

Reviewer #1

The authors put together an interesting 10 simple rules article sharing their experiences with code clubs.

I find the proposed code clubs an interesting idea and would be interested in trying this out in the lab. However, quite a bit more practical advice from this manuscript would be helpful to get up and running. Practical advice can be in the form of things that worked, but also explicitly mentioning what didn't work so that others can learn from the authors experience.

Thank you for the feedback on where the manuscript could provide more context. Throughout their review, the reviewer encourages us to make more comments about groups where there are more people with CS backgrounds. Admittedly, our lab is mostly made of researchers with non-CS backgrounds and so our implementation of Code Clubs is with that audience in mind. We have added text as appropriate to respond to the following suggestions. We have expanded our listing of example code clubs in Table 1 along with links to GitHub repositories where they were used. Examples of Code Clubs that did not work are highlighted throughout the manuscript (e.g. teaching Julia or not clearly stating goals).

Some suggestions: Put this into context of the literature: is there any formal description of code clubs? What other learning approaches are commonly used to introduce and improve coding skills in people with no formal programming training?

How does this connect to other activities such as hackathons, pair-programming (buddy programming)?

We have provided a broader context in the second paragraph of the Introduction and with an expanded discussion of pair-programming in Rule 8.

Highlight more the benefits of both junior and senior coders, for now the benefits appears to be solely for the junior coders. What motivates the senior folks to join? Or alternatively, in the context of a more diverse lab, how would this work with people with CS and non-CS backgrounds?

We have added text to Rule 3 describing the development of more challenging problems for more experienced group members. There is also now a longer section explicitly stating the

benefits to more experienced group members and ways to keep their attention by developing sessions around packages and concepts that are new and accessible to everyone.

Some suggestions on how to implement this would be nice, i.e. how to get started, how to get people motivated to join, etc.

It is challenging to comment on this point since we have only done this a part of our normal lab meetings where lab members are compelled to participate. We have added some text to the Conclusion paragraph describing why this model should be successful for more independent groups.

Are these code clubs unique to biologists? How would this set up work in a hybrid cs/bio environment or perhaps even in a non-bio environment?

There's no reason to think that Code Clubs would be unique to biologists. We have added text to the end of the Introduction indicating that as long as the Rules are followed it should be generalizable to groups that are more proficient at programming. Just as Journal Clubs are valuable to all fields of science and levels of experience, we think Code Clubs should also be valuable to the same diversity of participants.

I would be interested to read more about how much effort it takes in reality from both participants and presenters to prepare? How many hours each week need to be dedicated?

In the second to last paragraph of the Introduction we added a sentence indicating that we have found that it takes about as much effort to put together material for a Code Club as it does for a Journal Club session.

It would be interesting to see a much more extensive proposed program. The examples in Table 1 are illustrative, but far from complete. Can the list be expanded with repeatable activities (git was mentioned in the text) that could constitute a rotating 1 year program and a list of example side activities that are more one-off.

Our goal in this manuscript is not to be proscriptive about what the session should contain. We are presenting a framework that diverse groups can use to achieve their own goals with their

interests and level of background. We suspect one could certainly create a “schedule” of topics, but the challenge would be sustaining interest across years with the same cohort of participants.

It would be nice if the above suggestions can be embedded within the rules. To create the space, I think rule 7 and 8 can be merged. Rule 7 seems rather obvious. Rule 6 and 9 are connected and would make sense to have them subsequently, or alternatively as a single rule with two sides of the coin.

In our experience, Rule 7 was not obvious and led to the deeper discussion that caused us to re-form our Code Club format. Rather than moving Rules around, we’ve incorporated the reviewer’s comments throughout the manuscript.

Reviewer #2 The manuscript from Hagan, Schloss et al. presents a guide for developing computational skills in labs/groups/buildings/departments, and especially in those environments that include wet lab scientists. This is a timely piece given both the increased importance of computational analyses over the past 1-2 decades, as well as the more recent shuttering of many wet labs due to COVID-19. The piece describes steps to take to develop 1-hour “Code Club” sessions. At these sessions, presenters pose a problem and pair students to solve the problem. After the pair programming, there is a opportunity to reconvene and present solutions.

Two general scenarios are explored – one in which the presenters are helping to solve a problem for the presenter, and another in which the problem is didactic and the key beneficiaries are the other participants. Specific examples for sessions are provided, and additional guidance is offered throughout the rules. Overall, the rules are well organized and the direction is clear.

It would appear that the authors violated the first rule of “Code Club” in writing about it, and that is to the benefit of all of us. I have only minor comments:

We thank the reviewer for their comments on the manuscript and suggestions for where to add more detail and where we can clarify the text.

L81-91 (Rule 3): Can you provide more detail in this section? What are examples here of how to “experiment”, or what should be avoided? Additionally, returning to basics for new participants seems to conflict with the message to try something new.

We have added examples of how we've been experimenting with format during the COVID-19 pandemic to the Introduction as well as variation in difficulty of problems that are assigned in Rule 3.

L125 (Rule 6): Are the authors recommending a separate GitHub repository for each club meeting, or one general one for the club? Are any of these materials public that they could be shared here as an example? I reviewed the materials at “https://www.riffomonas.org/code_club” but I assume these are more elaborate than what is used in-house.

We have added text to Rule 6 detailing what we have found to be pros and cons of hosting material on GitHub. Although interacting with git and GitHub can seem trivial to more experienced people, it can be intimidating to beginners. In many cases, using a dedicated Slack channel or email has lower barriers and allows us to focus on the topic for the session rather than worrying about git configurations or GitHub permissions. Although not all of the examples in Table 1 have repositories on our group's GitHub account, we have included links to those that are available. Looking at these, one will see the variation in “polish” across the sessions, but will also get a sense of the overall approach we have taken to Code Club sessions.

L32: Suggest rewording “and resulted” to “, resulting”.

We have edited the text as suggested.

L60: Suggest rewording “last two for participants.” to “last two for non-presenters.”

We have edited the text as suggested.