Response to reviewer comments (The five pillars)

Mark Ziemann, Pierre Poulain, Anusuiya Bora

2023-08-31

Dear Editor,

Many thanks for your consideration of our manuscript. Below is our point-by-point response to reviewer comments.

## Reviewer 1

In this manuscript, the authors present a framework that computational researchers should follow to maximize the reproducibility of their computational research. The five pillars that they identified are 1) code version control and sharing, 2) compute environment control, 3) literate programming, 4) documentation and 5) FAIR data sharing. They give a fair amount of detail regarding each pillar. This paper contains no original research and is more of an opinion piece. With that in mind, if BiB is interested in this topic, I recommend publishing. The authors make very reasonable suggestions. My only request for edits is to rearrange the section structure of the manuscript to be in line with the numbering of the pillars. Right now it is quite confusing.

RESPONSE: We thank reviewer 1 for their positive feedback. With regards to the ordering of the five pillars, we have amended Figure 1 so that the pillars now appear in the same order as they are detailed in sections following.

## Reviewer 2

Comments to the Author: The article offers a comprehensive perspective on reproducibility in computational research, delivering clear and logically backed recommendations. It is indeed a very interesting topic that needs our attention. Nevertheless, there are areas that could benefit from further development and clarity:

RESPONSE: We appreciate reviewer 2’s positive feedback, insights and thoughtful suggestions.

### R2 Point 1

1. Inclusion of Case Studies or Examples: To strengthen your argument, consider incorporating case studies or practical examples that demonstrate the effectiveness of your recommendations. Highlighting specific instances where these best practices have enhanced reproducibility or solved particular issues could substantiate your points more effectively.

RESPONSE: To demonstrate the importance of computational reproducibility, we elaborate on the “forensic bioinformatics” case study of Baggerly and Coombes (2010) [1] critique of Potti et al (2006) [2] in the introduction:

For example, in 2006 an article entitled “Genomic signatures to guide the use of chemotherapeutics” generated a great deal of interest as it was an early application of high throughput transcriptomics in the prediction of individual patient responses to different chemotherapies [2]. After observing some unusual features of the patient group, Baggerly and Coombes attempted reproduction of some of the key findings [1]. Without the help of scripted workflows to guide the re-analysis, the team used forensic bioinformatic techniques to piece together how the dataset was originally analyzed. Their investigations found a litany of issues. Firstly, the labeling of patients in the test set as “responders” or “non-responders” had been reversed in the original analysis. Secondly, some of the patients were included more than once (some up to four times) in the analysis, likely to cause major distortions in results. Confusingly, some of the reused data has inconsistent grouping; ie: some of the samples are labeled both sensitive and resistant. Additional errors include two cases where results (charts) were ascribed to the wrong drug. Baggerly and Coombes highlight that such mistakes can inadvertently occur when conducting unscripted data analysis such as using spreadsheets, and such problems can be obscured by a lack of documentation. The article underwent two corrigenda, but was ultimately retracted in 2011, as the authors were not able to reproduce the findings themselves due to “corruption of several validation data sets” [3]. As the array findings were the basis for clinical trials where patients were allocated to treatments, the flawed data analysis may have harmed patients given the wrong drug in the period 2007-2010. In 2010, Duke University terminated the trials and suspended the lead author, Dr Anil Potti, who later resigned. Duke was served eight lawsuits by families of affected patients seeking compensation for exposure to harmful and unnecessary chemotherapy, which were settled out of court [4]. This worst-case scenario emphasizes that computational reproducibility is crucial for translating bioinformatics research into real-world outcomes.

### R2 Point 2

1. Expanding on Current Shortcomings: The article mentions that current practices are not meeting the goal of reproducibility. An expanded discussion on the ramifications of these shortcomings on scientific progress could accentuate the urgency and significance of addressing this issue.

RESPONSE: In the introduction we have spoken to the consequences of poor computational research, and emphasize the urgency of this issue (which complements the case study summary above):

The ramifications of irreproducible and unreliable research include misleading the community, wasting research funds, slowing scientific progress, eroding public confidence in science and tarnishing the reputation of associated institutions and colleagues. In certain cases, irreproducible bioinformatics has the potential to place patient safety at risk.

### R2 Point 3

1. Focus on Implementation Challenges: While the article acknowledges the importance of certain practices, an increased focus on potential obstacles researchers may face when trying to implement these suggestions, as well as strategies to overcome these challenges, would be insightful.

RESPONSE: We have re-written the Challenges section to highlight some of the barriers slowing the adoption of best practices, with reference to previous contributions on the topic. These are structured around what different stakeholders can do to improve the situation (including funders, journals, institutions and individual researchers). We specifically note the effect of incentives on motivation, and the role of automation for routine reproducibility checking.

### R2 Point 4

1. Clear Structure: Consider organizing the article into clearly defined sections such as ‘Introduction,’ ‘Background,’ ‘Recommendations,’ ‘Challenges,’ and ‘Conclusion.’ This could significantly enhance readability and comprehensibility.

RESPONSE: The subheading structure has been amended accordingly.

* Abstract
* Introduction
* Recommendations
  + End-to-end automated process
  + Literate programming
  + Code version control and persistent sharing
  + Compute environment control
  + FAIR and persistent data sharing
  + Documentation
  + Continuous validation
* Challenges
* Conclusion
* Key points

### R2 Point 5

1. Engaging Conclusion: Reworking the conclusion to be more impactful could be beneficial. Instead of merely summarizing the recommendations, it would be useful to indicate the potential trajectory of computational research if these recommendations are adopted, as well as highlighting the potential risks if they aren’t.

Response: We have completely rewritten the conclusion. Instead of reiterating the five pillars, we provide the following:

As the chemotherapeutics case study highlights, there are significant risks to poorly conducted bioinformatics research, and current standards are falling short. If adopted widely, the transparency and reproducibility measures proposed here will reduce the chance of such disasters from happening again. Researchers, their institutions, publishers and funders each have a major role to play in fighting the reproducibility crisis by encouraging highly reproducible research practices.

## Reviewer 3

### R3 Point 1

The paper is mostly well written and addresses an important topic: computational reproducibility and tools to achieve it. The authors describe five “pillars” under which various tools and techniques can be categorized. The paper is pretty comprehensive in what it covers and has useful supplementary material. My main issue with the paper is that it reiterates many things that have been covered in other papers. The authors cite 11 such papers. Many of the topics covered in submitted paper have already been covered well in other papers. I am most familiar with reference 13, which covers many of the same topics, although this paper goes into more detail on many issues. The authors could do more to differentiate their paper from previous ones and perhaps remove some topics that are already covered well elsewhere.

RESPONSE: We thank reviewer 3 for their careful assessment of our article and appreciate the suggestions which have improved the manuscript. Indeed, this article does reiterate a great deal of what has already been said, it is a review after all. We note that each of those recommendation articles mentioned have different emphases. For example, some of them don’t speak to the benefits of literate programming, while others do not mention the importance of documentation. Some are focused on specific concepts like infrastructure, containerization or continuous analysis. Some are proof of principle workflows, which present a rigid combination of tools. Here, we describe the ways computational researchers can make their own work more reproducible. We are assuming that the readers are not advanced analysts, so some topics like pipelining systems are covered in less detail and we refer the reader to those relevant reviews. What does make this article unique is the five pillars infographic, and specific combination of recommendations that incorporates new material (eg: Software Heritage, Guix, Continuous Analysis). This is useful, because bioinformatics analysts can use the infographic to benchmark their projects and identify areas where changes are needed. Our article also provides a handy list of recommended tutorials as a supplementary resource.

Aside from that, I have listed below some relatively minor issues that, if addressed, would improve the paper. When I list page numbers, I am using the PDF page numbers rather than the numbers shown in the top-left corner of the manuscript.

### R3 Point 2

1. Page 3, Line 36: “bioinformatics data analysts (not tool developers).” The authors state these individuals as the primary audience, but some parts of the paper seem to be targeted at a more technical audience. Or maybe I am misunderstanding the intent. If the audience is “bioinformatics data analysts,” that would imply people who are bioinformaticians but are analyzing data rather than creating tools. However, a much bigger audience (and perhaps more important audience) are non-bioinformaticians who analyze data.

RESPONSE: The passage has been amended for clarity:

The intended audience is anyone who analyzes biological data.

### R3 Point 3

1. Page 3, Line 43: “enshrined by code” (this language is awkward)

RESPONSE: The passage has been amended:

… it needs to be formalized in code and …

### R3 Point 4

1. Page 3, Lines 45-47: What about tasks that cannot be automated? I see that this topic is addressed later. But this part implies that everything can be automated.

RESPONSE: The section has been amended to acknowledge that not all steps can be automated.

To ensure reproducibility of bioinformatics workflows, they need to be formalized in code wherever possible, from inspecting the raw data to generating the outputs that form the conclusions of the study.

### R3 Point 5

1. Page 3, Line 52: It says that spreadsheets are “overused and misused.” This is subjective and not backed by evidence, other than the well-known examples of gene symbols being formatted as dates.

RESPONSE: The passage in question has been replaced, with reference to evidence from a large survey that found that ~69% of university academics used MS Excel in their research activities.

Spreadsheet errors could be widespread, given that it is used as an analysis tool by ~69% of researchers according to a survey undertaken in 2015-16 of 20,000 university academics [5].

### R3 Point 6

1. Page 3, Line 56: It is not necessarily true that analyses performed using web tools are not reproducible. Although rare, some web tools facilitate reproducibility by providing code or configuration files and/or allowing the apps to be executed locally.

RESPONSE: Thank you for raising this point. We have amended the section in question:

While webtools are valuable for data exploration, there are worries that they undermine reproducibility for the sake of convenience [6]. Transferring data between compute platforms is also discouraged. For example, having workflows that involve combinations of web-based and scripted tools require data transfer steps which are inefficient and error-prone. On the other hand, some web based tools excel at reproducibility. The web-based analysis platforms Galaxy and GenePattern enable sophisticated point-and-click bioinformatics analysis in the browser, and those workflows can also be shared in a reproducible way [7,8]. Some webtools facilitate reproducibility by providing code (e.g.: Degust [9]) or by allowing apps to be executed locally (e.g.: ShinyGO [10]).

### R3 Point 7

1. Page 5, Line 15, type-o = “authors provided along the”

RESPONSE: The passage in question has been amended:

In this work, authors provided alongside the traditional research paper a Jupyter notebook …

### R3 Point 8

1. Page 6, Line 21: “quantum leap” (this term is overly optimistic in this context)

RESPONSE: The passage in question has been amended:

These features make literate programming a useful tool for science communication in a range of situations.

### R3 Point 9

1. How do you get from a notebook to an actual paper submission if you have to do custom formatting of the document, including references? My understanding is that this is still not possible, but please correct me if I’m wrong.

RESPONSE: It is possible to achieve accurate journal-specific formatting of references with a Rmarkdown document. It requires a few things:

* A .bib file that contains the bibliography in bibtex format.
* A .csl file that specifies the citation style. These are available from [GitHub](https://github.com/citation-style-language/styles).
* Modification to the Rmd header section that specifies the path of the bib and csl files.

This enables convenient reformatting of the bibliography for different venues. In terms of overall document formatting, the rticles package provides a suite of custom R Markdown LaTeX formats suitable for many journals.

We have amended the Literate Programming section to mention these features briefly:

Quarto and R Markdown have the ability to automatically generate a bibliography in many different journal styles [11]. Further, dozens of journal-style Markdown document templates are freely available, and integrate with MyST notebook system or the rticles R package [12].

### R3 Point 10

1. What about when your data files are too large to fit on a personal computer?

RESPONSE: When data files are too large for a PC, it is common to use high performance computers, shared clusters or cloud computers. These computers are typically accessed via SSH (headless mode), so Quarto and Rstudio graphical interfaces might not be available. In those cases, the commands quarto render document1.qmd and rmarkdown::render("document2.Rmd") will work for Quarto and R Markdown respectively.

### R3 Point 11

1. What about computationally intensive tasks that must be performed using specialized computing environments like the cloud or clusters?

RESPONSE: We have amended the relevant section to suggest a specific workflow manager for compute intensive applications.

Projects involving computationally intensive tasks would benefit from a build/workflow automation solution. Guides for selecting and using such systems have been already reviewed by others [13,14]. But it is worth mentioning some of the most used solutions in bioinformatics, which include snakemake [15,16], targets [17], CWL [18], WDL [19] and nextflow [20]. The advantage of such tools is that if the analysis were to terminate due to an issue mid-way, for example, a hardware problem at step 8 of a 15-step workflow, the analysis of steps 1-7 wouldn’t need to be repeated. After fixing the issue, re-running the analysis would pick up at step 8 again, which saves labor and compute time.

### R3 Point 12

1. Page 6, line 32: The master script idea was already mentioned earlier.

RESPONSE: True. We have amended the “Documentation” section to remove the unnecessary reiteration:

As outlined above, one of the goals is to reduce the complexity of reproduction, in particular by minimizing the number of commands required.

### R3 Point 13

1. Section on version control: This section focuses mostly on using VC for software development (that is my interpretation). To be consistent with the introduction, it should focus more on data analyses. Although I use VC for analyses, I feel that simpler approaches are better in many cases. For example Dropbox and Google Drive provide some version-control and backup functionality and do not require the same level of knowledge as git.

RESPONSE: In accordance with this suggestion, we have rephrased some of the text to highlight the utility of DVCS for data analysts.

Although DVCSs come from the world of software engineering, data analysts can significantly benefit from their use in the life sciences [21–23].

The bullet points following this passage outline the benefits of using DVCS. While cloud storage like dropbox could protect against code loss, the other five points cannot be satisfied.

### R3 Point 14

1. Page 7: Many data analysts will not know what JupyterLab or VS Code are. References are also needed.

RESPONSE: The phrase has been updated accordingly:

git is typically used at the command line, however it is also incorporated into integrated development environments commonly used in bioinformatics including Rstudio, JupyterLab [24] and VS Code [25].

### R3 Point 15

1. Page 9, line 7: References are missing for these other tools.

RESPONSE: Citations have been added for Podman and Apptainer/Singularity. Rkt was removed, as it appears that development of the project has ceased.

### R3 Point 16

1. Page 9, lines 7-8: I believe you, but I am not aware of evidence that supports this claim.

RESPONSE: The claim made in the manuscript is that “Docker remains the most widely used containerization system for web development and research.” We do not have exact usage data on this, however we can look at other metrics like GitHub stars.

Docker: 72.2k

Podman: 18.9k

Singularity: 3.1k

(25th August).

Based on this we have altered the text:

While there are several alternatives such as Podman [26] or Apptainer/Singularity [27], Docker remains the most widely used containerization system (according to GitHub stars).

### R3 Point 17

1. The Biocontainers project should be mentioned.

RESPONSE: Biocontainers is mentioned in the last paragraph of the “Compute environment control” section.

### R3 Point 18

1. Page 10, lines 51-52: I disagree that the risk is small. There are many instances of using genomic data to identify individuals who have committed crimes.

RESPONSE: In light of this, we have changed the section in question:

While deidentification of data has been widely used to safeguard participant privacy, this may not be suitable for genomics data due to the availability of existing public genetic data sets that heighten reidentification risks [28]. For example, the surnames of some male participants can be inferred based on the Y chromosome variants seen in deidentified genomic data together with public geneology web sites [29]. To foster the responsible reuse of sensitive genomic and health data, the Global Alliance for Genomics and Health (GA4GH) initiative has proposed strategies, technical standards and policy frameworks designed to protect personal data in a way that preserves reproducibility [30,31].

### R3 Point 19

1. Figures 2 and 4 are very similar to figures used in reference 13.

RESPONSE: Figure 4 is adapted from the diagram on the Docker official website, which was originally published in 2015 or earlier (<https://web.archive.org/web/20151023003338/https://www.docker.com/what-docker>). We have cited the original source.

Figure 2 is somewhat different from reference 13, as it shows an Rstudio window with windows for raw code in text format beside the output document. The figure in reference 13 shows a Jupyter notebook window, which has a different arrangement of code and outputs.

We do not believe any changes to Figure 2 or 4 are necessary.

### R3 Point 20

1. Figure 5: I don’t think it’s really necessary to resummarize the FAIR principles. People can go to the source article for that.

RESPONSE: We acknowledge that it can sometimes be tiresome to reiterate things like the FAIR principles. However summarizing the key points succinctly will be in the interests of readers with limited time. We note that the original description of the FAIR principles by Wilkinson and colleagues is ~5,000 words, while this entire manuscript is ~7,000. We note that other reviewers did not insist on removing this section.

### R3 Point 21

1. Page 11, line 13: I disagree that repositories like GEO and SRA are FAIR. There are lots of problems with FAIRness in these repositories.

RESPONSE: While there may be problems with FAIRness with NCBI GEO and SRA, these repositories mostly satisfy FAIR principles (eg: [32]). From the perspective of a researcher sharing genomic data, they are “good enough” as compared to the well known alternatives (eg: general purpose repository, institutional data storage, supplementary file, dropbox, on request, etc.).

### R3 Point 22

1. Page 11, lines 13-15: That’s not true for some of these disciplines. Ecology has NEON, evo bio has NCBI.

RESPONSE: We acknowledge that NCBI is a well known repository for molecular evolution. We have altered the section for clarity:

The [re3data.org](https://www.re3data.org/) registry of data repositories may be useful to find FAIR repositories that accepts data from other domains of study like ecology, physiology, molecular simulation, social sciences and computing [33].

### R3 Point 23

1. Page 11, line 34: Need evidence to back this up.

RESPONSE: Reviewer 3 commented on the passage saying supplementary data files are less findable and accessible as compared to FAIR data repositories. We note that Roche and colleagues previously stated that [34]:

There are no standards for organizing supplementary data both within and across journals and such data are often not readily discoverable or openly accessible (to those without a relevant journal subscription, for example).

They also cite an article by Caetano and Aisenberg that speaks directly to findability and accessibility [35]:

Many journals provide the option for authors to include supplementary data that are made available in the digital version of the publication. However, repositories increase data discoverability, are more reliable for long-term storage and make it possible to make data sets available under open access even when authors transfer copyright ownership to publishers.

In light of this argument, we have added citations to Caetano and Aisenberg in the updated text.

### R3 Point 24

1. Page 11, line 39: It is not necessarily true that CSV files are better than Excel. Excel can retain information about data types, for example, whereas CSVs do not. It depends on what you are trying to accomplish.

RESPONSE: It is true that in some cases, Excel files are more convenient to view. However, in the context of large data sets with thousands of rows and their analytical reproducibility, Excel is a non-starter. For example, it is well known that the behaviour of Excel changes between versions. Earlier versions have limitations around the number of allowed rows and columns. Small changes to Excel files such as merging cells can break machine readibility. Excel users often use cell highlighting or text colour to encode some information which isn’t easily machine readable. The simplicity of TSV and CSV formats means they are readily and rapidly ingested by commonly used programming languages without needing to install extra packages. In line with this and suggestions from others, the passage been amended:

1. Use file formats that are machine-readable and compatible with many different types of software. Some examples include comma- and tab-separated values (CSV/TSV) formats, eXtensible Markup Language (XML), JavaScript Object Notation (JSON), Hierarchical Data Format version 5 (HDF5) and Apache Parquet.

### R3 Point 25

1. One thing that could be added is something about Common Workflow Language. It’s a community-supported specification for accomplishing many of the objectives described here. There are some recent papers about this.

RESPONSE: Thanks for raising this. We have added CWL to the section mentioning pipeline solutions, and referred readers to dedicated reviews on the topic.

Projects involving computationally intensive tasks would benefit from a build/workflow automation solution. Guides for selecting and using such systems have been already reviewed by others [13,14]. But it is worth mentioning some of the most used solutions in bioinformatics, which include snakemake [15,16], targets [17], CWL [18], WDL [19] and nextflow [20].

### R3 Point 26

1. Page 14, lines 9-10: The big question is how to make more progress. We have the tools to achieve reproducibility, but why are we rarely achieving it? The paper mentions incentives and lack of training, which are true. You might consider elaborating a bit. My cynical view is that writing more papers and tutorials will do little without strong incentives and more automation.

RESPONSE: Thanks for raising this, and I note that similar sentiments were shared by reviewer 2 about the challenges that are holding back progress. Please refer to our response to Reviewer 2’s Point #3. The Challenges section now includes brief mentions of incentives and automation.

## Bibliography

1. Baggerly KA, Coombes KR. Deriving chemosensitivity from cell lines: Forensic bioinformatics and reproducible research in high-throughput biology. 2010; 1309–1334.

2. Potti A, Dressman HK, Bild A, Riedel RF, Chan G, Sayer R, et al. Genomic signatures to guide the use of chemotherapeutics. Nat Med. 2006;12: 1294–1300.

3. Potti A, Dressman HK, Bild A, Riedel RF, Chan G, Sayer R, et al. Retraction note: Genomic signatures to guide the use of chemotherapeutics. Nat Med. 2011;17: 135–135.

4. Dyer C. Duke university settles lawsuits alleging that patients were harmed in chemotherapy trials. BMJ. 2015;350: h2559–h2559.

5. Kramer B, Bosman J. Innovations in scholarly communication - global survey on research tool usage. F1000Res. 2016;5: 692.

6. Wijesooriya K, Jadaan SA, Perera KL, Kaur T, Ziemann M. Urgent need for consistent standards in functional enrichment analysis. PLoS Comput Biol. 2022;18: e1009935.

7. The Galaxy Community, Afgan E, Nekrutenko A, Grüning BA, Blankenberg D, Goecks J, et al. The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update. Nucleic Acids Res. 2022;50: W345–W351.

8. Reich M, Liefeld T, Gould J, Lerner J, Tamayo P, Mesirov JP. GenePattern 2.0. Nat Genet. 2006;38: 500–501.

9. Powell D. Drpowell/degust 4.1.1. Zenodo; 2019. doi:[10.5281/zenodo.3501067](https://doi.org/10.5281/zenodo.3501067)

10. Ge SX, Jung D, Yao R. ShinyGO: A graphical gene-set enrichment tool for animals and plants. Bioinformatics. 2020;36: 2628–2629.

11. Citations & footnotes. <https://quarto.org/docs/authoring/footnotes-and-citations.html>;

12. Xie Y, Allaire JJ, Grolemund G. R markdown: The definitive guide. Chapman; Hall/CRC; 2018.

13. Leipzig J. A review of bioinformatic pipeline frameworks. Brief Bioinform. 2016; bbw020.

14. Reiter T, Brooks PT, Irber† L, Joslin† SEK, Reid† CM, Scott† C, et al. Streamlining data-intensive biology with workflow systems. Gigascience. 2021;10.

15. Köster J, Rahmann S. Snakemake–a scalable bioinformatics workflow engine. Bioinformatics. 2012;28: 2520–2522.

16. Mölder F, Jablonski KP, Letcher B, Hall MB, Tomkins-Tinch CH, Sochat V, et al. Sustainable data analysis with snakemake. F1000Res. 2021;10: 33.

17. Landau WM. The targets r package: A dynamic make-like function-oriented pipeline toolkit for reproducibility and high-performance computing. Journal of Open Source Software. 2021;6: 2959. doi:[10.21105/joss.02959](https://doi.org/10.21105/joss.02959)

18. Crusoe MR, Abeln S, Iosup A, Amstutz P, Chilton J, Tijanić N, et al. Methods included: Standardizing computational reuse and portability with the common workflow language. Commun ACM. 2022;65: 54–63.

19. Voss K, Gentry J, Van der Auwera G. Full-stack genomics pipelining with GATK4 + WDL + cromwell. F1000Research; 2017. doi:[10.7490/f1000research.1114634.1](https://doi.org/10.7490/f1000research.1114634.1)

20. Di Tommaso P, Chatzou M, Floden EW, Barja PP, Palumbo E, Notredame C. Nextflow enables reproducible computational workflows. Nat Biotechnol. 2017;35: 316–319.

21. Ram K. Git can facilitate greater reproducibility and increased transparency in science. Source Code for Biology and Medicine. 2013;8: 7. doi:[10.1186/1751-0473-8-7](https://doi.org/10.1186/1751-0473-8-7)

22. Blischak JD, Davenport ER, Wilson G. A Quick Introduction to Version Control with Git and GitHub. PLOS Computational Biology. 2016;12: e1004668. doi:[10.1371/journal.pcbi.1004668](https://doi.org/10.1371/journal.pcbi.1004668)

23. Perez-Riverol Y, Gatto L, Wang R, Sachsenberg T, Uszkoreit J, Leprevost F da V, et al. Ten Simple Rules for Taking Advantage of Git and GitHub. Markel S, editor. PLOS Computational Biology. 2016;12: e1004947. doi:[10.1371/journal.pcbi.1004947](https://doi.org/10.1371/journal.pcbi.1004947)

24. Kluyver T, Ragan-Kelley B, Pérez F, Granger B, Bussonnier M, Frederic J, et al. Jupyter notebooks – a publishing format for reproducible computational workflows. Loizides F, Schmidt B, editors. IOS Press; 2016. pp. 87–90.

25. Vscode: Visual studio code. Available: <https://github.com/microsoft/vscode>

26. Heon M et al, Walsh D, Baude B, Mohnani U, Cui A, Sweeney T, et al. Podman - : A tool for managing OCI containers and pods. Zenodo; 2018.

27. Kurtzer GM, Sochat V, Bauer MW. Singularity: Scientific containers for mobility of compute. PLoS One. 2017;12: e0177459.

28. Heeney C, Hawkins N, Vries J de, Boddington P, Kaye J. Assessing the privacy risks of data sharing in genomics. Public Health Genomics. 2011;14: 17–25.

29. Gymrek M, McGuire AL, Golan D, Halperin E, Erlich Y. Identifying personal genomes by surname inference. Science. 2013;339: 321–324.

30. Rehm HL, Page AJH, Smith L, Adams JB, Alterovitz G, Babb LJ, et al. GA4GH: International policies and standards for data sharing across genomic research and healthcare. Cell Genom. 2021;1: 100029.

31. Thorogood A, Rehm HL, Goodhand P, Page AJH, Joly Y, Baudis M, et al. International federation of genomic medicine databases using GA4GH standards. Cell Genom. 2021;1: 100032.

32. Bonaretti S, Willighagen E. Two real use cases of FAIR maturity indicators in the life sciences. bioRxiv. 2019.

33. Pampel H, Vierkant P, Scholze F, Bertelmann R, Kindling M, Klump J, et al. Making research data repositories visible: The re3data.org registry. PLoS One. 2013;8: e78080.

34. Roche DG, Kruuk LEB, Lanfear R, Binning SA. Public data archiving in ecology and evolution: How well are we doing? PLoS Biol. 2015;13: e1002295.

35. Caetano DS, Aisenberg A. Forgotten treasures: The fate of data in animal behaviour studies. Anim Behav. 2014;98: 1–5.