

ENCORE: A practical implementation to improve reproducibility and transparency of computational research



Open Access This file is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. In the cases where the authors are anonymous, such as is the case for the reports of anonymous peer reviewers, author attribution should be to 'Anonymous Referee' followed by a clear attribution to the source work. The images or other third party material in this file are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

Reviewer #1 (Remarks to the Author):

This manuscript introduces the Enhancing Computational Reproducibility (ENCORE) standard for sharing computational project results in a way that is both highly transparent and supports full scientific reproducibility. The manuscript clearly represents years of effort and dedication on this topic. This reviewer commends the good work! However, there are the following outstanding issues with the manuscript.

Major:

- 1) Line 63: The description of the research life cycle in the Introduction leaves out critical steps, especially QC/QA of input data and evaluation/validation of data analysis results or software, depending on the research product produced.
- 2) Line 85: Some tools that pull directly from knowledgebases and repositories also hinder reproducibility. Thus, use of certain tools in data analysis pipelines will prevent full reproducibility.
- 3) Line 103: Must be very careful mentioning GitHub as supporting “scientific reproducibility”, which requires persistence of research results. The owner of the GitHub repository can delete the repository at any time. Therefore, technically GitHub and other code hosting websites that lack persistence do not adequately support “scientific reproducibility”.
- 4) There is no LICENSE file included in the standardize File System Structure in Figure 4. While this file would be optional, having it in sFSS examples is an important reminder. Also, the Navigation.conf file is not described anywhere. Is this suppose to be Navigate.html as described on line 329?
- 5) While the focus is on human readability, there should be some effort to have ENCORE support machine readability. A separate optional media_types.json file could be used to identify the media type (formerly MIME type) of each file in the sFSS. Such a file could be very useful for automated validation and reuse.
- 6) There does not appear to be any ENCORE-associated software to evaluate compliance with this standard. Such a tool would be immensely useful for supporting ENCORE use and compliance. The sFSS Navigator appears to be the logical package to extend for these purposes. Also, at line 308, the instructions and template do NOT “enforce internal consistency”, they only “support internal consistency”.
- 7) Be careful with guidelines on when to begin using ENCORE in an active project, which will likely depend on the type of project. Likely, ENCORE should not be used during active prototyping/testing/redesign cycles of software or pipeline development, since it would slow down the development. It seems natural to use ENCORE when software is being applied, maybe tested, but not actively being developed. Strongly encourage qualifying the recommendation of using ENCORE at the start of a project at Line 498. It really depends on the project. Also, ENCORE should not be thought of as an alternative to electronic lab notebooks, even though electronic lab notebooks could be used to populate ENCORE.

8) Line 484: why does ENCORE not consider replicability? Maybe you mean it does not “directly” consider replicability. Reproducibility can be thought of as a generalization of replicability that is not as strict. Replication implies exactness, while reproducibility implies equivalence. This is something that should be discussed, since precision may need to be mentioned in terms of reproducibility. For example, what level of precision is expected for “reproducibility”? Thus, metrics of variability of results should be encouraged in order to judge reproducibility.

9) The following article suggests that adequately supporting “scientific reproducibility” should be used as a litmus test for deciding correction versus retraction. Adoption of this guideline by journals would strongly support scientific reproducibility. The authors may want to embellish their discussion with this idea, probably around Line 554.

Hunter N.B. Moseley. "In the AI science boom, beware: your results are only as good as your data — Machine-learning systems are voracious data consumers — but trustworthy results require more vetting both before and after publication" Nature Feb 1 (2024).

<https://doi.org/10.1038/d41586-024-00306-2>

10) Line 576: Given the ubiquitous use of JSON with parsers in every significant programming language, JSON is probably a better ENCORE metadata format than YAML. Would suggest at least mentioning JSON, when mention YAML.

11) Line 583: Would suggest mentioning more container platforms than just Docker, given the Docker is being replaced by alternatives like Singularity.

12) Line 588: Workflow managers have their own versioning issues, which should be mentioned.

13) Explicit capture of software and package versions is likely something that should be encouraged in ENCORE and likely should be its own required metadata file within the sFSS. For knowledgebases accessed, the access dates can be used to represent version.

14) Line 598: Given that a list of entry IDs can be used to pull data from a knowledgebase or repository in an automated manner, it feels like ENCORE could handle remote datasets if an entry IDs file is considered part of the input data. This may not be what the authors intended, but it is a logical approach or “hack” to use, depending on your perspective. If analyses are sequential, then an entry IDs approach would also minimize the local storage needed.

15) Line 676: It may be better to create separate augmented ENCORE specifications for specific types of computational projects, instead of creating one monolithic ENCORE specification. Would allow separate working groups to develop these for specific purposes, without the limiting each other.

Minor:

Line 101: recommend mentioning other code revision tracking systems like Bitbucket from Atlassian.

Line 117: add comma after “In this paper”.

Line 134: change “FAIR requirements” to “FAIR guiding principles”.

Line 441: add comma after “In our view”.

Line 475: add comma after “remote computer system”.

Line 483: suggest adding “scientific” before “reproducibility”.

Line 483: change “allow harmonization within a research group.” to “improve harmonization within and across research groups.”

Line 495: use a colon after “computational project” instead of a comma.

Line 538: something feels grammatically incorrect in the sentence starting with “There are clear advantages for working reproducibility...”.

Reviewer #2 (Remarks to the Author):

The paper describes a very interesting approach to improving the reproducibility of computational workflows (called "ENCORE"), with (implicitly) a focus on computational pipelines that involve large amounts of data such as those occurring in bioinformatic analyses in biology and medicine. As such, the paper does not contain any new research or research results - its novelty lies in the infrastructure it provides to (potentially) the research community. The paper is very well-written and easy to follow. A comprehensive guide to the use of ENCORE is included as supplementary Information.

Whilst it is undoubtedly the case that were such a system to be adopted universally by the academic publishing industry then some of the issues around reproducibility would be greatly reduced. However, as I explain further below, I am not convinced that the approach advocated in the paper is generally applicable, or that it would be widely adopted. I also think that the issues addressed are not the primary cause of the lack of reproducibility of computational research. However, the work described does make a strong contribution in terms of the analysis of the problem, and in terms of the solution for certain types of research. Reframing the paper in terms of this wider contribution would, for me, make it relevant to a much wider audience.

Generalising the approach

The approach described has been 6 years in the making. It appears to have been used on 20 projects within the host organisation and supports 50 projects by “other groups”. The paper is (laudably) very candid about the complexities and difficulties of developing and using such a system, and the ENCORE guide in the supplementary information is equally open about the technical difficulties and limitations of the current version. The paper is written in an application-agnostic style, but the system appears to be designed to support bioinformatics “pipeline-style” projects. Reading both the paper and the Guide highlighted very well for me the issues in developing such a system, but the complexities and caveats of implementation and use suggested to me that to extend this out to other areas would require substantial work. I did really like the principles on page 5 of the guide and felt that a paper that highlighted the

principles and used the ENCORE system to illustrate in a particular application domain how these principles could be met would be a much stronger contribution than trying to suggest that other researchers might adopt ENCORE as is. This could then contribute to a wider debate with the academic publishing industry about how the industry itself might develop appropriate information platforms to support reproducible research. I have written about this previously and so won't repeat the arguments here about why I think this should be the responsibility of publishers not researchers.

Wider problems with reproducibility of computational research

Whilst the creation and support of information sharing platforms such as ENCORE is very helpful in moving us towards reproducible computational research, and may be the whole solution if all of the computational tools being used have been shown to be fully tested and reliable, in my view in most of computational science the biggest problem is that we have no way of knowing whether the code being used to generate the computational results is doing what the researchers think it is doing. This is because most computational software is not refereed, most software is not developed using sound software engineering principles (since researchers developing that software are rarely trained in software engineering), and computational software is often not made openly available (or where it is, it is not accompanied by appropriate documentation and installation guidance). These issues are discussed briefly in the penultimate section of the discussion, but I think should be brought forward into the introduction of the much more general paper that I suggest above discussing the principles underpinning the development of ENCORE. I would add a principle on software engineering practices, and discuss how an information sharing platform might implement simple checks on software quality (and in particular on software testing).

Conclusion

Whilst the manuscript highlights very well some key problems with reproducibility, and describes a really interesting approach within this lab to address this problem, the suggested approach would, in my opinion, have limited take-up even in the bioinformatics domain. A better approach for a journal such as Nature Communications would be to use the development of ENCORE to highlight lessons learnt that might be applied in a wider domain, and that might be incorporated into the data and information platforms provided by academic publishers. This would, I think, appeal to a much wider readership.

David Gavaghan, Oxford.

Reviewer #3 (Remarks to the Author):

See attached comments.

Reviewer #3 Attachment on the following page

Summary

The authors propose ENCORE as template for enhancing scientific project reproducibility. While I applaud these efforts towards reproducible science, I think this work could use significant improvement in several key areas. First, as ENCORE is GitHub-based it would make sense to leverage the breadth of tools for reproducible research already integrated into the GitHub environment. I feel that some opportunities have been missed to build upon features such as:

- Wikis: <https://docs.github.com/en/communities/documenting-your-project-with-wikis/about-wikis>
- Discussions: <https://docs.github.com/en/discussions>
- Projects: <https://docs.github.com/en/issues/planning-and-tracking-with-projects/managing-your-project/managing-project-templates-in-your-organization>
- Codespaces: <https://docs.github.com/en/codespaces/reference/using-github-copilot-in-github-codespaces>
- Issues templates: <https://github.com/devspace/awesome-github-templates>
- GitHub Actions: <https://github.com/features/actions>
- GitHub Container Registry: <https://github.blog/2020-09-01-introducing-github-container-registry/>

Similarly, I understand the intent of providing templates in multiple formats is to let users choose what format they're most comfortable with. But as it's currently implemented, I think these duplicate files may unintentionally add to confusion of which version is the one being used (unless the users are diligent about deleting unused versions of files, which I tend not to assume). Instead, it would seem a more natural solution to this is something like Quarto or Jupyter (which can accommodate many languages) or Rmarkdown (which is primarily for R but has extended to python and other languages in recent years), where the same script can generate multiple output formats (md, HTML, latex) by simply changing one line in the yaml header. This notebook approach also enables documentation, figures, and tables to be embedding directly next to the code in the same document which goes a long way towards reproducibility. As an example, tools such as BiocBook demonstrate this very nicely, all bundled within a fully containerised environment with CI/CD via GitHub Actions: <https://github.com/js2264/BiocBook>

To their credit, the authors do mention the desire to implement containerisation in the future. But in a world where dependency conflicts and installation issues is an almost universal barrier to reproducibility, I feel some degree of containerisation (or at very least a conda environment) is a minimum prerequisite to make ENCORE substantially impactful.

To some extent, many of the features proposed in ENCORE feel like a more cumbersome reinvention of the existing tools. I say this not to discourage the authors (I think their efforts have value!), but rather to point to resources that may make ENCORE more appealing and valuable to the broader community. Pending significant improvements, I think ENCORE could evolve into a valuable resource for the community that makes reproducible practices easier and more appealing. I would be happy to review a second round if these issues can be addressed.

Major Comments

1. "Another option is to generate a small program (script) that retrieves the sFSS template from the ENCORE GitHub repository and automatically fills in certain portions of the required documentation, which is particularly useful if one conducts projects with a certain degree of similarity."

- a. Great! I think this is a key selling point for ENCORE. To me, automation is one of the most effective ways of convincing people to use reproducible practices, ideally with as little effort as possible as the automation gets better and is more extensive.
 - b. Another way to scale things up may be with LLMs. I've been using GitHub Copilot a lot to help document some of my code that I hadn't bothered to go back and properly document. And it does a remarkable job in most instances. Of course, still requires inspection by a human to make sure it's correct, but even so it greatly accelerates the whole process and gets rid of a lot of the tedium.
 - c. GitHub Copilot Chat attacks this from a slightly different angle by providing a chat interface for coding questions: <https://docs.github.com/en/copilot/github-copilot-chat/about-github-copilot-chat>
 - d. You can even use GH Copilot on GitHub itself via Codespaces, and GH plans on making lots more integrations directly into the GH browser interface: <https://docs.github.com/en/codespaces/reference/using-github-copilot-in-github-codespaces>
 - e. Adding some CI that runs unit tests on the code through GitHub Actions could also be a very helpful avenue towards ensuring the work is indeed reproducible. Ideally, this would include programmatic data procurement from external sources like Zenodo, or at least toy versions of those datasets.
2. "The LabJournal template (tex, docx, md, txt) in ProjectDocumentation should contain general project documentation including but not limited to an explanation of the project's background and concepts, computational approaches, summaries of project discussions, new research ideas, and to-do lists. Preferably, the lab journal should contain pointers to relevant sub-directories and files whenever needed. Alternatively, one may maintain multiple lab journals in different directories containing documentation for specific parts of the project. The lab journal is important for the scientific legacy of the research group by ensuring that others can replicate what the original researcher(s) has done. We decided to deviate from standard practice and to also use the lab journal to record new ideas, provide summaries of (email) discussions, and to-do lists, since it is important to have a record of these for the supervision of the projects and for follow-up projects. Consequently, not all information in the lab journal can be shared with others (see Discussion). Each directory also."
 - a. While I completely agree with the idea of documenting these things, I think this would be much better done using the existing GitHub Discussions, GitHub Projects, and GitHub Issues. Since the ENCORE framework is already GitHub based, I think it makes sense to take advantage of all the awesome features they continue to develop wherever possible. That said, ENCORE could build upon this and provide ways to make these existing features even more useful and approachable.
 - i. <https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects>
 - ii. <https://docs.github.com/en/discussions>
 - b. For example, you can provide templates for GitHub Projects layouts, or templates for Issues.
 - i. <https://docs.github.com/en/issues/planning-and-tracking-with-projects/managing-your-project/managing-project-templates-in-your-organization>
 - ii. <https://docs.github.com/en/communities/using-templates-to-encourage-useful-issues-and-pull-requests/configuring-issue-templates-for-your-repository>
 - iii. See here for some examples of Issues templates that you may find helpful: <https://github.com/devspace/awesome-github-templates>
3. "One decision to be made by the project team is how to distribute the documentation over the various README files and the lab journal(s). However, as a rule one should document any

project file (code, data, results) in the directories in which these are located. The lab journal can then be used for more general documentation.”

- a. GitHub Wikis seems like it would be a useful solution for exactly this, as it’s all based on a series of markdown files: <https://docs.github.com/en/communities/documenting-your-project-with-wikis/about-wikis>
4. Notes on the ENCODE GitHub repo: <https://github.com/EDS-Bioinformatics-Laboratory/ENCORE>
 - a. I think it would be helpful to have the getting started instructions in the landing page README. I see there’s a file named 00_README_FIRST.md, but it’s easy to gloss over this at first. Having a README that appears on the landing page in the template makes this more obvious.
 - b. The file system feels a bit cluttered with duplicate versions of the same files in different formats. For example, 0_GETTINGSTARTED has four different versions! (0_GETTINGSTARTED.docx, 0_GETTINGSTARTED.html, 0_GETTINGSTARTED.tex, 0_GETTINGSTARTED.txt). Same goes for many of the other files.
 - c. If the goal is to provide users different choices of formats, would it not make more sense to use a single file type that can be rendered into whatever other file type they want? This is what Quarto and Rmarkdown do, and the outputs get automatically organised into a standard folder structure. This would reduce clutter and potential confusion over the duplicate files.
5. “- Researchers that aim to reproduce the project: start by opening ****Navigate.html**** in your browser.”
 - a. Why not launch this as a GitHub Pages website, so users can start exploring this before even cloning the repo locally?
6. “- The FSS can be tested by executing the Navigate.py script (if Python is installed on your computer), or by downloading and executing one of the executables.”
 - a. I think this approach could be problematic as it means everyone will have their own copy of the python script. If any updates/fix need to be made to the script, every user must independently update their copy of the script. Instead, having a centralised distribution of the software, e.g. as a python module distributed via pip and/or conda, would be a more sustainable approach.
7. I would like to see a lot more work done on automating many of the processes involved in setup and maintenance of the ENCORE framework. Automation takes away a lot of tedium that drives people away from reproducible practices and makes it more likely they’ll bother to follow the guidelines.
8. “For example, recently detailed guidelines for a standardized file system structure for scientific data were published by Spreckelsen and co-workers (Spreckelsen et al., 2020), which inspired the sFSS used in ENCORE. However, they use a different organization, i.e., the top-level of their file system layout is not an individual project like in ENCORE but an experiment, simulation, data analysis, or publication.”
 - a. Given the similarity, I think this warrants a much more detailed discussion of the strengths of ENCORE over the prior work. This should at least be mentioned in the introduction as well.
 - b. Have there been other proposals like this? If so, would be good to talk about them, what they contributed. It’s also worth studying them to examine why they may not have caught on as intended, and what ENCORE can learn from those mistakes.
 - c. One that comes to mind is more R-based but nevertheless may be informative: <https://ro.uow.edu.au/smhpapers/5389>
9. For the code scripts, I think there’s a good argument to use Quarto files which can accommodate a number of different programming languages and provide lots of ways to document code in-line. Check out BiocBook for a good example of how this can be taken even further and create a

- Docker container with all dependencies. BiocBook is more geared towards R users but could definitely be extended to any programming language. <https://github.com/js2264/BiocBook>
10. GitHub Codespaces is another resource to potentially leverage. No downloads necessary, and you can even bypass installation with preinstalled Docker containers that users can spin up and run the code right away.
 11. “This stack comprises (i) (Bio)Conda (Anaconda Software Distribution, 2020; Gruning, Dale, et al., 2018) to provide virtual execution environments addressing software versions and dependencies, (ii) container platforms such as Docker (Nust et al., 2020) to preserve other aspects of the runtime environment, and (iii) virtual machines using cloud systems or dedicated applications such as VMware, to overcome the dependencies on the operating system and hardware. We are currently investigating how to best approach this within the context of ENCORE.”
 - a. I think this will be crucial moving forward. Containerisation can make a massive difference in reproducibility and installability. And there’s an increasing number of ways to make the creation and use of these containers easier, eg the GitHub Container Registry and GitHub Codespaces: <https://github.blog/2020-09-01-introducing-github-container-registry/>

Minor Comments

12. “Outcomes of computational approaches are occasionally difficult to reproduce”
 - a. I would say often! It’s grim out there:
 - i. <https://journals.plos.org/plosgenetics/article?id=10.1371/journal.pgen.1010210>
 - ii. <https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.3001077>
 - iii. <https://pubmed.ncbi.nlm.nih.gov/27331905/>
 - iv. <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0205898>
13. “preprint repositories (e.g., bioRxiv, medRxiv)”
 - a. Would add arXiv to this, as it’s the original (established 1991)
14. “Over the last decade, FAIR data management has been introduced to ensure availability”
 - a. Define FAIR acronym.
15. Would be useful to discuss how controlled ontologies may be helpful as well, for both biological concepts (MONDO, Human Phenotype Ontology, Uberon) and technical steps (antibodies, protocols).
16. Machine readability of results is also important. So much data is locked away in PDFs, or unparseable Excel sheets.
17. “Perhaps the most significant challenge to overcome for routine usage of initiatives such as ENCORE is the lack of incentives to sufficiently motivate researchers to spend sufficient time and effort on reproducibility.”
 - a. Indeed, would like to hear some ideas on how to improve this situation.
18. GitHub Releases lets users store intermediate sized data in a version-controlled way. This Releases feature can be combined with Zenodo and allow you to assign DOIs to GitHub repos. <https://docs.github.com/en/repositories/archiving-a-github-repository/referencing-and-citing-content>
19. “In addition, the size of the data and results may be too large to host on GitHub.”
 - a. GitHub Large File Storage is an option, but personally I always found this a bit tricky to work with. <https://docs.github.com/en/repositories/working-with-files/managing-large-files/configuring-git-large-file-storage>
20. “This approach is documented in detail in the ENCORE Step-by-Step guide (Supplementary File 1) and takes less than 30 minutes to complete”

- a. That's good to highlight, time investment is the number one thing stopping many people from pursuing good reproducible practices. Recording user-friendly youtube videos at differing lengths and levels of detail could also help, as it offers users an even lower bar of entry to get them started. It also gives you a chance to give them your hook for reproducible practices if they didn't take the time to read your paper or documentation.
- 21. I think it will be quite important to get feedback from other labs that aren't directly connected with yours. This will help to reveal any pitfalls of ENCORE, and determine what the needs are of other groups. This will be especially important when trying to understand what would motivate people most to use ENCORE, and what might drive them away.
- 22. "In September 2022, we evaluated ENCORE 3.0 to test if ENCORE had indeed improved the reproducibility of our projects."
 - a. This is a great start, and I commend you taking the time to take on this level of rigour. That said, I think an even better way to evaluate this would be to have totally independent labs run this evaluation and see if they can reproduce the results, as they won't have the benefit of being familiar with these projects peripherally (by virtue of being in the same lab). Of course, you'll need to find someone willing to do this, but it could be a valuable experience for both parties.
- 23. "One specific issue was that the lab journal often did not contain an adequate summary of (supervisory) meetings and email exchanges"
 - a. Zoom now has automated note taking that summarises the meeting. Similar integrations exist for Teams, etc. This could be one way to improve this. If you want to take this even further, you could create a hook that automatically collects these notes and pushes them to GitHub, preferably to the relevant project.
- 24. "Finally, it became clear that project organization and documentation often had lower priority than doing the actual research. For these reasons, most projects did not fully adhere to the ENCORE guidelines."
 - a. Not ideal but I appreciate the honesty and can commiserate with the difficulty of convincing others to engage in these practices. It's the classic carrot vs. stick dilemma. I find the carrot tends to work better as first line approach, by making these practices as easy and immediately beneficial to users as possible, with even greater benefits in the long run.
- 25. "Most of our ENCORE projects reside in the cloud"
 - a. Could you clarify what you mean by cloud here? Do you mean GitHub, some cloud service like AWS/Google, or an institutional cloud service?

Response to reviewer comments

We thank the reviewers for their critical yet positive and constructive feedback, which helped us to improve the manuscript and, equally important, also helped us to further shape our thinking about ENCORE. We made improvements to ENCORE and updated the manuscript based on the comments of the reviewers.

Below is our response to all the comments after some general notes and an overview of implemented improvements to ENCORE. We apologize that our response has become rather long, but the interesting questions, concerns, and thoughts raised by the reviewer prompted for an extensive answer.

We hope that we sufficiently addressed all points and we made our view more clear.

General notes

- The remarks of Reviewer 3 about the use of existing (GitHub) tools prompted us to add another Supplementary File (**Supplementary Method 4.docx**).
- We replaced 'project bundle' with 'project compendium' throughout the manuscript.
- In the revised manuscript with track changes we have indicated where we addressed specific comments of the reviewers using notation such as R1 (1), denoting comment 1 from Reviewer 1. We used different colors for each reviewer.
- The line numbers mentioned by the Reviewers below still refer to the original manuscript. The line numbers in our response refer to the revised manuscript with track changes.
- The Figures are renumbered. The numbers in our response refer to the new figure numbering.
- We switched the Result and Method sections according to Nat. Comm. guidelines.
- We updated the figures and made various textual changes to the text.

ENCORE improvements

(1) Template

Based on the comments of the reviewers we updated the ENCORE template to Version 4.1.1. We made the following changes:

- Rename 00-README-FIRST.md to README.md such that it occurs on the landing page of the GitHub Template Repository (<https://github.com/EDS-Bioinformatics-Laboratory/ENCORE>).
- Updated README.md to make a distinction between information for the '**Project Team**' running a project, and the '**Compendium Recipient**' with whom a project is shared.
- To avoid confusion about the various formats of files that are only for use by the Project Team, we removed 00_README-FIRST.txt, 2_CITATION.txt and 3_LICENSE.txt. Explained in more detail below.
- Update of 0_README.md (=Supplementary Method 3) in \Code. Now refers to 0_README.md in 0_SoftwareEnvironment underlining the importance of preserving the software environment/dependencies. We also extended the information about Software Engineering practices in the 0_README.md file.
- Update of 0_README.md in 0_SoftwareEnvironment. Now contains more detailed information and links to conda, renv, etc.
- Updated gitignore-FSS-template.txt in \Processing to exclude all Data directories.
- Release 4.1.1 has been added to the GitHub repository (<https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/releases/tag/V4.1.1>).

(2) 'GitHub Discussions' and 'GitHub Wiki' have been activated for the ENCORE template GitHub repository.

- The 'Discussions' (<https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/discussions>) now provides one possible channel to communicate about ENCORE within our group and with the scientific community at large.
 - This Discussion section currently reflects possible avenues and considerations for the further development of ENCORE. We restarted to discuss the inclusion of approaches to preserve the compute environment (<https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/discussions/12>).
- The WIKI (<https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/wiki>) contains general information about ENCORE. The ENCORE version history has been moved from 'GitHub Issues' to the Wiki.
- The WIKI now also provides information about the compatibility of specific versions of the sFSS Navigator with versions of the ENCORE template (<https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/wiki/sFSS-Navigator>). This version information is also added to the FSS Navigator repository (<https://github.com/EDS-Bioinformatics-Laboratory/FSS-Navigator/releases>). This is now referred to from the README.md file in the sFSS template root.

(3) ENCORE AUTOMATION

We created a separate GitHub repository (ENCORE AUTOMATION; https://github.com/EDS-Bioinformatics-Laboratory/ENCORE_AUTOMATION), which currently contains scripts to automatically setup an ENCORE project (including new GitHub repository) and to generate an ENCORE template for AIRR-seq analysis. We will expand this repository in the future with additional scripts to automate tasks.

(4) Update of the Step-by-Step-ENCORE guide

- Various small improvements
- Now refers to the ENCORE repository Discussion and Wiki, and to the repository ENCORE AUTOMATION
- In the instructions for setting up an ENCORE project and GitHub repository, it is now explained which steps are taken care of by the scripts in ENCORE AUTOMATION.
- We explained the compatibility between sFSS Navigator and ENCORE template versions and refer to the WIKI.
- We clarified that files that are not relevant for the Compendium Recipient can/should be removed.

Reviewer #1 (Remarks to the Author)

This manuscript introduces the Enhancing Computational Reproducibility (ENCORE) standard for sharing computational project results in a way that is both highly transparent and supports full scientific reproducibility. The manuscript clearly represents years of effort and dedication on this topic. This reviewer commends the good work! However, there are the following outstanding issues with the manuscript.

Major

1. Line 63: The description of the research life cycle in the Introduction leaves out critical steps, especially QC/QA of input data and evaluation/validation of data analysis results or software, depending on the research product produced.

Very true. Generally, we consider this to be part of “Computational approaches” (Step 3 in Fig. 1). This has now been made explicit in the Introduction (LINE 78).

2. Line 85: Some tools that pull directly from knowledgebases and repositories also hinder reproducibility. Thus, use of certain tools in data analysis pipelines will prevent full reproducibility. We have added this to the Introduction, since it is indeed an important reason for irreproducibility (LINE 99).

3. Line 103: Must be very careful mentioning GitHub as supporting “scientific reproducibility”, which requires persistence of research results. The owner of the GitHub repository can delete the repository at any time. Therefore, technically GitHub and other code hosting websites that lack persistence do not adequately support “scientific reproducibility”.

We state in the Introduction that GitHub may “contribute to” reproducibility and transparency. But indeed, this requires that the software repository is not deleted. However, Zenodo can be authorized to archive a GitHub repository release and issue a DOI for this archive. We have added this in the Introduction. (LINE 112)

4. There is no LICENSE file included in the standardize File System Structure in Figure 4. While this file would be optional, having it in sFSS examples is an important reminder. Also, the Navigation.conf file is not described anywhere. Is this suppose to be Navigate.html as described on line 329?

We have added the LICENSE file to Figure 6 and to the same figure in Supplementary Method 2.

Indeed, the Navigation.conf file was not mentioned in the manuscript. We clarified this in the Methods section about the sFSS Navigator (LINE 474)

5. While the focus is on human readability, there should be some effort to have ENCORE support machine readability. A separate optional media_types.json file could be used to identify the media type (formerly MIME type) of each file in the sFSS. Such a file could be very useful for automated validation and reuse.

Automated validation came up in our group discussions multiple times but, so far, we did not have the time nor the capacity to implement software to accomplish such a task, which is currently difficult given the fact that none of the information provided in ENCORE is ‘machine readable’ in the sense that, for example, JSON, XML, RDF, or other standards are used. Moreover, it potentially increases the overhead/burden on the user.

We have now added a paragraph about machine readability to the Discussion (LINE 812).

It is not entirely clear what is meant by 'reuse'. However, part of our group provides routine data analyses (e.g., AIRRseq and Lipidomics) for research groups within our institute. For these projects, part of the documentation and code is always identical/similar or can be automatically generated. Therefore, this type of projects has partially standardized the documentation (e.g., README files) for reuse. This was not mentioned in the manuscript because we are preparing a separate manuscript for these type of support projects.

However, we now have added a paragraph about automation to the Discussion (**LINE 812**).

In addition, we now have started the ENCORE AUTOMATION repository that contains scripts for starting a new ENCORE project and for the generation of an ENCORE template for AIRR-seq experiments:

https://github.com/EDS-Bioinformatics-Laboratory/ENCORE_AUTOMATION

An example of an ENCORE AIRR-seq template is found at:

<https://github.com/EDS-Bioinformatics-Laboratory/ENCORE-AIRRseq-TEMPLATE>

In addition, we are currently working an automation script and a standard template for lipidomics experiments, which we can show on request.

6. There does not appear to be any ENCORE-associated software to evaluate compliance with this standard. Such a tool would be immensely useful for supporting ENCORE use and compliance. The sFSS Navigator appears to be the logical package to extend for these purposes. Also, at line 308, the instructions and template do NOT "enforce internal consistency", they only "support internal consistency".

The evaluation of compliance relates to the previous remark about 'automated validation' with the difference that machine readability may not be a strong requirement. At this stage we do not have software to evaluate compliance. It is on our todo list (see <https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/discussions/11>) but will take time to implement. We included this together with the previous response in the Discussion (**LINE 812**).

We changed 'enforce' to 'support' (**LINE 446**)

7. Be careful with guidelines on when to begin using ENCORE in an active project, which will likely depend on the type of project. Likely, ENCORE should not be used during active prototyping/testing/redesign cycles of software or pipeline development, since it would slow down the development. It seems natural to use ENCORE when software is being applied, maybe tested, but not actively being developed. Strongly encourage qualifying the recommendation of using ENCORE at the start of a project at Line 498. It really depends on the project. Also, ENCORE should not be thought of as an alternative to electronic lab notebooks, even though electronic lab notebooks could be used to populate ENCORE.

We understand the argument that the reviewer brings forward. However, in our group we decided to use ENCORE from day 1 of a project because:

- We use ENCORE as part of supervision of, for example, PhD students for which we find it important that we can easily track and check developments and choices. For example, notes of supervisory meetings become part of ENCORE;
- It may be useful (for future colleagues) to, for example, document approaches that didn't work out or alternative approaches that didn't end up in a publication;
- In our experience, if one starts documenting too late, the documentation remains incomplete. For example, important details may not be recalled be missed. Or it becomes too much work at a later stage.

We understand this is a trade-off and some rudimentary data/code/results may not be worth the time and effort for documentation, but it may not always be possible to decide upfront if this is the case. However, in practice, a researcher may decide to start documenting at a later stage of a project (which might be project dependent) but this would not be our preference.

To take the argument of the reviewer into account, we have reformulated the text in the Discussion (**LINE 572**).

We do not consider ENCORE as an alternative for an electronic lab notebook. This is also not suggested in the manuscript.

8. Line 484: why does ENCORE not consider replicability? Maybe you mean it does not “directly” consider replicability. Reproducibility can be thought of as a generalization of replicability that is not as strict. Replication implies exactness, while reproducibility implies equivalence. This is something that should be discussed, since precision may need to be mentioned in terms of reproducibility. For example, what level of precision is expected for “reproducibility”? Thus, metrics of variability of results should be encouraged in order to judge reproducibility.

The definitions of ‘replicability’ and ‘reproducibility’ unfortunately depend on the source that is cited. In the Discussion we provided the sources of the definitions used by us, which are opposite to the definitions given by the reviewer.

In the Discussion we write (**LINE 532**): “ENCORE does not consider replicability (sometimes referred to as repeatability), which is about strengthening scientific evidence through replication studies by other research groups using independent data, and experimental and computational methods.”

...and (**LINE 585**): “Computational reproducibility is concerned with the reproduction of results using the same data and computational methodology.”

In the Introduction we write (**LINE 128**) “In this paper we focus on computational reproducibility, that is the reanalysis of the same data using the same computational methods.”

Therefore, we don’t consider ‘equivalence’ and ‘precision’. Consequently, we did not change the text other than adding ‘and software versions’ to this sentence in the Introduction (**LINE 586**).

9. The following article suggests that adequately supporting “scientific reproducibility” should be used as a litmus test for deciding correction versus retraction. Adoption of this guideline by journals would strongly support scientific reproducibility. The authors may want to embellish their discussion with this idea, probably around Line 554. Hunter N.B. Moseley. "In the AI science boom, beware: your results are only as good as your data — Machine-learning systems are voracious data consumers — but trustworthy results require more vetting both before and after publication" Nature Feb 1 (2024). <https://doi.org/10.1038/d41586-024-00306-2>

Thanks for this suggestion. Nicely adds to the discussion about incentives. Added to Discussion section (**LINE 815**).

10. Line 576: Given the ubiquitous use of JSON with parsers in every significant programming language, JSON is probably a better ENCORE metadata format than YAML. Would suggest at least mentioning JSON, when mention YAML.
JSON and reference added to **LINE 537**

Reference: JSON. (2024). JavaScript Object Notation. Retrieved 28 March 2024 from <https://www.json.org>

11. Line 583: Would suggest mentioning more container platforms than just Docker, given the Docker is being replaced by alternatives like Singularity.

Singularity (currently Apptainer) and reference added to **LINE 845**

Kurtzer, G. M., Sochat, V., & Bauer, M. W. (2017). Singularity: Scientific containers for mobility of compute. *PLoS One*, 12(5), e0177459.

12. Line 588: Workflow managers have their own versioning issues, which should be mentioned. Indeed, workflow management systems (WMS), don't solve versioning problems of their (remote) components. In addition, the WMS itself may become outdated as well, remote web-services may no longer be available, or older workflows may not run with newer versions of the WMS. We have added this to **LINE 863**.

13. Explicit capture of software and package versions is likely something that should be encouraged in ENCORE and likely should be its own required metadata file within the sFSS. For knowledgebases accessed, the access dates can be used to represent version.

We very much agree with this comment. This is something we encourage and are currently discussing in our group (e.g., <https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/discussions/12>). In the Discussion, we mentioned the use of Conda, Docker and virtual machines to handle software versions and dependencies. We also mentioned that "*We are currently investigating how to best approach this within the context of ENCORE*" (**LINE 848**). See also <https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/discussions/12>

We are also actively using the renv package to capture R package versions. We have added this and a reference to **LINE 849**. We did not explicitly mention the capture of (external) database versions. We have added this at **LINE 112**.

Ushey, K., & Wickham, H. (2024). renv: Project Environments. Retrieved 28 March 2024 from <https://rstudio.github.io/renv>.

Note: we are preparing a manuscript describing a use-case of ENCORE, in which we will discuss our experience with renv and the reproduction of results of a specific project by a colleague from a different department who was not involved in the project.

14. Line 598: Given that a list of entry IDs can be used to pull data from a knowledgebase or repository in an automated manner, it feels like ENCORE could handle remote datasets if an entry IDs file is considered part of the input data. This may not be what the authors intended, but it is a logical approach or "hack" to use, depending on your perspective. If analyses are sequential, then an entry IDs approach would also minimize the local storage needed.

Any data should be accessible by the software that uses it. This 'minimal requirement' does not require that the data is stored within the sFSS. In practice, one might indeed download the data from external (non-public) repositories into the sFSS when needed for processing. However, this increases the risk that at a certain moment data may no longer be available. Moreover, in these cases one should take care that no manual/undocumented steps are required (e.g., subset

selection) prior to the use with the software, since this may cause difficulties in reproducing results. This was mentioned in the Introduction (**LINE 132**). Therefore, the philosophy of ENCORE is to have everything locally in the sFSS, if this is not restricted by the research infrastructure.

We rephrased the text in the Discussion (**LINE 868**):

“Finally, ENCORE requires that all data used by the computations are within the sFSS structure. For large datasets this implies that sufficient storage space must be available on the computer that hosts the project, which can be a local (private) desktop computer, computer servers of the research institute, or remote (cloud) compute systems (e.g., as provided by Amazon Web Services; **Figure 3**). If data storage within the sFSS is not possible then documentation and/or software should be available to retrieve the data from another persistent location, and to ensure that this step does not break reproducibility.”

15. Line 676: It may be better to create separate augmented ENCORE specifications for specific types of computational projects, instead of creating one monolithic ENCORE specification. Would allow separate working groups to develop these for specific purposes, without the limiting each other.

This is an interesting suggestion. In practice this depends on the level of detail of the ENCORE specification, which is now very general. But in the context of the guidelines/standards mentioned in the Discussion, this would make sense. On the other hand, so far, we tried to avoid imposing any specific standard or software tool to leave this kind of flexibility to the researcher.

We added the following line to the Discussion (**LINE 1021**): “This may require the development of ENCORE specifications tailored for different types of computational projects by different specialized working groups.”

Minor

- Line 101: recommend mentioning other code revision tracking systems like Bitbucket from Atlassian.
- Line 117: add comma after “In this paper”.
- Line 134: change “FAIR requirements” to “FAIR guiding principles”.
- Line 441: add comma after “In our view”.
- Line 475: add comma after “remote computer system”.
- Line 483: suggest adding “scientific” before “reproducibility”. – **NOT CHANGED**
- Line 483: change “allow harmonization within a research group.” to “improve harmonization within and across research groups.”
- Line 495: use a colon after “computational project” instead of a comma.
- Line 538: something feels grammatically incorrect in the sentence starting with “There are clear advantages for working reproducibility...”.

We corrected all the minor issues.

Reviewer #2 (Remarks to the Author)

The paper describes a very interesting approach to improving the reproducibility of computational workflows (called "ENCORE"), with (implicitly) a focus on computational pipelines that involve large amounts of data such as those occurring in bioinformatic analyses in biology and medicine. As such, the paper does not contain any new research or research results - its novelty lies in the infrastructure it provides to (potentially) the research community. The paper is very well-written and easy to follow. A comprehensive guide to the use of ENCORE is included as supplementary Information.

Whilst it is undoubtedly the case that were such a system to be adopted universally by the academic publishing industry then some of the issues around reproducibility would be greatly reduced. However, as I explain further below, I am not convinced that the approach advocated in the paper is generally applicable, or that it would be widely adopted. I also think that the issues addressed are not the primary cause of the lack of reproducibility of computational research. However, the work described does make a strong contribution in terms of the analysis of the problem, and in terms of the solution for certain types of research. Reframing the paper in terms of this wider contribution would, for me, make it relevant to a much wider audience.

It is good to hear that the reviewer believes that our problem analysis is useful. Yet, there are other papers (some of which we refer to in our manuscript) that also made problem analyses. We believe that our main contribution is a practical approach towards reproducibility and not another general paper about reproducibility. We believe that it is our practical approach that makes our work of interest to a wide audience. Therefore, we have not reframed the paper as suggested, but we hope to have sufficiently addressed the comments in this rebuttal and in the revised manuscript to convince the reviewer about our view.

The reviewer is *"not convinced that the approach advocated in the paper is generally applicable, or that it would be widely adopted"*. We believe our approach is quite generic because it does not depend on any software tools while at the same time it can be used with most software tools of choice (see also the new **Supplementary Method 4**). However, ENCORE has not yet been put to practice outside our group, but this is on our roadmap (we come back to this below). It might well be that ENCORE will not be widely adopted but we will continue to promote and improve our approach. Publication in a journal with a wide audience from a broad range of scientific disciplines will certainly help us with this.

Below, we address the specific comments of the reviewer.

Generalising the approach

1. The approach described has been 6 years in the making. It appears to have been used on 20 projects within the host organisation and supports 50 projects by "other groups". The paper is (laudably) very candid about the complexities and difficulties of developing and using such a system, and the ENCORE guide in the supplementary information is equally open about the technical difficulties and limitations of the current version. The paper is written in an application-agnostic style, but the system appears to be designed to support bioinformatics "pipeline-style" projects.

The projects conducted within our group consist of both bioinformatics/omics projects and projects involving mathematical modelling (e.g., differential equations). This implies that ENCORE has indeed been developed from the perspective of a restricted research domain and a specific group of researchers. It is important to note, that most of the research projects for which we use ENCORE are not "pipeline-style" projects. We clarified this in the Discussion section (**LINE 717**).

2. Reading both the paper and the Guide highlighted very well for me the issues in developing such a system, but the complexities and caveats of implementation and use suggested to me that to extend this out to other areas would require substantial work.

We believe that our approach is quite domain independent and would not require additional work to be used in other contexts.

However, in most cases, adopting ENCORE would require researchers to change their project organization, which might be a barrier (added to **LINE 727**). Nevertheless, the use of ENCORE by other research groups will undoubtedly give new insights and improvements to ENCORE. In response to Reviewer 1's suggestion, it may be beneficial to establish different working groups to further develop ENCORE if domain-specific standards need to be included (**LINE 1021; R1 (15)**).

So far, we did not involve other research groups, because we were still gaining experience ourselves resulting in changes to the setup. Within our group, we organized several evaluation sessions (Figure 2) and, currently, we are taking initial steps to evaluate and further develop ENCORE in collaboration with other research groups (added to **LINE 727**).

For information: in this context, three new project proposals are relevant:

- A project proposal for a Dutch Open Science call to further develop ENCORE with two other groups was written but due to personal circumstances not submitted. (NWO; <https://www.nwo.nl/en/researchprogrammes/open-science/open-science-fund>). We now aim to submit this proposal in the new Dutch OpenScience.nl program (<https://www.openscience.nl/en>)
- An MSCA EU proposal (<https://marie-sklodowska-curie-actions.ec.europa.eu/actions/doctoral-networks>) involving a larger consortium in which ENCORE is part of the advocated Open Science approach. Will re-submitted in November 2024.
- An NWO Dutch Research Infrastructure proposal was submitted in which ENCORE will be part of the lipidomics research infrastructure (<https://www.nwo.nl/en/researchprogrammes/research-infrastructure-ri-national-consortia>).

Thus, we started efforts to further disseminate/test ENCORE.

3. I did really like the principles on page 5 of the guide and felt that a paper that highlighted the principles and used the ENCORE system to illustrate in a particular application domain how these principles could be met would be a much stronger contribution than trying to suggest that other researchers might adopt ENCORE as is.

In the manuscript we don't (explicitly) try to convince other groups to adopt ENCORE, but of course we hope that other groups will adopt and help to improve ENCORE. We also hope that our implementation gives new impetus to thinking about reproducibility.

We believe that the manuscript is already in agreement with this suggestion of the reviewer but in a slightly different way: we defined eight "main requirements" in the Methods section (including adherence to proposed published guidelines; point 4, **LINE 336**). Our manuscript describes how these requirements can be met in practice using ENCORE.

The four 'principles' in the Step-by-Step guide (page 5) only provide further practical instructions for the user to meet the eight main requirements and to make optimal use of ENCORE.

Thus, ENCORE is not a system to illustrate how the 'principles' (user instructions) can be met but how the 'main requirements' can be met in practice. In other words, the principles in the Step-

by-Step guide follow from our main requirements. It is exactly because of this that we believe this manuscript is of interest to a wider audience.

We now made this point clearer in the Introduction (LINE 143). We now also explicitly refer to the user guide Principles from the Methods section (LINE 353). Finally, we briefly mention it again in the Discussion section (LINE 684).

4. This could then contribute to a wider debate with the academic publishing industry about how the industry itself might develop appropriate information platforms to support reproducible research. I have written about this previously and so won't repeat the arguments here about why I think this should be the responsibility of publishers not researchers.

ENCORE was not primarily developed to serve as a supplement to a final publication (LINE 185), but also, for example, for supervisors to provide timely and more constructive feedback to PhD students during a project (LINE 765). However, the reviewer has an interesting view on the role of the publishing industry in making research reproducible (Gavaghan, 2018). In the Discussion we briefly mentioned several initiatives of publishers (LINE 945) to which we have now added the reference to (Gavaghan, 2018) and the following text to the Discussion (LINE 955):

"It has also been suggested to rethink the concept of a "journal" as a community-driven information repository containing, among others, the data and software, to enable reproducibility, reuse, and comparisons (Gavaghan, 2018). In this scenario, academic publishers would have a key role in stimulating (standard-based) approaches to research dissemination. We believe that such an effort should be a joint undertaking of research communities and publishers aiming to improve reproducibility".

Gavaghan, D. (2018). Problems with the Current Approach to the Dissemination of Computational Science Research and Its Implications for Research Integrity. *Bull Math Biol*, 80(12), 3088-3094.

Wider problems with reproducibility of computational research

5. Whilst the creation and support of information sharing platforms such as ENCORE is very helpful in moving us towards reproducible computational research, and may be the whole solution if all of the computational tools being used have been shown to be fully tested and reliable, in my view in most of computational science the biggest problem is that we have no way of knowing whether the code being used to generate the computational results is doing what the researchers think it is doing. This is because most computational software is not refereed, most software is not developed using sound software engineering principles (since researchers developing that software are rarely trained in software engineering), and computational software is often not made openly available (or where it is, it is not accompanied by appropriate documentation and installation guidance). These issues are discussed briefly in the penultimate section of the discussion, but I think should be brought forward into the introduction of the much more general paper that I suggest above discussing the principles underpinning the development of ENCORE. I would add a principle on software engineering practices, and discuss how an information sharing platform might implement simple checks on software quality (and in particular on software testing).

One reason for developing ENCORE is to ensure that software can easily be made available, shared, and executed. Our guidelines mention to clearly document how software should be executed.

Lack of good software engineering practices, review and testing of code are serious issues that are, indeed, only mentioned briefly in the Discussion section (**LINE 887**). It is a well-known problem that is discussed in much more length and depth in other papers (some of these were referred to in the manuscript). Clearly, also in the context of ENCORE, good software engineering practices are very important for transparency and detecting (conceptual) errors as was mentioned in the Introduction and Discussion. However, at this moment, ENCORE does neither enforce nor use tools for e.g., (automated) documentation or testing and only provides a limited and still incomplete set of software engineering guidelines in one of the README files, which we completely revised (**Supplementary Method 3**). However, currently it is left to the researcher to follow best practices and to use tools of choice to assist in this.

We can only agree with the reviewer that also this problem needs to be addressed by training and (automated) quality checks and testing (as we already argued in the Discussion). Implementation of best software engineering practices as part of ENCORE has been under discussion but at this stage, we are uncertain how to best approach this in the context of ENCORE. One issue being that not everyone is trained in software engineering best practices and that we are (still) reluctant to enforce specific tools, IDE's, etc. (See also <https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/discussions/15>). Also because of ENCORE's requirement to not rely on specific tools.

We also would like to remark that although sub-optimal software engineering practices are a problem and contribute to irreproducible results, it is neither the task nor aim of ENCORE to solve this problem. However, we will further improve the ENCORE requirements and guidelines, and create more awareness about best practices and availability of software engineering tools.

As mentioned earlier, we prefer to not completely reframe the paper. We already mentioned the problem of software engineering in the Introduction (**LINE 119**) but have now extended the Discussion paragraph on software engineering (**LINE 885 and 893**) to emphasize the point of the reviewer. We added:

“One resulting problem is that we have no way of knowing whether the code being used to generate the computational results is doing what the researchers think it is doing. This is one reason why ENCORE proposes to start organizing and documenting from the start of a project, since this increases the chance that conceptual errors or software bugs are detected in an early stage by the researchers or their supervisors.”

“The use of integrated development environments, automated quality checks, and (unit) testing would also help to improve software (Ebert et al., 2022). In addition, Large Language Models may increasingly play a role in software development, testing, and documentation (Copilot, 2024; Wang et al., 2024).”

In the new **Supplementary Method 4** we now also mention GitHub functionalities that also assist in better software engineering practices

Ebert, C., Bajaj, D., & Weyrich, M. (2022). Testing Software Systems. *IEEE Software*, 39(4), 8-17.

Wang, J., Huang, Y., Chen, C., Liu, Z., Wang, S., & Wang, Q. (2024). Software Testing with Large Language Models: Survey, Landscape, and Vision. *IEEE Transactions on Software Engineering*, 50(4), 911-936.

Conclusion

6. Whilst the manuscript highlights very well some key problems with reproducibility, and describes a really interesting approach within this lab to address this problem, the suggested approach would, in my opinion, have limited take-up even in the bioinformatics domain. A better approach for a journal such as Nature Communications would be to use the development of ENCORE to highlight lessons learnt that might be applied in a wider domain, and that might be incorporated into the data and information platforms provided by academic publishers. This would, I think, appeal to a much wider readership.

We appreciate the honest comments of the reviewer, which we certainly can relate to. We are the first to admit that a broad adoption of ENCORE or similar initiatives (in the bioinformatics / computational biology) domain is not going to happen without further effort. However, we can only iterate and emphasize what we wrote at the beginning of this rebuttal: *"It might well be that ENCORE will not be widely adopted but we will continue to promote and further improve our approach. Publication in a journal with a wide audience from a broad range of scientific disciplines will certainly help with this."* The future will tell how successful this will be.

The reviewer proposes to rewrite the manuscript with a different emphasis and highlighting the lessons learned. However, our aim was to implement a practical approach based on lessons learned from other initiatives (that is, the work cited in the manuscript). This is not to say that there are no lessons learned from ENCORE, but we discussed these in the manuscript. To emphasize the most important lessons learned we have summarized these in five points at the beginning of the Discussion. **(LINE 698).**

We also fully appreciate and support the idea of the reviewer to have publishers play a key role in the development of an information platform. But this is probably also not an easy road to travel and might not be happening soon. A paper about previous and current efforts in reproducibility and (FAIR) data management in the context of a next step toward such platform is a compelling idea, but this was not the aim of the current project/manuscript.

ENCORE contributes not only a concrete approach but also experiences needed to improve transparency and reproducibility. From this perspective, the manuscript is relevant for the wider readership of Nature Communications. We also believe that ENCORE is a quite generic approach because it does not depend on any software tools but at the same time can be used with most software tools of choice and this, we feel, is one of the reasons that it might be compelling to a wider scientific audience.

David Gavaghan, Oxford.

Reviewer #3 (Remarks to the Author)

Summary

The authors propose ENCORE as template for enhancing scientific project reproducibility. While I applaud these efforts towards reproducible science, I think this work could use significant improvement in several key areas.

1. First, as ENCORE is GitHub-based it would make sense to leverage the breadth of tools for reproducible research already integrated into the GitHub environment. I feel that some opportunities have been missed to build upon features such as:

- Wikis: <https://docs.github.com/en/communities/documenting-your-project-with-wikis/about-wikis>
- Discussions: <https://docs.github.com/en/discussions>
- Projects: <https://docs.github.com/en/issues/planning-and-tracking-with-projects/managing-your-project/managing-project-templates-in-your-organization>
- Codespaces: <https://docs.github.com/en/codespaces/reference/using-github-copilot-in-github-codespaces>
- Issues templates: <https://github.com/devspace/awesome-github-templates>
- GitHub Actions: <https://github.com/features/actions>
- GitHub Container Registry: <https://github.blog/2020-09-01-introducing-github-container-registry/>

The reviewer raises a valid concern about ENCORE: our approach seems to neglect many tools that contribute to reproducible research. In particular, the reviewer points to a range of GitHub tools.

This concern led us to write an additional supplementary file (**Supplementary Method 4**), where we extensively explain our view on this. Supplementary Method 4 should therefore also be seen as part of our response to this concern of the reviewer.

We also need to make the following remarks:

- ENCORE is not primarily GitHub-based. It is only used for code versioning. If it was GitHub-based, it would indeed make sense to fully utilize the GitHub functionalities mentioned by the reviewer by design.
- To avoid misunderstanding about what is synchronized with GitHub:
 - The complete file structure (sFSS), not GitHub, is the hub of the project. The sFSS can be shared with peers and contains much more than only the code.
 - Sharing an associated GitHub repository is optional (see the revised Figure 4).
 - In the manuscript and in Figures 4 and 5, We explained that only part of the \Processing (i.e., the code, Notebooks, and code documentation) is synchronized with GitHub.
 - There are several reasons why we do not host the full sFSS structure on GitHub (see **Supplementary Method 4**).
- ENCORE does not neglect other (GitHub) tools that improve reproducibility. However, by design, it does not impose the use of specific tools except for the basic functionalities of Git/GitHub. Instead, the choice to use specific (GitHub) tools is left to the project team. This was not emphasized sufficiently in the ENCORE documentation. We now emphasize this:
 - In the main manuscript. Discussion section **LINE 721**.

- In the Step-by-Step User guide (Section 5)
- Software Environment: see 0_README.md in \Processing\0_SoftwareEnvironment
- ENCORE WIKI: <https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/wiki/Complementary-Tools>
- Software engineering: see 0_README.md in \Processing\NameOfComputation_1\Code

We acknowledge the value of the GitHub features mentioned by the reviewer and recognize that GitHub offers more features than we currently use. We agree that these features may further support reproducibility. We provide a more detailed discussion about the individual GitHub functionalities in the context of ENCORE in **Supplementary Method 4**. We copy the summary here:

With respect to the use of complementary tools our view (in the context of ENCORE requirements 1, 5, 7, and 8) is as follows:

1. Any platform designed to support reproducibility, including ENCORE, should avoid imposing significant restrictions on researchers (e.g., in terms of use of software tools). Excessive constraints may be perceived by individual researchers as too disruptive, leading to a platform that will not be adopted by the community.
2. ENCORE is designed to accommodate various styles of working and to be compatible with a large range of software tools. Consequently, it does not impose the use of any specific software tool except for Git/GitHub (which is easily replaced with another versioning system). However, ENCORE users are encouraged to utilize complementary tools that enhance reproducibility, including those for (i) preservation of the compute environment, (ii) software development, (iii) workflow management, (iv) (software) documentation, and (v) project management. In fact, some of these tools are essential for improving reproducibility but currently it is left to the individual researcher to make appropriate choices.
3. The first ENCORE requirement (a single self-contained project compendium) excludes the use of external of (cloud-based) tools/platforms to host project documentation, project discussions, data, etc but do not allow to synchronize/download this information into the ENCORE project. In addition, for project documentation we prefer to store documentation in the appropriate ENCORE subdirectories.

Since ENCORE relies on Git/GitHub for software versioning, it raises the question of how other functionalities of GitHub can complement ENCORE or whether the GitHub platform could serve as an alternative to ENCORE. We summarize our evaluation in the following points:

4. Git/GitHub has been developed for software development and version control of text-based documents, while the focus of ENCORE is on computational research which includes but goes beyond mere software development. ENCORE, therefore, includes much more than code, and using the GitHub platform as a replacement for ENCORE could be considered misuse of the GitHub platform.

5. GitHub offers numerous functionalities that could enhance reproducibility. However, there may be limitations depending on the type of project. For example, projects with high CPU or storage requirements may face functional constraints and incur additional costs. Additionally, not all GitHub functionalities align with the first ENCORE requirement of a single self-contained project compendium.
6. A researcher may prefer the use of (free/cheaper) local/national compute infrastructure, or there might be data privacy and confidentiality issues that prevent the use of GitHub for large-scale calculations and/or data storage.

Therefore, it is up to individual researchers to decide which GitHub functionalities to use in conjunction with ENCORE for their specific projects.

I hope this makes clear our view. We don't feel that we missed opportunities but took a different approach to stay generic. In addition, we made this more explicit to the ENCORE user as outlined above.

2. Similarly, I understand the intent of providing templates in multiple formats is to let users choose what format they're most comfortable with. But as it's currently implemented, I think these duplicate files may unintentionally add to confusion of which version is the one being used (unless the users are diligent about deleting unused versions of files, which I tend not to assume).

Yes, this can be confusing when using ENCORE for the first time.

It is important to distinguish between (i) the Project Team (Figure 4) who executes a project and is responsible for organizing and documenting according to ENCORE guidelines, and (ii) the 'Compendium Recipient' (e.g., peer, reviewer, Zenodo). According to ENCORE guidelines the 'Compendium Recipient' will never see the multiple formats. We now clarified this in various places and updated the ENCORE template.

- In the Step-by-Step guide (**Supplementary Method 1**) we explicitly mentioned that files and directories that are not used should be removed by the Project Team to avoid that the Compendium Recipient has to browse/open empty directory/files. We have added that also files that are not relevant for the Compendium Recipient should be removed (when sharing).
- We have updated the **00_README-FIRST.md** file of the ENCORE to better make the distinction between the Project Team and Compendium Recipient, and to include instructions for file removal.
- We changed **00_README-FIRST.md** to **README.md** such that it occurs on the landing page of the ENCORE template GitHub Repository.
- This explanation is also added to **Supplementary Method 2** (explanation of the predefined files)
- In addition, we removed **00_README-FIRST.txt** since there is no longer a need for it. We removed **2_CITATION.txt** and **3_License.txt** since these are not relevant for the Compendium Recipient (no need for multiple formats).
- We now made this distinction explicit in the Methods section (**LINE 402**) and changed **Figure 4** and **6** accordingly.

A Project Team who uses ENCORE for the first time should acquaint themselves with the ENCORE guidelines, but this will become routine and a 1-minute-job for subsequent projects.

3. Instead, it would seem a more natural solution to this is something like Quarto or Jupyter (which can accommodate many languages) or Rmarkdown (which is primarily for R but has extended to python and other languages in recent years), where the same script can generate multiple output formats (md, HTML, latex) by simply changing one line in the yaml header. This notebook approach also enables documentation, figures, and tables to be embedding directly next to the code in the same document which goes a long way towards reproducibility. As an example, tools such as BiocBook demonstrate this very nicely, all bundled within a fully containerised environment with CI/CD via GitHub Actions: <https://github.com/js2264/BiocBook>

We use Jupyter and R Markdown and we are aware that these tools can export multiple output formats. However, as explained in the previous point, the project team decides on the file format(s) used and removes the formats are not used in a specific project. These will not be seen by the Compendium Recipient. Thus, ENCORE does not try to provide all information in different formats. Instead, we provide multiple formats to the project team to make a quick start. We clarified this point:

- We modified the text of the section ‘Instantiation of a new ENCORE-based project’ (**LINE 402**).
- We added this information to **Supplementary Method 2**.

We agree that Notebooks as provided by Jupyter (Rule, 2019) or R Markdown (Siraji, 2023; Xie, 2018) provide a convenient way to have code, documentation, results, and data together. This certainly contributes to reproducibility and BiocBook is a nice example. However, notebooks also have drawbacks. E.g., the JSON format of Jupyter Notebooks (Rule, 2019) and the inclusion of binary blobs and metadata makes versioning more troublesome. In addition, their scalability and performance are limited. Therefore, Notebooks are very suitable for interactive and exploratory small-scale data analysis tasks, but less suitable for, for example, running large compute-intensive analyses and simulations.

The information/documentation in ENCORE goes well beyond what one would or can store in a Notebook (e.g., PowerPoint presentations, PDF documents, project-specific background information, large datasets, metadata). Note also that ENCORE is used to track/monitor a project (e.g., by supervisors of a PhD student) during the execution of a research project and not only meant to accompany a final publication.

ENCORE, does not exclude the use of Notebooks, and in part of our projects we use R Markdown or Jupyter notebooks. However, it is left to the researcher to decide if a notebook is suitable for the project at hand.

To not further increase the length of the manuscript we did not add this discussion about notebooks to the manuscript. However, if the reviewer prefers we can of course further elaborate in the Discussion.

Siraji, M. A., & Rahman, M. (2023). Primer on Reproducible Research in R: Enhancing Transparency and Scientific Rigor. *Clocks Sleep*, 6(1), 1-10.

Xie Y., Allaire J.J., Golemund G. R Markdown: The Definitive Guide. Chapman and Hall/CRC; Boca Raton, FL, USA: 2018.

To their credit, the authors do mention the desire to implement containerisation in the future. But in a world where dependency conflicts and installation issues is an almost universal barrier to

reproducibility, I feel some degree of containerisation (or at very least a conda environment) is a minimum prerequisite to make ENCORE substantially impactful.

In practice, we make use of Conda environments for Python-based projects and renv for R-based projects. For part of the projects we use VMs, sometimes Docker. However, this is not yet part of ENCORE by design. As stated in the Discussion, we are investigating the best approach towards using container platforms (and virtual machines) in the context of ENCORE. In other words, so far, it was left to the researcher to decide what to use.

Supplementary Method 2 also mentions the use of conda, docker, etc (at the description of the \Processing\0_SoftwareEnvironment subdirectory). In the Step-by-Step guide we refer to the 0_README.md file in \Processing\0_SoftwareEnvironment, which provides some basic information about conda, etc.

For this moment, we made the following updates:

- In the ENCORE template, the 0_README.md file in \Processing\0_SoftwareEnvironment is updated and provides more information and added links to pointers to conda, renv, docker, and similarity, and VMware.
- In this README file we now also state that the use of Conda, renv, or similar is a minimal requirement.
- In the ENCORE template, we updated the 0_README.md in \Processing\NameOfComputation_1\Code (and Supplementary Method 3) in which we now explicitly refer to the 0_README.md file in 0_SoftwareEnvironment
- Discussion about containerization/VM/etc added to <https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/discussions/12>.

To some extent, many of the features proposed in ENCORE feel like a more cumbersome reinvention of the existing tools. I say this not to discourage the authors (I think their efforts have value!), but rather to point to resources that may make ENCORE more appealing and valuable to the broader community. Pending significant improvements, I think ENCORE could evolve into a valuable resource for the community that makes reproducible practices easier and more appealing. I would be happy to review a second round if these issues can be addressed.

We understand the point the reviewer is making, but the main idea behind ENCORE was to reuse and translate existing ideas/guidelines about reproducibility into a practical approach to support project organization, reproducibility and transparency while at the same time not dictating researchers which tools to use since this would be met with too much resistance. In other words, and as explained under point (1) of the reviewer, we believe that incorporating specific tools would make it less attractive to the broader community.

Major Comments

1. "Another option is to generate a small program (script) that retrieves the sFSS template from the ENCORE GitHub repository and automatically fills in certain portions of the required documentation, which is particularly useful if one conducts projects with a certain degree of similarity."
 - a. Great! I think this is a key selling point for ENCORE. To me, automation is one of the most effective ways of convincing people to use reproducible practices, ideally with as little effort as possible as the automation gets better and is more extensive.
We added a section about automation to the Discussion (LINE 1048)

- [about-projects/about-projects](#)
 - ii. <https://docs.github.com/en/discussions>
 - b. For example, you can provide templates for GitHub Projects layouts, or templates for Issues.
 - i. <https://docs.github.com/en/issues/planning-and-tracking-with-projects/managing-your-project/managing-project-templates-in-your-organization>
 - ii. <https://docs.github.com/en/communities/using-templates-to-encourage-useful-issues-and-pull-requests/configuring-issue-templates-for-your-repository>
 - iii. See here for some examples of Issues templates that you may find helpful:
<https://github.com/devspace/awesome-github-templates>

We addressed this point under **point (1) of the reviewer** and, more specifically, in **Supplementary Method 4**, Sections 4.3.4, 4.3.5 and 4.3.6.

3. “One decision to be made by the project team is how to distribute the documentation over the various README files and the lab journal(s). However, as a rule one should document any project file (code, data, results) in the directories in which these are located. The lab journal can then be used for more general documentation.”

GitHub Wikis seems like it would be a useful solution for exactly this, as it's all based on a series of markdown files: <https://docs.github.com/en/communities/documenting-your-project-with-wikis/about-wikis>

We addressed this point under point (1) of the reviewer and, more specifically, in Supplementary Method 4, Section 4.3.8.

We have now created a WIKI for the ENCORE template GitHub repository to provide basic documentation (<https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/wiki>) and to the new ENCORE-AUTOMATION repository (https://github.com/EDS-Bioinformatics-Laboratory/ENCORE_AUTOMATION/wiki).

4. Notes on the ENCODE GitHub repo: <https://github.com/EDS-Bioinformatics-Laboratory/ENCORE>
 - a. I think it would be helpful to have the getting started instructions in the landing page README. I see there's a file named 00_README_FIRST.md, but it's easy to gloss over this at first. Having a README that appears on the landing page in the template makes this more obvious.

We updated the ENCORE template. See **point 2** in the summary of this reviewer.

- b. The file system feels a bit cluttered with duplicate versions of the same files in different formats. For example, 0_GETTINGSTARTED has four different versions! (0_GETTINGSTARTED.docx, 0_GETTINGSTARTED.html, 0_GETTINGSTARTED.tex, 0_GETTINGSTARTED.txt). Same goes for many of the other files.

We clarified this in **point 2** in the summary of this reviewer.

- c. If the goal is to provide users different choices of formats, would it not make more sense to use a single file type that can be rendered into whatever other file type they want? This is what Quarto and Rmarkdown do, and the outputs get automatically organised into a standard folder structure. This would reduce clutter and potential confusion over the duplicate files.

The Compendium Recipient will never get multiple formats. The Project Team will choose for one format and will remove the formats that they will not use. We clarified this in

point 2 in the summary of this reviewer.

5. “- Researchers that aim to reproduce the project: start by opening ****Navigate.html**** in your browser.”

Why not launch this as a GitHub Pages website, so users can start exploring this before even cloning the repo locally?

We did not make clear enough in the manuscript that there is a distinction between the Project Team and the Compendium Recipient (see point 2 in the summary of this reviewer).

The Project Team is responsible for defining the content of two key files: 0_PROJECT.md, which provides a general description of the project, and 0_GETTINGSTARTED, which contains essential information for the Compendium Recipient to begin exploring the project. This process of populating these files occurs after the Project Team has cloned the ENCORE template to initiate a new project.

The Compendium Recipient opens Navigate.html (0_PROJECT.md, 0_GETTINGSTARTED.html and the sFSS directory tree) in the web-browser to start exploring a project. Thus, navigate.html is not (primarily) meant for the Project Team to explore the template.

We have now clarified this:

- We added navigate.html to Figure 4
- We clarified the text in the section ‘Instantiation of a new ENCORE-based project’ (**LINE 402**)
- We clarified the text in the section ‘Component 4. The sFSS Navigator’ (**LINE 596**).

6. “- The FSS can be tested by executing the Navigate.py script (if Python is installed on your computer), or by downloading and executing one of the executables.”

I think this approach could be problematic as it means everyone will have their own copy of the python script. If any updates/fix need to be made to the script, every user must independently update their copy of the script. Instead, having a centralised distribution of the software, e.g. as a python module distributed via pip and/or conda, would be a more sustainable approach.

Good point.

The Python scripts and executables are distributed with the ENCORE template on GitHub and therefore come as a compatible package. If the Python script, sFSS structure, or any of the predefined files change, then we update the GitHub ENCORE template. Thus, a researcher starting a new project by cloning the template will always have the latest and compatible version of the Python script. We currently maintain only the latest version in a separate repository (<https://github.com/EDS-Bioinformatics-Laboratory/FSS-Navigator>).

For updates and bugfixes we currently do not have a specific mechanism in place, which would require to keep separate updates for any previous version of the script because newer versions of the Python script may no longer be compatible with older versions of the sFSS structure/pre-defined files.

We also cannot assume everyone is using Python and/or Conda (the reason for providing executables).

Therefore, for now, we did the following:

- Updates of the Python scripts and executables for different versions of the ENCORE template will be provided through 'Releases' of the Navigator repository. These releases also document the compatibility with template versions. We now released Version 1.5. <https://github.com/EDS-Bioinformatics-Laboratory/FSS-Navigator/releases>.
 - Template versions and associated Python versions are now also documented in the new ENCORE wiki (<https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/wiki>)
 - We refer to this information from the top-level README.md file of the template, and from the Step-by-Step user guide.
7. I would like to see a lot more work done on automating many of the processes involved in setup and maintenance of the ENCORE framework. Automation takes away a lot of tedium that drives people away from reproducible practices and makes it more likely they'll bother to follow the guidelines.

We agree that automation can support reproducible practices. In fact, we already do this for B-cell/T-cell repertoire sequencing experiments in which we use the same data analysis workflows for different datasets (= different projects). For these projects we have a specific ENCORE template that already contains the code and part of the documentation as a starting point. The modified code will end up in a separate repository branch. We are currently setting this up for lipidomics experiments as well. We also made scripts to automatically instantiate an ENCORE project and corresponding GitHub repository. We have added this information to the Discussion section (**LINE 1049**) and to the Step-by-Step guide (**Supplementary Method 1**).

The automation in terms of the generation of code documentation (e.g., GitHub Copilot, ChatGpt, Sphinx) or writing unit tests (e.g. Copilot) is not excluded by ENCORE but also not explicitly included in the specifications. However, we have extended the Discussion paragraph on software engineering and mention the use of such tools (**LINE 894**).

8. "For example, recently detailed guidelines for a standardized file system structure for scientific data were published by Spreckelsen and co-workers (Spreckelsen et al., 2020), which inspired the sFSS used in ENCORE. However, they use a different organization, i.e., the top-level of their file system layout is not an individual project like in ENCORE but an experiment, simulation, data analysis, or publication."
- a. Given the similarity, I think this warrants a much more detailed discussion of the strengths of ENCORE over the prior work. This should at least be mentioned in the introduction as well.

We now mention this in the Introduction (**LINE 146**) and have added the following text to the Discussion (**LINE 964**).

"The use of directory structures to organize research projects has been proposed in the past, However, in general these initiatives do not provide specific templates or are limited to a specific programming language (Marwick, 2018, Noble, 2009; Wilson, 2017). In particular, the approach presented by (Marwick, 2018) is useful for R-based projects and uses the 'rrtools' package to setup a project compendium suitable for writing a reproducible manuscript. It supports Quarto (an open-source scientific and technical publishing system), Docker, package versioning using renv, and integration with GitHub Actions. "

"ENCORE is inspired by the standardized file system structure that was proposed by

Spreckelsen and co-workers (Spreckelsen et al., 2020). Their file system structure comprises four top-level directories denoted as categories (Experimental data, Simulations, Data analysis, and Publication). Each of these categories contains subdirectories to hold specific projects in which other subdirectories may exist. This implies that, in practice, each of the four top-level directories will contain subdirectories and files related to multiple projects. The subdirectories and files are manually annotated and connected using YAML headers (key-value pairs) in README markdown files, which allows, for example, to trace simulations and data belonging to a specific project. The ENCORE sFSS is project-oriented, which facilitates sharing with peers without first having to reassemble a project. At the same time this reduces the need to manually annotate and define relationships, since these are implied by the sFSS structure. Nevertheless, ENCORE would also benefit from integration approaches such as YAML to improve transparency and reproducibility. However, ENCORE does not yet require the use of YAML since it may require too much effort from researchers to specify and maintain. Similar to ENCORE, the approach of Spreckelsen et al. has a strong focus on transparently organizing and linking data, code, and publications to support reproducibility. However, there are no further requirements except that one should be able to re-run code from within their file system structure. In addition, there are no requirements by design for further user and code documentation and the use of GitHub. Compared to the approach of Spreckelsen et al., ENCORE also aims to translate published guidelines into specific instructions and templates in, for example, the README markdown files to guide researchers in making their computational work reproducible. This is supported by providing our sFSS as a template together with predefined files while. In contrast to the file system structure proposed by Spreckelsen, the ENCORE sFSS is much more detailed. ENCORE goes beyond a structured file system and through regular internal and external evaluations we aim to gradually improve the ENCORE approach to improve reproducibility. “

Marwick, B., Boettiger, C., & Mullen, L. (2018). Packaging Data Analytical Work Reproducibly Using R (and Friends). *American Statistician*, 72(1), 80-88.

Noble, W. S. (2009). A quick guide to organizing computational biology projects. *PLoS Comput Biol*, 5(7), e1000424.

Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L., & Teal, T. K. (2017). Good enough practices in scientific computing. *PLoS Comput Biol*, 13(6), e1005510. <https://doi.org/10.1371/journal.pcbi.1005510>

- b. Have there been other proposals like this? If so, would be good to talk about them, what they contributed. It's also worth studying them to examine why they may not have caught on as intended, and what ENCORE can learn from those mistakes.

See **point (a)** above

It is difficult to determine to what extent and how the approaches by Spreckelsen et al. or Marwick et al. have been used by other groups.

- c. One that comes to mind is more R-based but nevertheless may be informative: <https://ro.uow.edu.au/smhpapers/5389>

References added; See **point (a)** above

9. For the code scripts, I think there's a good argument to use Quarto files which can accommodate a number of different programming languages and provide lots of ways to document code in-line. Check out BiocBook for a good example of how this can be taken even further and create a Docker container with all dependencies. BiocBook is more geared towards R users but could definitely be extended to any programming language. <https://github.com/js2264/BiocBook>

This was addressed in **point 3** in the **reviewer's summary**.

10. GitHub Codespaces is another resource to potentially leverage. No downloads necessary, and you can even bypass installation with preinstalled Docker containers that users can spin up and run the code right away.

We addressed this point under **point (1) of the summary of this reviewer** and, more specifically, in **Supplementary Method 4**, Section 4.3.2.

11. "This stack comprises (i) (Bio)Conda (Anaconda Software Distribution, 2020; Gruning, Dale, et al., 2018) to provide virtual execution environments addressing software versions and dependencies, (ii) container platforms such as Docker (Nust et al., 2020) to preserve other aspects of the runtime environment, and (iii) virtual machines using cloud systems or dedicated applications such as VMware, to overcome the dependencies on the operating system and hardware. We are currently investigating how to best approach this within the context of ENCORE."

I think this will be crucial moving forward. Containerisation can make a massive difference in reproducibility and installability. And there's an increasing number of ways to make the creation and use of these containers easier, eg the GitHub Container Registry and GitHub Codespaces: <https://github.blog/2020-09-01-introducing-github-container-registry/>

This was addressed in **point 3** in the **reviewer's summary**.

Minor Comments

12. "Outcomes of computational approaches are occasionally difficult to reproduce"

I would say often! It's grim out there:

- i. <https://journals.plos.org/plosgenetics/article?id=10.1371/journal.pgen.1010210>
- ii. <https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.3001077>
- iii. <https://pubmed.ncbi.nlm.nih.gov/27331905/>
- iv. <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0205898>

Yes, also from our own experience we can say that we virtually never succeed in exactly reproducing a computational study. Didn't want to offend to computational scientists but now changed this in the abstract.

13. "preprint repositories (e.g., bioRxiv, medRxiv)"

Would add arXiv to this, as it's the original (established 1991)

Added to introduction (**LINE 94**)

14. "Over the last decade, FAIR data management has been introduced to ensure availability"

Define FAIR acronym

Added to introduction (**LINE 129**)

15. Would be useful to discuss how controlled ontologies may be helpful as well, for both biological concepts (MONDO, Human Phenotype Ontology, Uberon) and technical steps (antibodies, protocols).

In the last part of the discussion, we already mentioned a few ontologies and other standards from which ENCORE could benefit (**LINE 1014**). In the Introduction we mentioned (**LINE 184**) “ENCORE documentation of computational protocols provides information for annotating the (pre)processed data to meet the FAIR guiding principles”. This ‘documentation’ could be standardized using ontologies/vocabularies. Within an ENCORE project this could also improve machine readability depending on its implementation.

However, we (currently) consider the use of ontologies beyond the scope of ENCORE. We mentioned in the Discussion we will need to consider which standards (including ontologies) are really relevant, if any, for ENCORE and reproducibility. The main problem is that many are domain dependent (In **LINE 1021** in the Discussion we now make a remark about this; **R1 (15)**) but the main problem, in our view, is that it requires easy-to-use software tools to actually use these ontologies in the context of ENCORE without increasing the overhead for the Project Team without any real benefit.

We added to the manuscript (Discussion, **LINE 1023**): “However, the main challenge we see is the development of software tools to support and use ontologies and standards in the context of ENCORE without introducing much overhead while providing clear benefit”

16. Machine readability of results is also important. So much data is locked away in PDFs, or unparsable Excel sheets.

We agree although from a reproducibility perspective perhaps not that important. Currently, we follow the guidelines from Sandve (2013). In the README.md file in the \Result directory one finds a reflection of part of these guidelines as instructions to the ENCORE user:

- store intermediate results
- store tables as tab delimited files
- store the data on which a figure is based
- provide a stand-alone piece of code to reproduce the figure from this data.
- document the important figures and tables.
- document the interpretation of the important figures/tables.

This should ensure that the (main) results of computational analyses are machine readable and clearly documented.

Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten simple rules for reproducible computational research. PLoS Comput Biol, 9(10), e1003285.

We added a brief section about machine readability to the Discussion (**LINE 1027**) although from a different perspective.

17. “Perhaps the most significant challenge to overcome for routine usage of initiatives such as ENCORE is the lack of incentives to sufficiently motivate researchers to spend sufficient time and effort on reproducibility.”

Indeed, would like to hear some ideas on how to improve this situation.

We wish we had good ideas on this other than the few general points mentioned in the Discussion (to which one suggestion of Reviewer 1 has been added: ““scientific reproducibility” should be used as a litmus test for deciding correction versus retraction.” (LINE 815)

Moseley, H. (2024). In the AI science boom, beware: your results are only as good as your data. Nature. <https://doi.org/10.1038/d41586-024-00306-2>

We have also added a reference to the UNESCO working group on open science funding and incentives. <https://unesdoc.unesco.org/ark:/48223/pf0000383806> and refer to the recently initiated Dutch OpenScience.nl initiative (LINE 797) that will also have a focus on this question.

18. GitHub Releases lets users store intermediate sized data in a version-controlled way. This Releases feature can be combined with Zenodo and allow you to assign DOIs to GitHub repos. <https://docs.github.com/en/repositories/archiving-a-github-repository/referencing-and-citing-content>

We were not aware of this. We have added this to (LINE 128). Thanks for the tip.

19. “In addition, the size of the data and results may be too large to host on GitHub.”
GitHub Large File Storage is an option, but personally I always found this a bit tricky to work with. <https://docs.github.com/en/repositories/working-with-files/managing-large-files/configuring-git-large-file-storage>

We also looked into this but since GitHub is not the hub for a project in the context of ENCORE, we did not use it. See also our comments **point (1) of the in the ‘Summary’** of this reviewer, and **Supplementary Method 4**, Section 4.3.1.

20. “This approach is documented in detail in the ENCORE Step-by-Step guide (Supplementary Method 1) and takes less than 30 minutes to complete”
That’s good to highlight, time investment is the number one thing stopping many people from pursuing good reproducible practices. Recording user-friendly youtube videos at differing lengths and levels of detail could also help, as it offers users an even lower bar of entry to get them started. It also gives you a chance to give them your hook for reproducible practices if they didn’t take the time to read your paper or documentation.

Indeed, only few people (carefully) read manuals/instructions nowadays. The recording of a few videos explaining ENCORE and how to use it, is on our to do list. A first video (but not a tutorial) was made about one month ago to explain the core facilities (eg NGS, metabolomics, imaging) about ENCORE: https://www.youtube.com/watch?v=VVVPO_M0hvg

21. I think it will be quite important to get feedback from other labs that aren’t directly connected with yours. This will help to reveal any pitfalls of ENCORE, and determine what the needs are of other groups. This will be especially important when trying to understand what would motivate people most to use ENCORE, and what might drive them away.

We consider this as very important. We have been talking about this with colleagues from other groups. Below, I repeat my answer for Reviewer 2 who made a similar remark. We added a remark to the Discussion (LINE 726).

So far, we did not involve other research groups, because we were still gaining experience ourselves resulting in changes to the setup. Within our group, we organized several evaluations (Figure 6) and, currently, we are taking initial steps to evaluate (and further develop) ENCORE in

collaboration with other research groups. In this context, three project proposals related to ENCORE have been submitted or will be submitted soon:

- A project proposal for a Dutch Open Science call (NWO; <https://www.nwo.nl/en/researchprogrammes/open-science/open-science-fund>) to obtain funding to take this next step.
- An MSCA EU proposal (<https://marie-sklodowska-curie-actions.ec.europa.eu/actions/doctoral-networks>) involving a larger consortium in which ENCORE is part of the advocated Open Science approach;
- An NWO Dutch Research Infrastructure proposal (<https://www.nwo.nl/en/researchprogrammes/research-infrastructure-ri-national-consortia>)

22. “In September 2022, we evaluated ENCORE 3.0 to test if ENCORE had indeed improved the reproducibility of our projects.”

This is a great start, and I commend you taking the time to take on this level of rigour. That said, I think an even better way to evaluate this would be to have totally independent labs run this evaluation and see if they can reproduce the results, as they won’t have the benefit of being familiar with these projects peripherally (by virtue of being in the same lab). Of course, you’ll need to find someone willing to do this, but it could be a valuable experience for both parties.

It is our intention to write a few manuscripts about specific projects for which we used ENCORE and our experience. The first manuscript is being prepared. For this manuscript, which is about the use of ENCORE for a project about the simulation and analyses of spatial transcriptomics data, we will also include the experience in reproduction from a colleague from another research group who was not involved in this project nor in ENCORE.

23. “One specific issue was that the lab journal often did not contain an adequate summary of (supervisory) meetings and email exchanges”

Zoom now has automated note taking that summarises the meeting. Similar integrations exist for Teams, etc. This could be one way to improve this. If you want to take this even further, you could create a hook that automatically collects these notes and pushes them to GitHub, preferably to the relevant project.

We briefly discussed this within our group some time ago, but so far couldn’t find the time to put this to the test. But such (AI) tools will increasingly help to document projects. We made a remark about this in the Discussion (**LINE 733 and 1049**). Currently, our institute doesn’t offer such tools and installation of additional software with Teams is not allowed. Zoom is not allowed either. In addition, the use of AI tools like chatGPT and copilot is under discussion for confidential/patient related projects since it is unknown what the companies behind these tools do with such data.

24. “Finally, it became clear that project organization and documentation often had lower priority than doing the actual research. For these reasons, most projects did not fully adhere to the ENCORE guidelines.”

Not ideal but I appreciate the honesty and can commiserate with the difficulty of convincing others to engage in these practices. It’s the classic carrot vs. stick dilemma. I find the carrot tends to work better as first line approach, by making these practices as easy and immediately beneficial to users as possible, with even greater benefits in the long run.

We agree.

25. “Most of our ENCORE projects reside in the cloud”

Could you clarify what you mean by cloud here? Do you mean GitHub, some cloud service like AWS/Google, or an institutional cloud service?

We clarified in the Result section (**LINE 289**).

Reviewer #1 (Remarks to the Author):

The authors have adequately addressed the main issues raised by this reviewer.

Reviewer #2 (Remarks to the Author):

The authors have clearly put a lot of thought and effort into addressing in a very comprehensive manner the rather diverse comments of all three reviewers. I would like to thank the authors for the constructive and helpful responses (and changes) to my own comments on their manuscript, and in particular their careful explanations of their motivations and goals in developing ENCORE. I found their rebuttals of some of my points very convincing, and I think the changes to the manuscript make the goals and contributions of the work much clearer for the reader. The changes to the technical descriptions have also much improved the revised manuscript. I am now very pleased to recommend acceptance of the revised manuscript.

Reviewer #3 (Remarks to the Author):

See attached comments.

Reviewer #3 Attachment on the following page

Summary

The manuscript has been much improved since the initial submission. Many of the suggestions by the reviewers have been addressed through implementation or convincing rebuttal. While I think there is still much room for improvement and I have concerns about the reality of uptake by the broader research community, these limitations are thoroughly discussed in the revised manuscript. Even for those that do not adopt ENCORE, I think that this manuscript is useful for highlighting the challenges associated with trying to achieve reproducibility and demonstrates some concrete avenues towards addressing them.

Major Comments

- Please change the “0_README.md” files to “README.md” in all subfolders in addition to the main directory. GitHub automatically displays README.md files in all locations which makes navigating directories far less tedious. Realistically, many users will want to navigate the project on the GitHub site rather than utilising the included Navigator.
- I appreciate the in-depth assessment of GitHub tools. While I have some minor disagreements with some of the conclusions, I think many of the concerns are valid.
- To ensure users can utilise the Python-based Navigator, provide a minimal conda environment yaml and/or requirements.txt file, along with instructions to install it. This of course will be optional for users. *markdown* is the only non-standard Python module (<https://docs.python.org/3/library/index.html>) so this should be pretty easy. Just specify the minimum Python version as well.
- Line 614: “We are in the process of improving the Navigator.”
 - o Please elaborate a bit.
- The authors mention that json/yaml offer machine readability at the detriment of human readability. This is only partly true, since these formats can be rendered by front-end GUI applications as text fields for users to enter (and only be shown in raw format on the backend). Therefore, I think a future avenue that should be mentioned is further developing interactive tools for users to set up and populate their projects, while still retaining machine readability (and thus portability to other applications).
- I think it would make sense to enable the Template feature for the main ENCORE GitHub repo (<https://github.com/EDS-Bioinformatics-Laboratory/ENCORE>). See instructions here: <https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository>
- I don't think the Manuscript subfolder belongs in the ENCORE template or ENCORE_AUTOMATION repos. While it's great that the manuscript is made available, I'd recommend moving it to its own dedicated repo so it's not copied every time someone forks/clones the template:
 - o https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/tree/main/Manuscript/ENCORE_v22_MetaArXiv
 - o https://github.com/EDS-Bioinformatics-Laboratory/ENCORE_AUTOMATION/tree/main/Manuscript/ENCORE_v22_MetaArXiv

Minor Comments

- Just an idea, GitHub provides API access to repo Issues, Projects, Discussions. Scripts that read/write machine-readable files in ENCORE could transfer notes into GitHub tools (and vice-versa). Given that GitHub is currently far more popular than other repository platforms (eg GitLab) I think these sorts of integrations would be a worthwhile investment. Something to consider for future versions.
- Line 600: "ENCORE perfectly matches with these suggestions and contributes to building the researcher's scientific track record."
 - o Change to "ENCORE aligns with...". Currently overstates the advantages of ENCORE.
- Nice work on the Wiki. I think this is a helpful resource for users to easily access info about how to use ENCORE.

RESPONSE TO THE REVIEWER's COMMENTS

Summary

The manuscript has been much improved since the initial submission. Many of the suggestions by the reviewers have been addressed through implementation or convincing rebuttal. While I think there is still much room for improvement and I have concerns about the reality of uptake by the broader research community, these limitations are thoroughly discussed in the revised manuscript. Even for those that do not adopt ENCORE, I think that this manuscript is useful for highlighting the challenges associated with trying to achieve reproducibility and demonstrates some concrete avenues towards addressing them.

Major Comments

- Please change the "0_README.md" files to "README.md" in all subfolders in addition to the main directory. GitHub automatically displays README.md files in all locations which makes navigating directories far less tedious. Realistically, many users will want to navigate the project on the GitHub site rather than utilising the included Navigator.

As explained in the rebuttal, we only changed 00_README-FIRST.md to README.md in the top-level directory of the sFSS since this is the default GitHub README file of the ENCORE GitHub repository. The readme file in \Processing was already named README.md since this file serves as the default readme file for ENCORE project (i.e., only the \Processing directory of ENCORE-based projects are sync'ed with a project repository).

However, we will not change names of the other 0_README.md files for several reasons that were explained in the (revised) manuscript

1. The GitHub repository does not contain a complete project. It only contains versions of the code and code documentation in the \Processing directory. Thus, not all 0_README.md files that are found in the sFSS end up in a project repository. Also the sFSS top-level README.md file will not end-up in a project GitHub repository.
2. GitHub is not the main point of entry for a project (since it only contains part of the project). Instead, the full sFSS is the main point of entry (where ever it may reside on a local hard disk, Zenodo, Cloud, etc). Therefore, from this point of view there is no reason to rename to README.md since the project is not browsed/used from GitHub.
3. The reason for using 0_README.md instead of README.md is to ensure that these files are at the top of the file list when browsed on e.g., a laptop/PC. This makes it easy to locate these files if the directories fill up with other files during a project. This was stated in Supplementary-Methods-2.docx (page 6).

4. Thus, it also was not to facilitate the Navigator.

We made a small correction in Supplementary-Methods-2.docx: we removed 00_README.md since it does no longer exist in ENCORE.

- I appreciate the in-depth assessment of GitHub tools. While I have some minor disagreements with some of the conclusions, I think many of the concerns are valid.
- To ensure users can utilise the Python-based Navigator, provide a minimal conda environment yaml and/or requirements.txt file, along with instructions to install it. This of course will be optional for users. *markdown* is the only non- standard Python module (<https://docs.python.org/3/library/index.html>) so this should be pretty easy. Just specify the minimum Python version as well.
 - o We now provide instructions in https://github.com/EDS-Bioinformatics-Laboratory/FSS-Navigator/tree/main/0_SoftwareEnvironment
 - o These instructions are referred to from the
 - Navigator README.md file <https://github.com/EDS-Bioinformatics-Laboratory/FSS-Navigator/>
 - ENCORE README.md file <https://github.com/EDS-Bioinformatics-Laboratory/ENCORE>
 - Step-by-Step Guide found in <https://github.com/EDS-Bioinformatics-Laboratory/ENCORE>
 - ENCORE Wiki: <https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/wiki/sFSS-Navigator>
- Line 614: “We are in the process of improving the Navigator.”
 - o Please elaborate a bit.

We added a few lines to the Discussion sections (LINE 607).
- The authors mention that json/yaml offer machine readability at the detriment of human readability. This is only partly true, since these formats can be rendered by front-end GUI applications as text fields for users to enter (and only be shown in raw format on the backend). Therefore, I think a future avenue that should be mentioned is further developing interactive tools for users to set up and populate their projects, while still retaining machine readability (and thus portability to other applications).

Good suggestion. We have added this to the section Machine readability (LINE 770).
- I think it would make sense to enable the Template feature for the main ENCORE GitHub repo (<https://github.com/EDS-Bioinformatics->

[Laboratory/ENCORE](#)). See instructions here:
<https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository>
We did not follow up on this suggestion.

It is not clear what this would actually achieve since the ENCORE repository is already a template in itself. GitHub templates are useful for creating new repositories, but in our case, we do not want to create a repository from the full ENCORE template since for new projects only part of the \Processing directory will end up in a project repository. Moreover, in the Step-by-Step Guide we explain in detail how to initialize a new ENCORE project, and we now have made scripts available on https://github.com/EDS-Bioinformatics-Laboratory/ENCORE_AUTOMATION to do this automatically (as described in the revised manuscripts).

- I don't think the Manuscript subfolder belongs in the ENCORE template or ENCORE_AUTOMATION repos. While it's great that the manuscript is made available, I'd recommend moving it to its own dedicated repo so it's not copied every time someone forks/clones the template:
 - o https://github.com/EDS-Bioinformatics-Laboratory/ENCORE/tree/main/Manuscript/ENCORE_v22_MetaArXiv
 - o https://github.com/EDS-Bioinformatics-Laboratory/ENCORE_AUTOMATION/tree/main/Manuscript/ENCORE_v22_MetaArXiv

We agree.

We made an additional GitHub repository (<https://github.com/EDS-Bioinformatics-Laboratory/ENCORE-SUPPLEMENT>) that now contains supplementary information for ENCORE, including the pre-print manuscript.

Minor Comments

- Just an idea, GitHub provides API access to repo Issues, Projects, Discussions. Scripts that read/write machine-readable files in ENCORE could transfer notes into GitHub tools (and vice-versa). Given that GitHub is currently far more popular than other repository platforms (eg GitLab) I think these sorts of integrations would be a worthwhile investment. Something to consider for future versions.

Yes. We are aware of the API's but never looked into these. Could indeed provide some benefits.

- Line 600: "ENCORE perfectly matches with these suggestions and contributes to building the researcher's scientific track record."
 - o Change to "ENCORE aligns with...". Currently overstates the advantages of ENCORE.

Changed.

- Nice work on the Wiki. I think this is a helpful resource for users to easily access info about how to use ENCORE.

Thanks, will be further improved and extended.