Response to the reviewers

January 17, 2023

## Editor

As a long-term but sporadic GitHub user myself, I agree with most of the points and cases presented in the paper.

#### Streamline use cases (E1.1)

I agree with Reviewer 1 that these could be streamlined somewhat into a smaller number of more generalizable cases. They also are a bit heavily weighted towards more experienced users. “Simple” operations like push, pull, and commit are rarely understandable to new users, who routinely forget to commit their actions. The GitHub Desktop app makes many of the operations and use of GitHub somewhat easier, and is a good entree into GitHub and Git for novices. More could be made of the desktop app.

1. we reduced number of sections. See also our response to reviewer 1 ([Section 2.0.1](#sec-R1.1)

We have also reduced the amount of jargon repeated throughout the use cases by adding a paragraph in the introduction that describes a basic git/GitHub workflow and directs readers to other sources for learning the more technical aspects. This allows us to be more concise in the use case sections. See also our response to reviewer 1 ([Section 2.0.1](#sec-R1.1) and [Section 2.0.2](#sec-R1.2)). Use of the GitHub desktop app and RStudio as GUI interfaces for git is discussed in Box 1 of our manuscript and also in our introduction . We believe that these mentions, combined with reduction of git jargon and directing readers to additional sources to learn git and GitHub, is sufficient to address this comment.

#### Address private repos (E1.2)

In my own experience, the “public” nature of repos is a major barrier to the use of GitHub for exactly the reason you suggest - EEB practioners remain very uncomfortable with sharing data pre-publication (and often even post-publication). Some additional emphasis on the ability to make repos “private” should be included beyond the Boxes.

We now directly discuss the possibility of keeping repositories private in the section “Storing and sharing research compendia” and in the “Open science discussion” to emphasize that there is not necessarily a trade-off between privacy, collaboration, and open science practices.

#### Discuss documentation of single-purpose scripts (E1.3)

Another reason that many EEBers don’t use GitHub appears to be related to how we develop code for projects. Unless we are developing packages that others are likely to use, we tend to develop one-off scripts or functions that, once they work and are run, are never returned to. Version control is not used widely, or even needed in these cases (and my own work in software engineering provenance software has similarly had little uptake for the same reasons). I recognize that this makes it difficult to document the processes involved, but if the goal is simply to analyze the data at hand and push out another paper, why deal with the overhead of version control? At root, this is a sociological phenomenon, not a scientific one. Within your ms., consider saying something about how even one-off scripts may need to be documented, archived, and assigned a DOI to conform to evolving archiving requirements of journals.

This is a great point. We have added the following sentences to the end of the “Why aren’t more EEB researchers using GitHub” section :

Other scientists may simply lack the time or incentives to document and version control their code if the code is unlikely to be reused beyond their analysis. However we (and others, \_e.g.,\_ [@doi:10.1098/rspb.2022.1113]) argue that the endeavour of open science and collaboration requires code owners to document and version control code despite uncertainty around future use.

#### Discuss Overleaf integration (E1.4)

I am surprised you didn’t include integration between GitHub and Overleaf. Overleaf is also a good collaborative latex-based text processor, and provides syncing with GitHub (and Dropbox) in a way that will be more familiar to beginning users. (Actually, I eventually found it buried in Table 1)

We now discuss the possibility of integrating GitHub into existing workflows on Overleaf, Google Drive, and DropBox among others to collaboratively write scientific manuscripts, in the section “Writing a manuscript” . We refer to Table 1 for further details on these integrations.

#### Improve discussion of GitHub advantages over other existing file-sharing tools (E1.5.)

Finally, DropBox, SharePoint, and OneDrive are file-sharing systems that include version control, are widely supported, especially in colleges and universities, and are much easier to use for those who might be uncomfortable learning a system such as GitHub. Other than the ability to clone and fork, why should one choose GitHub over, for example, DropBox or SharePoint? More should be made of this in text, as opposed to burying it in Table 1, which is referenced only once in text (line 232) and only in the context of other longterm archives (line 231). Indeed, I suggest a somewhat expanded section of the ms. dedicated to a detailed exploration of the comparisons made in this Table. Otherwise, the ms. comes across as more proselytizing and less analytical/convincing.

We think this is a good point since many readers may not immediately see the value in a time investment to learn a new tool when they are comfortable with tools that meet many or most of their needs. Rather than create a new section, we opted to identnify specific points in the manuscript where we could directly call out where GitHub as a platform enables greater or different types of collaboration and open science than platforms like dropbox or google. Due to these edits (below) we now more often refer to table 1, and earlier in the manuscript as well.

In the first paragraph of the introduction we made the text more concise throughout and also added new sentence :

Of all these tools, GitHub is uniquely positioned (Table 1) to benefit scientists because the platform is designed to enable collaboration on computer code--a fundamental component of modern research.

In the second paragraph we added :

Additionally, unlike sharing documents and code via DropBox or Google Drive links, GitHub defaults to public sharing of code and documents and has features like Issues and Discussions (see Box 1) that can immediately and directly contribute to more open science and knowledge sharing.

Under the “storing a research compendium section” we added :

While some EEB researchers may prefer to store their code in Google Drive or DropBox, the process of sharing code through these services becomes cumbersome when collaborators have to download and re-upload edited code.

On GitHub, code is by default public, and the process of forking and submitting pull requests (see Box 1) ensures a code owner has full control over code and documents in a repository.

In the first paragraph of discussion we added :

While tools like the Google Suite and DropBox enable rapid sharing and collaboration of some research documents, GitHub brings together features that directly address open science such as tools for soliciting public discussions on code and findings, tracking multiple rounds of edits to files in a sensible way for easy review, and managing complex research projects with many collaborators and goals.

And finally, in the discussion, under “why aren’t more EEB researchers using github”, we added :

And with institutional availability of software licenses for tools like the Google Suite or Microsoft Office, researchers may be hesitant to spend valuable time learning another tool.

## Reviewer 1

This paper clearly communicates the enthusiasm of the authors for integrating GitHub in the EEB research process. Advantages in areas of collaboration, transparent and reproducible science are clear and nicely discussed. This is a well written paper.

#### Reduce repetition and number of use cases (R1.1)

I would, however, recommend some restructuring and shortening. As it stands, this paper is too long and some of the use cases are too nerdy to spark enthusiasm in somebody who is not already using GitHub. Breaking it into 13 use cases leads to some repetition and those use cases may be combined more effectively for convincing a new user.

In order to reduce repetition and jargon in the use cases sections, we’ve added a paragraph to the introduction explaining a basic git & GitHub workflow and pointing the reader to other resources for learning these tools. This allows us to focus less on explaining the specific mechanisms of git/Github in the use cases sections and more on the usefulness of the tools. In regards to the specific suggestion about deprioritizing GitHub communication tools, we believe these are extremely useful alternatives to reply-all email chains for a variety of reasons and have attempted to improve the clarity of this section of our manuscript [end of project management section].

#### Reduce jargon (R1.2)

The title and abstract gave me the impression that the goal is to convince EEB researcher to start using GitHub. If that’s the case, it might be better to tailor the use cases to that entry level and use less GitHub specific lingo. Advanced usage may be mentioned but not detailed as much as it is currently done. E.g. collaboratively writing a paper in GitHub is probably out of the question for most. Most GitHub options for communication, discussion, issue tracking are still somewhat esoteric for most non-programmers.

As mentioned above ([Section 2.0.1](#sec-R1.1), [Section 1.0.1](#sec-E1.1)), we were able to reduce jargon by making it clear in the introduction that our aim was not to *teach* the use of git and GitHub, and by discussing a basic git workflow early on.

#### Define use-cases in terms of return on investment (R1.3)

Although the abstract states ‘We outline features ranging from low to high technical difficulty’ the paper reads a bit like a laundry list of what GitHub can do (in fact, the word ‘can’ is used about 140 times, which makes for tedious read). Figure 2 helps sort through this laundry list and defines the technical difficulty. It might be better to clearly lay out where anybody can start using GitHub effectively in the text. And the emphasis is on ‘effectively’. Most people are not likely to learn a new piece of software if it does not promise to reduce effort and time. So, defining tasks where GitHub shines in terms of return on investment maybe the better approach to convincing new users and then only mention the advance use cases with some pointers to further reading, but not going into too much detail.

We appreciate this comment and while we do agree that ordering the use cases by value to the reader would be ideal, we believe that value or return on investment is subjective and varies among fields, labs, and individuals. For example, the reviewer sees collaborative writing of a manuscript using GitHub as an “out of the question” advanced use. However, all of the co-authors on this manuscript have done exactly this and found it an excellent return on investment. At the same time, not all co-authors have experience in many of the other use cases we present and may find them less useful depending on their needs. We do make an attempt to quantify the investment effort in Figure 2 already. We’ve added a supplemental document better explaining *how* we made these quantifications (see also our response to reviewer 2 [Section 3.0.2](#sec-R2.2)). However, we have taken the advice of reviewer 1 in other comments ([Section 2.0.4](#sec-R1.4), [Section 2.0.1](#sec-R1.1)) to reduce the number of use cases and re-order them.

#### Reorder sections based on importance (R1.4)

In my experience, the project continuity is actually very high on the importance list for researchers, i.e., knowing that the code and data will be findable by the next student. This includes the discussion of organizing and managing teams, keeping lab information in one place etc. Followed by code versioning and the ability to go back to older versions. Interest in website development is picking up because it really is simple to do in GitHub, and the information can be maintained by several people (i.e., a lab group).

#### Detailed comments:

**R1.5**

Line 346 delete second ‘can’: who can also can change through time

Corrected! Thank you for catching this!

**R1.6**

Line 389 it should be ‘each other’s work’: contribute to each other work without necessarily

Corrected! Thank you once again!

**R1.7**

Line 457 missing ‘collaborator’? especially when many may be

We have added “prospective users” to this sentence. It now is “*It can be challenging to learn Git alongside scripting languages, statistical theory, and file system navigation, especially when many prospective users may be inexperienced with programming.”*

**R1.8**

Line 477: not sure what this sentence is saying: ‘requiring the complementation of other tools to fully integration project files and GitHub repositories’

We agree that this sentence needed clarification. We have simplified it and it reads now as: “*Moreover, additional tools may be required to fully integrate project files and GitHub repositories […]*”. Thank you once again!

## Reviewer: 2

The manuscript presents how ecologists and evolutionary biologists (EEB) can use collaborative software development project management tools. The authors present a high-level view of Github and the Git version control system that is at its core. As the authors detail, the tools that Github provides has broad potential to be leveraged within the EEB community. The paper is well written and helps to make Github tools and related concepts legible to EEB audiences. On the whole, I see this as potentially being an impactful paper for improving EEB research. I have the following Major and Minor comments for the authors:

#### Figure 1 was more confusing to me than helpful (R2.1)

Figure 1 was more confusing to me than helpful. It presents a view of a Github interface in order to detail generalized features; however, it is edited/abstracted so much that it doesn’t map easily to the interface as it would be viewed by a reader of the paper. To improve this I would reduce the level of abstraction of the web interface.

#### Improve Figure 2 (R2.2)

Figure 2 was useful, especially as it summarized across multiple areas that a reader make in-roads into using Github. Please clarify how the “Technical Difficulty” was assessed. Was this based on the impression of the working group? Was it quantified using different types of required knowledge (e.g., programming, software design, working in the Terminal, etc.)?

#### Add brief history of git (R2.3)

I would find it helpful to give a brief (1-2 sentence) history of Git (L85-89). Namely, that it was developed as an aid for software development within distributed groups of software engineers and was developed within an open-source framework so that it could be improved by the community. This provides more context as to why Github (as it extends Git) is a useful tool for collaboration “by design”.

We now briefly discuss the origin of Git and its expansion to applications beyond software development to help manage collaborations and community contributions, in the introduction .

#### Additions to Automation use case (R2.4)

I suggest adding a discussion on dependency testing within or following the paragraph on Automation (Lines 373-382). This is a project “ecosystem” phenomenon that comes from collaboration, where you build your project on the work of someone else. As projects change over time, they can alter other projects. Software engineers have been working on this challenge for a long time in distributed teams where different parts of software are being built by different programmers. Checks can be done automatically within the software engineering framework (see Pasquier et al. 2017 <https://www.nature.com/articles/sdata2017114>). Beyond detection, any major changes can be detected and presented without additional work from the user via features like badges (<https://shields.io/>) within the project page (e.g., README).

#### Figure detailing general scientific workflow (R2.5)

Related to Major Comments 1 and 4, given that the focus of the paper is toward integration into EEB scientific research workflows, I suggest adding another figure (or replacing Figure 1) that details a generalized scientific research workflow (e.g., Munafò et al. 2017 <https://www.nature.com/articles/s41562-016-0021>) and how Github supports/augments that workflow. This could relate to or build on Table 2, which a useful summarization of important features as they relate to different research participants.

#### Minor Comments

**R2.6**

The sentence from Line 66-67 is missing a close parenthesis.

Thank you for catching this. We closed the brackets.

**R2.7**

There appears to be a “fourth” barrier at Line 478. I suggest splitting the paragraph from Line 472 to 483 into two paragraphs and expanding on both “reluctance to share data” and “language-specific resources” as barriers to adoption.

Indeed! We have numbered all five barriers to using GitHub to clarify this section.

**R2.8**

Line 570 “could also be e” should be “could also be a”.

You are right! We have corrected this issue. Thank you once again!