournal.github.io/authors/policies/index.html#article-submission).

Additionally, for well-formatted manuscripts, we recommend that you let LaTeX handle figure/table placement for you as much as possible, so please avoid specifying strenuous float instructions like '[h!]' and '[H]' as much as possible.

1.1 Scope

Training articles are largely self-contained and introduce readers to the theory and/or typical procedures for a particular aspect of molecular simulation.

Training articles need not refer to specific software commands and procedures but may do so to further the pedagogical goals of the paper.

As needed, training articles are likely to display more details of calculations than typically are shown in research articles, for example, noting any mathematical principles or "tricks" used to derive key results.

The scope of the training article should be clearly defined. This will often happen in a specific section or subsection in the article itself.

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

Training articles do not necessarily require the use of a checklist as in Best Practices documents; however, they can include these if desired. Several useful checklist formats are available, with examples presented in `sample-document.tex` in github.com/livecomsjournal/article_templates/templates. One example is shown here.

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

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.

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