Meta-Psychology, 2024, vol 8, MP.2023.4021
https://doi.org/10.15626/MP.2023.4021
Article type: Original Article
Published under the CC-BY4.0 license

Open data: Not Applicable
Open materials: Yes
Open code: Not Applicable
Open and reproducible analysis: Not Applicable
Open reviews and editorial process: Yes
Preregistration: No

Edited by: Rickard Carlsson
Reviewed by: Gilad Feldman, Erin M. Buchanan
Analysis reproduced by: Not Applicable
All supplementary files can be accessed at OSF:
https://doi.org/10.17605/OSF.IO/K9EBM

ReproduceMe: Lessons from a pilot project on computational reproducibility

Daniel H. Baker[1], Mareike Berg[1], Kirralise J. Hansford[1],
Bartholomew P.A. Quinn[1], Federico G. Segala[1], and Erin L. Warden-English[1]
[1]Department of Psychology, University of York, York, UK

If a scientific paper is computationally reproducible, the analyses it reports can be repeated independently by others. At the present time most papers are not reproducible. However, the tools to enable computational reproducibility are now widely available, using free and open source software. We conducted a pilot study in which we offered 'reproducibility as a service' within a UK psychology department for a period of 6 months. Our rationale was that most researchers lack either the time or expertise to make their own work reproducible, but might be willing to allow this to be done by an independent team. Ten papers were converted into reproducible format using *R markdown*, such that all analyses were conducted by a single script that could download raw data from online platforms as required, generate figures, and produce a pdf of the final manuscript. For some studies this involved reproducing analyses originally conducted using commercial software. The project was an overall success, with strong support from the contributing authors who saw clear benefit from this work, including greater transparency and openness, and ease of use for the reader. Here we describe our framework for reproducibility, summarise the specific lessons learned during the project, and discuss the future of computational reproducibility. Our view is that computationally reproducible manuscripts embody many of the core principles of open science, and should become the default format for scientific communication.

Keywords: computational reproducibility; markdown; CI/CD; open science

Background

In the early days of scientific inquiry, the results of experiments and observations were recorded on paper, and calculations and analyses were performed by hand. Journal articles aimed to provide a summary of these findings for other scientists, who implicitly trusted that the author(s) had communicated a fair and accurate account of their results. The widespread use of computers in the late 20th century updated many of these practices. However most analyses were still performed on an individual researcher's computer, often using closed commercial software, and code and data were rarely available for scrutiny. In recent decades this has begun to change, and sharing of code and data is now commonplace. Furthermore, the tools and infrastructure are available to produce empirical papers that are fully computationally reproducible: all data analyses are calculated from the raw data, and a formatted version of the final paper (e.g. a pdf), including all figures, tables and statistical results, is automatically generated.

Computational reproducibility has multiple benefits. The analysis pipeline is fully transparent to readers, reviewers, and editors of a paper. This encourages confidence in the analytic process, allows it to be checked and verified (perhaps as part of a formal audit), and should reduce errors and preclude some types of academic misconduct. Sharing of code and data makes analyses future-proof, encourages secondary data analysis, and spreads good practice across a research field. Integrating code and data into a single self-contained repository can also minimise dependencies on local files and directory structures, reducing technical barriers for re-use. Ultimately, reproducibility increases confidence in the scientific method and is a cornerstone of modern open research practice. However there are some barriers to widespread adoption of these approaches. Critically, generating reproducible manuscripts requires substantial technical ability and time, which might not be available for all researchers. In addition, reproducible analyses should ideally use free and open source software (FOSS) that does not require commercial licenses, and can be downloaded by anyone with an internet connection. However many researchers still use commercially licensed analysis software (often with their institution providing the license) that in many cases involve graphical user interfaces that make scripting and automation difficult or impossible. Finally, even for papers that are claimed to be reproducible, this is rarely a 'frictionless' process (Crüwell et al., 2023; Hardwicke et al., 2021; Obels et al., 2020), and often becomes more challenging over time as data and software become unavailable or obsolete.

Aims and objectives

We obtained funding to conduct a pilot project (called the 'ReproduceMe' project) that aimed to make 10 empirical papers in psychology and neuroscience fully computationally reproducible during a 6-month period (February to July 2023). Although we had seen examples of reproducible manuscripts in the past (a paper by Rouder and Haaf (2018) sparked our initial enthusiasm) these were rarely for empirical studies, perhaps owing to the complexity and 'messiness' of real data. Before the pilot, the first author had developed some of these techniques by publishing two papers that were fully reproducible – one largely involving statistics and simulation (Baker, 2021), and the other reporting novel empirical data (Baker et al., 2021). Our key objectives were:

1. To take 10 manuscripts that were in a non-reproducible format, and convert these into fully reproducible documents
2. To include some studies in which the original analyses used commercial software, and reproduce these analyses using FOSS such as *R* and *python*
3. To cover a wide range of study designs and subdisciplines within psychology and human neuroscience
4. To develop training materials on using tools such as markdown to make papers reproducible, as well as producing standardised onboarding and feedback forms
5. To further develop our methods to include features such as automatic downloading of data, preservation of the computational environment, and automated execution on remote platforms

The pilot achieved all of these objectives within the funding period, and within our budget of £10,000. The funds were primarily used to pay for analyst time, at a rate of around £17/hr (588 hours total), as well as for some small items of IT equipment. The training materials are available at https://github.com/bakerdh/ReproduceMe, and example onboarding and feedback forms are given in Appendices B and C.

A framework for computational reproducibility

Our guiding principle when converting papers into a reproducible format was to keep everything as simple as possible for the end user. Wherever practical, we preferred to create a single script that contained all of the manuscript text and code, rather than requiring multiple separate files that need to be run for different parts of an analysis. Our experience (and that of others, see Crüwell et al., 2023; Hardwicke et al., 2021) is that complex code repositories often constitute a barrier to reproducibility, as it is difficult for the end user to work out which scripts need to be run, and in what order. Having a single master document that can load in data files, packages, and other resources as required, greatly simplified the structure of analysis code in most cases. We used *R markdown* (https://rmarkdown.rstudio.com/) for this project, because most members of the project team were familiar with *R*, and we were aware that many of our colleagues use it for data analysis. However alternatives are available for a range of programming environments, including Jupyter notebooks (https://jupyter.org/), Quarto (https://quarto.org/), and MyST (https://mystmd.org/).

In keeping with this principle, we developed procedures to automate package installation and data download, removing this burden from the end user. Here the intention was that only the master markdown script would be required, and this would be able to obtain any additional files and install any software packages it needed to run the analysis. A quirk of package installation in the *R* programming language is that code to install packages (i.e. calls to the *install.packages* function) needs to be run only once on a given system, after which the package is present locally and does not need to be downloaded again. We used some custom code to check which packages were installed already, and only download those that were missing (see example in section 3.13 of the training materials linked in Appendix A). Data files were automatically downloaded from public repositories on the Open Science Framework website (https://osf.io) using the *osfr* (Wolen et al., 2020) package (see also the pip-installable *osfclient* repository for a similar toolbox in *python*; https://github.com/osfclient/osfclient). Again, we wrote code to check if the files were available locally and only download them when they were missing. This allows the code to run robustly, but avoids unnecessary downloads.

Some studies involve large data files or computationally intensive analyses. It might be the case that not all users would wish to (or be able to) run the full analysis from scratch. We therefore found it helpful to have one or more 'flag' variables (usually called *processdata*) at the start of a script to specify the level of analysis required. Exactly how this worked differed across papers, but the general principle was that a flag value of 0 would build the paper using pre-computed results and pre-generated figures. Values above 0 would conduct differing amounts of analysis, with the highest value specifying that the full analysis should be run using the raw data. Analyses involving multiple stages usually saved the results of each stage to intermediate data files, which could also be downloaded from the OSF repository if required. For example, an analysis requiring 8GB of raw data files, and taking over 6 hours to analyse from scratch, can instead be run using the processed data file (<1MB) in a matter of seconds. This also dramatically facilitated our code development, as it meant that complex analyses did not need to be repeated each time we wanted to build and check the final document. We note that *R markdown* also offers a 'caching' function that is conceptually similar, but would still require the analyses to be run from scratch on a given system. Our solution instead means that the processed data files could be downloaded locally if required, without the full analysis (which can take many hours) needing to be run.

Modern programming environments such as *R* and *python* are actively developed and updated. Over time these updates can break analysis code and prevent it from running. One solution is to 'freeze' the computational environment used for an analysis, so that the same software version is always used (see Peikert & Brandmaier, 2021; Wiebels & Moreau, 2021). Currently the most straightforward way to do this is to use a 'containerised' computing environment, such as that provided by the Docker project (https://www.docker.com/). Docker 'images' specify a basic Linux operating system, to which 'layers' of additional software can be added. The container can then be run as a virtual machine on any computing hardware, including local desktop computers, high-performance computing facilities, and cloud-based services. We used the *Rocker* images (https://rocker-project.org/), as these add a versioned *R* installation, which also freezes the package versions at a specific date. Containerised environments have further advantages, for example they do not interfere with software installed locally on a machine, meaning that an end-user can have multiple independent images running different software versions.

Finally, we aimed for analyses to be platform invariant, and tested each script on multiple operating systems, including Windows, Linux and Mac OS. This occasionally necessitated minor modifications, for example to ensure that the required fonts and other system resources were available. We also encountered an issue preventing versioned *Rocker* images running on recent Apple hardware (systems with an ARM rather than an AMD processor architecture), which we hope will be resolved in the future.

Step-by-step process

Our process began when a study author made contact and volunteered their paper to be made reproducible. This typically happened over email, and we would then organise an initial in-person meeting between one or more authors, and one or more members of the analyst team. At that meeting, the analyst(s) would outline the reproducibility process, and ask the author(s) the questions about their study summarised in the onboarding form in Appendix B. Assuming that everyone involved agreed that the work was feasible and would add value, we would then agree an approximate timescale for completion (usually within two weeks).

Following the onboarding meeting, the authors would provide all required materials. Usually this included a word processing file of the manuscript text, any data files, and any existing analysis code. The analysts met to scope out the job requirements, and divide up the work. In some cases a single analyst did the majority of the conversion to markdown format, in other cases multiple analysts worked on different parts of the document. Files were stored on a server so that everyone had access to the latest version, though often analysts worked on local copies and then uploaded these to the server.

Conversion to markdown format began by creating a template markdown document, and copying in the plain text from the manuscript. We also converted any equations and special symbols into markdown/LaTeX syntax, and replaced all references with BibTeX handles linking to records in a bibliography file. Several manuscripts involved analysis that had originally been conducted in *R*, so we copied the original code into 'chunks' of code embedded in the markdown file. Other manuscripts required translation of the analysis from other software (such as SPSS or Stata) into *R*; for these we generated the *R* code ourselves, by using the equivalent functions in *R*, and inserted it into the markdown file. In both cases, we then replaced all numerical values in the text with the appropriate variables calculated by the analysis code, such that the correct values would be inserted when the code was executed. This was the most technically challenging aspect of the work, often requiring detailed formatting of numerical values, and their insertion into tables. Where appropriate we also included code to generate figures in pdf format, and inserted these into the final manuscript along with the figure captions. We performed detailed comparisons between the submitted manuscript and the pdf file created by the markdown script, and resolved any discrepancies between the two as far as possible (see below for examples).

Once the conversion process was complete, we then created a Docker image with the required packages, and uploaded this along with the markdown code to a GitHub repository. Small data files (<100MB) were included in the repository, and larger data sets were stored on the Open Science Framework repository. The markdown file included code to automatically download the data files, and any other required resources, if they were not present locally on the computer running the code. We also included the scripts and resource files for a GitHub Actions pipeline (described below) to automatically execute the code each time the repository was updated, and ensured that the pdf was generated correctly.

The penultimate step was to test the execution of each paper on multiple 'clean' systems. We did this by downloading the markdown file from GitHub to computers running Windows, Mac OS and Linux. We then executed the script and checked that the final pdf was created correctly. The GitHub Actions workflow additionally ensured that the analysis reproduced within a Docker container, running remotely on a Linux server. Finally, the analyst team would complete the feedback form in Appendix C, and provide this to the authors, along with links to the online materials.

What went well

One of the clearest successes from the project was the strong buy-in and commitment from the authors of papers submitted for processing. Our contributing authors were in complete agreement with the ethos of computational reproducibility, and were exceptionally open and collegiate in all interactions. Several authors expressed their intention to make their future work reproducible, and in one case, the lead author even did much of the conversion to markdown format themselves after seeing how we had processed their first experiment. Most members of the project team have also started using markdown routinely in their own work. We think this will help to spread good practice across the discipline, as more examples of reproducible papers become available.

In some contexts, identifying errors in an analysis might be considered a negative outcome. However during our reproducibility work, discovering errors and inconsistencies was generally positive, because it allowed the study authors to check and, if necessary, correct any mistakes. In most cases these were minor rounding errors or transcription issues (such as the wrong number being copied into a manuscript from an SPSS or *R* output). Occasionally there were situations where an unorthodox statistical test was reported (e.g. an ANOVA with only two levels, instead of a t-test), that we thought reviewers of the paper might object to. In these cases we fed the suggestion to use a more standard approach back to the authors. We feel that identifying such errors is an important benefit of offering a service to make work reproducible, and stands apart from related approaches such as post-publication statistical auditing, which might be viewed as implicitly questioning the authors' credibility or competence (see e.g. Fiske, 2016). Error checking is most useful if it takes place before a manuscript has been submitted for review (and certainly before publication).

We discovered some useful general (i.e. non-*R*) resources for converting bibliographies to the BibTeX format used in markdown documents. In particular, we found the *Reference Extractor* tool by Rintze Zelle (https://github.com/rmzelle/ref-extractor/) helpful for manuscripts that had used Zotero to generate the references list, and the *AnyStyle* tool by Sylvester Keil (https://github.com/inukshuk/anystyle) useful for plaintext bibliographies. However for both of these websites it was still necessary to manually enter field codes for each citation in the markdown file (see e.g. https://rstudio.github.io/visual-markdown-editing/citations.html), which was somewhat labour intensive.

Perhaps the most exciting development was that we learned to implement automatic execution of manuscript code using GitHub Actions, a so-called Continuous Integration and Continuous Delivery (CI/CD) platform. By including specific files in a GitHub repository, it can be set to automatically launch a Docker environment on a remote server, and execute specific code items, each time the repository is updated. The results (i.e. a pdf) are then posted back into the main repository. This has several major advantages over running code locally. First, it places minimal requirements on the end user - no code or data needs to be downloaded, and no software needs to be installed. Edits can even be made through the GitHub web interface using devices on which programming languages cannot be installed (e.g. mobile devices such as iPads), with the results becoming available shortly afterwards. Second, because the code runs remotely on a server, the environment is completely standardised, and does not interact with any locally installed software or package versions. Third, the full history is retained, including the logs and console output for each build of the code. Finally, it is possible to request more powerful multi-core processors, with additional memory and storage capacity, for complex analyses (though this may incur charges - currently executing single-core Actions jobs is free for public repositories). After investigating several platforms for online deployment of manuscript code, GitHub Actions proved to be the most elegant and straightforward solution currently available, with many of our workflows producing a pdf in around 10-15 minutes.

What didn't go well

In most cases, we were able to exactly reproduce analyses from commercial software such as SPSS and Stata using *R*. For example, to reproduce an SPSS t-test in *R*, we used the *t.test* function, and then verified that the results were the same as those produced by SPSS. However some analyses did not reproduce perfectly. These tended to involve more complex methods, such as multiple imputation (which includes a stochastic component) and compiler average causal effects (CACE) analyses (Imbens & Angrist, 1994). We think that the implementations of these methods must differ between software packages, as although the exact numbers were not identical, the results did replicate qualitatively (i.e. significant effects remained significant). We also note that the default method in *R* for performing factorial ANOVA uses the Type 1 sums of squares calculation (in which successive predictors are added sequentially, much like in multiple regression), whereas other software such as SPSS reports the Type 3 sums of squares (in which all predictors are considered simultaneously). In order to reproduce Type 3 analyses, it was necessary to use additional packages, such as the *ez* package (https://cran.r-project.org/package=ez), or the *car* (companion to applied regression; Fox & Weisberg, 2019) package (https://cran.r-project.org/package=car). We also discovered a quirk of the *R* implementation of the Wilcoxon rank sum test, such that the order in which the variables are entered can affect the test statistic (i.e. *wilcox.test(A,B)* ≠ *wilcox.test(B,A)*). This caused some confusion in reproducing the SPSS results, although the *p*-values were the same for both orderings.

Some issues were encountered during our early attempts at preserving the computational environment. We initially used the *renv* package (https://cran.r-project.org/package=renv) to record package versions. However because we were developing the manuscript code across multiple systems, all with different package versions, the 'snapshot' directories became confusing and unwieldy. We therefore advise that *renv* should only be used once the final version of a manuscript is available, and captured on a single computer. We also discovered that versioned *Rocker* images are not available for recent Apple Mac computers that use an ARM chip (currently the M1 and M2 processors) - at time of writing (mid-2023) the latest build is available for this architecture, but specific *R* versions are not. Running in emulation mode does not appear to be possible. We hope that this situation will change as the new processor architecture becomes more widely used. Finally, we attempted to use the *Binder* system (https://mybinder.org/) to host interactive versions of a manuscript based on a Docker image. This was possible for very simple manuscripts, but for papers requiring large numbers of *R* packages to be installed or large data files it became very slow to start, and was impractical for our purposes.

We tested the manuscript build process on multiple operating systems (Windows, Mac and Linux). Whilst this was almost always a smooth process, we did experience occasional issues. One quite surprising discovery was that the widely-used *Times New Roman* font is not available by default in the Linux operating system (because the font is owned by Microsoft). The font was required to render some text in Russian (Cyrillic script), so we needed to code an exception that selected an alternative font (*Nimbus Roman*) when the script was run on a Linux system. The *LuaLatex* engine (https://www.luatex.org/) was also required for correct rendering. Additionally, to use some *R* packages we encountered occasional system-level dependency issues, particularly when using Docker containers - in most cases it was possible to add the required libraries and packages to the Dockerfile, however this took some time to resolve.

There is an occasional glitch in the *osfr* package (Wolen et al., 2020), which we used for downloading data stored on the Open Science Framework website (see e.g. https://github.com/ropensci/osfr/issues/142). For repositories with large numbers of files, sometimes specific individual files are omitted from the node listing when using the *osf_ls_files* function. It is not entirely clear why this occurs, but it does seem to be consistent across time and computer for a given repository, and so is unlikely to be an intermittent server error. One solution might be to include a hard-coded list of file identifiers rather than directly index the repository. Alternatively, large numbers of small files can be contained in a zip or other archive, and decompressed after download. However these solutions are rather inelegant, and we hope that the issue is resolved in future versions of the software.

We had occasional difficulties with certain *R* package versions when running the code on an individual machine. If a user already had one version of a package installed and the code required a different version, it would not run, requiring the user to uninstall a previous package and reinstall a specific version. These issues can all be avoided by using Docker containers, which preserve and isolate the computational environment from other software on the host computer. We provide instructions in a text file within the 'docker' subdirectory of each GitHub repository though we acknowledge that not everyone is able to use Docker. We think that the GitHub Actions solution may help increase accessibility in this instance, as the code is executed remotely on a server, avoiding any local system restrictions.

Finally, we had hoped to complete the reproducibility process on at least one paper primarily using *python* for analysis. However no suitable papers were submitted so this was not possible.

Notes on individual papers

The first paper we converted was a recently published study by Meese and Baker (2023). It was chosen because the analysis code was already written in *R*, and the data files were relatively small. The paper describes a psychophysical study on size adaptation, and reports both inferential statistics (mostly repeated measures ANOVAs) and computational modelling. The original analysis code was spread across several different files, which we collated into a single markdown document in different code chunks. There was some minor confusion around figure numbering, and working out which pieces of code were required for each analysis, but ultimately a reproducible manuscript was created with no major issues. The work took around 64 programmer-hours, as being our first paper there was a substantial overhead for most team members in learning the basics of markdown. The finished reproducible manuscript is available at: https://github.com/bakerdh/SizeAdaptation

Our second paper was a manuscript by Grigoryan et al. (2024) that had not yet been submitted to a journal. The paper reports an online study investigating Russian citizens' feelings about Russia's invasion of Ukraine in 2022. Again the code had been written in *R*, and did not involve large data files. Analyses were primarily linear regressions, reported in tables (we learned some useful features of the *kableExtra* package when typesetting these). Progress was substantially faster than for the first paper, taking around 24 programmer-hours. Some minor technical issues to do with fonts needed to be resolved so that the manuscript would build properly on all operating systems. The finished manuscript is available at: https://github.com/gitgrigolus/political_action

A manuscript by de Bruin et al. (2023), reporting a psycholinguistic study on language switching, was noteworthy because it was converted under time pressure. The paper was due to be submitted to a journal special issue, and we had around 10 days to create the markdown version. This was greatly aided by the fact that the original *R* code was extremely well structured and commented, making it straightforward to extract the relevant results. However the original code included several mixed effects models that did not properly converge, and took a long time to run. As these were not reported in the manuscript, we commented out these lines of code, but they remained in place for interested readers. The reproducible version of the manuscript is available at: https://github.com/AMTdeBruin/Bilingual-switching-ageing

The study by Mak et al. (2023) was interesting because it was a Stage 2 registered report, which already had in-principle acceptance at the journal *Royal Society Open Science*. The study concerned the effects of sleep on false memories in the classic Deese-Roediger-McDermott paradigm, and involved a large online sample (N=488). The manuscript was written in a somewhat more structured way than typical papers in psychology. The analysis had been completed using *R*, and was relatively straightforward to convert into markdown format, despite being very sophisticated, including both frequentist and Bayesian methods, as well as supplementary simulations. Some of the analyses took a long time to run (around 6 hours), and so we inserted level flags into the code so that the user could choose whether to execute the full analysis, or load in a precomputed results file. This saved substantial time during development. The final reproducible manuscript is available at: https://github.com/bakerdh/sleepDRM

The paper by Baxter and Hobson (2024) was also a Stage 2 registered report, on the topic of face processing in autism. Most of the analysis had been conducted using SPSS, and involved a series of stepwise linear regressions. Although functions to perform stepwise regression are available in *R* (i.e. the *stepAIC* function from the *MASS* package; Venables & Ripley, 2002), these report different statistics from the standard SPSS implementation. Reproducing the analyses in the paper therefore required separate calculation of each linear model (i.e. each step), followed by model comparison using the *anova* function from the base *stats* package. This fully reproduced the analyses reported in the paper, and can be seen in the reproducible manuscript available at: https://github.com/bakerdh/EmotionFaceAutism

Brennan et al. (2024) report the results of six experiments using a range of techniques, to test specific hypotheses about dehumanization and character traits. Most of the analyses were conducted in SPSS, and these reproduced exactly in *R*. The figures had been generated in *R*, and involved some ingenious solutions to the limits of the *ggplot2* package. Despite being a substantial paper, we were able to complete the work relatively quickly with no major issues. The final markdown file is available at: https://github.com/robraonain/DualModel-Criminals

A study by Lee et al. (2023) compared 'mind-mindedness' between British and South Korean mothers (mind-mindedness is the extent to which caregivers consider the mental states of infants). The analyses were originally conducted in SPSS, and primarily used MANCOVA (multivariate analysis of covariance). We were able to reproduce the results in *R*, though we did need to use the *Anova* and *Manova* functions from the *car* package to specify the Type 3 sums of squares values that are typically returned by SPSS (see the "What Didn't Go Well" section above). We also found it quite challenging to extract the results of these tests. Although the functions were able to print the results to the command window, it was more difficult to save them to a variable for automatic insertion into the manuscript text. We used the *capture.output* function from the base *R utils* package to save the results to a string, and then extracted the appropriate values. This was substantially more complicated than extracting the results of most statistical tests. The final manuscript is available at: https://github.com/yl2944/parental-mentalization-across-cultures-

We processed one paper, by Larkin et al. (2024), where much of the analysis was performed in Stata (with some additional analyses conducted in SPSS). The paper described a randomised controlled trial of a smartphone app intervention intended to increase mind-mindedness in new mothers. The Stata script was provided, which made it possible to reproduce the analyses in *R*. Most of the straightforward statistical tests, such as correlations, regressions and t-tests, replicated exactly. However there were some more sophisticated analyses, involving multiple imputation and CACE analysis, where the implementation appeared to differ between Stata and *R*. Our markdown script can therefore be considered a qualitative replication, in that the main conclusions were unaffected (i.e. effects that were significant in the Stata analysis remained so in the *R* analysis), despite differences in the values calculated. The reproducible manuscript is available at: https://github.com/bakerdh/testbabymind

The paper by Hansford et al. (2024) explored multisensory modulation of illusory finger resizing. The original analysis had been partly written in Matlab, and involved using a simple algorithm to determine the position of the index finger in still images of the hand. This code was translated line-by-line into *R*, and was able to exactly reproduce the original results. The remainder of the analysis had been written using *R*, and so was straightforward to convert into markdown format. One minor difficulty concerned augmenting graphs created using the *ggplot2* package with additional text and other components (such as arrows) - in the original version of the paper this had been done using external software. Our eventual solution for this was to create a blank pdf document, and use the base plotting functions to add the required features, followed by the *grid.arrange* function from the *gridExtra* package (https://cran.r-project.org/package=gridExtra) to superimpose the graphs over the top. The markdown files are available at: https://github.com/KJHansford/AuditoryResizing

The final study, by Segala et al. (2023), investigated binocular combination of light signals and involved analysis of several diverse data types. These included EEG data, pupillometry data, and psychophysical data, spread across four experiments, and all analysed using *R*. The raw data for the entire study required around 48GB of storage space, and took many hours to analyse from scratch. We therefore included a *processdata* flag with four possible values, so that users could choose the extent of the analysis required. We also added an option to delete the raw data for each participant after processing, to save storage space on capacity limited systems. We think these features will be useful for other studies involving large amounts of raw data, such as neuroimaging studies. The analysis involved Bayesian computational modelling using Stan (Carpenter et al., 2017), which took approximately 24 hours to run from scratch. We added a further option flag to determine whether the modelling should be conducted, or the results loaded from a pre-generated file. One restriction of the GitHub Actions build pipeline is that analyses must complete within a 6 hour time window. This means that running the full pipeline from scratch was not possible within GitHub, and instead the code would need to be downloaded and run locally. The repository for this study is available at: https://github.com/bakerdh/PupillometryEEG

The future of computational reproducibility

Increasing numbers of journals are now requiring papers to share analysis code (Abbasi, 2023), and many already mandate sharing data. However simply sharing code and data does not guarantee computational reproducibility (Crüwell et al., 2023; Hardwicke et al., 2021), and very few journals currently (in 2023) include reproducibility checks as part of the review process (*Meta-Psychology* being a rare exception). Making an entire manuscript reproducible exceeds most current journal requirements, and makes verification of reproducibility much more straightforward. A manuscript written in markdown is *self-evidently reproducible*, because the markdown file specifies the analyses, and how the outputs are presented in the manuscript. If the code has been executed using a cloud-based service such as GitHub Actions, there is also a fully documented audit trail of exactly what has been done (including storage of the server logs for the computer that ran the analysis). We think that self-evidently reproducible manuscripts of this type should render explicit reproducibility checks either extremely straightforward, or perhaps even unnecessary. They will also greatly facilitate the process of formal statistical audit of manuscripts (see e.g. Nuijten & Wicherts, 2024), though we note that our aim in this project was to reproduce the analyses conducted by each study's authors, rather than to provide commentary on whether we thought this analysis was appropriate. We consider that to be the responsibility of the journal reviewing and editorial process.

Despite the many benefits of reproducible manuscripts, there will always be some types of analysis that are inherently unsuitable for making reproducible from the raw data. For example, studies using primarily qualitative methods, and those that require manual coding or processing cannot be easily replaced by automated methods. This might include human ratings of video recordings, or manual segmentation of MRI scans, for example. In such situations, a pragmatic solution would be to take the results from the manual stages, and use this as the data on which a reproducible analysis is based. However, rapid progress is being made in using artificial intelligence tools, including deep learning, to automatically code complex data such as images and movies (see e.g. *DeepLabCut*; Mathis et al., 2018). It is therefore likely that increasingly accurate automation might replace human labour in some areas of data analysis, with the added benefit that these analyses will then become more reproducible.

We did not process any papers that used fMRI during this pilot project. The analysis methods for MRI are relatively complex, and there are often data sharing restrictions on MRI images, primarily because participants can in principle be identified from their scan data. We note that the fMRI community is now embracing standardised data formats (i.e. BIDS; Gorgolewski et al., 2016) and analysis pipelines (i.e. *fMRIprep*; Esteban et al., 2019), which will aid substantially in reproducibility. However it may not currently be feasible to use the pipeline we propose here to analyse large MRI datasets from scratch. In particular the GitHub Actions workflows have a time limit of 6 hours, and most MRI preprocessing requires substantially longer (i.e. *FreeSurfer* reconstruction typically takes around 12 hours per participant). An alternative solution might be to conduct the group-level analyses on preprocessed data, reducing the time and storage requirements. Alternatively, deployment on high performance computing facilities or cloud platforms would permit the analysis of many large data sets in parallel. We look forward to future technical developments in this direction.

Another barrier to reproducibility is that many software packages require a commercial license, and/or cannot be scripted or automated. A substantial proportion of scientific research uses commercial programming platforms such as Matlab and Mathematica, often relying on specialist toolboxes that are only available in those languages. In some cases it may be possible to translate code into more open languages (as we did for the Hansford et al. study), and we note again that artificial intelligence tools show increasing utility in this area. In our pilot study we were able to successfully reproduce analyses that were originally conducted using commercial statistical packages such as SPSS and Stata, by running equivalent analyses in *R*. Furthermore, a recent development with some GUI-based statistics packages such as JASP (https://jasp-stats.org/) and Jamovi (https://www.jamovi.org/) is that they allow analyses to be exported as *R* syntax (or will do in the near future), which opens the door to automation and reproducibility. We hope that over time more researchers realise the benefits of FOSS, and move their analysis pipelines away from commercial software. We are aware that GitHub itself is a commercial entity (currently owned by Microsoft), that might restrict its free functionality at some point in the future, making it less useful for hosting repositories without charge.

Our pilot project was funded for 6 months, and a natural question is how we could continue this work beyond the end of that period. One option is to seek further grant funding, so that 'reproducibility as a service' can be provided at no cost to a paper's authors. However without very substantial funding this would necessarily remain limited in reach, most likely to our own department. Another option would be to charge a fee for each paper that is processed. We ran an informal poll on social media that indicated relatively few respondents had the financial resources to pay for such a service. Over time this may change, and making papers reproducible could be costed into grant proposals if a commercial service were available. Alternative routes to sustainability might include direct support from academic institutions (Universities are increasingly assessed on open research practices), charities, or national interest groups such as the UK reproducibility network (and international equivalents). Offering training courses to disseminate these methods may also prove useful. A final possibility is support from journal publishers, who should have a vested interest in quality control of their outputs, and in many cases have sufficient financial resources to make investments in hiring technically capable staff.

The pilot project made ten papers reproducible. In doing so, we hope that the authors of those papers have experienced the benefits of reproducibility, and might aim to make more of their future work reproducible in a similar way. Often seeing an example of something that is familiar helps us to put new concepts into a meaningful framework, and we suspect this to be the case for markdown scripting. Seeing how the analyses and reporting conventions in a particular sub-field can be made reproducible should therefore serve as a template for future studies. It is also the case that all members of the project team have acquired extensive skills and experience in creating reproducible documents, and in most cases have begun to apply this to our own research. In this way we expect that good practice will spread throughout the discipline, gradually changing research culture and norms around computational reproducibility. We believe that communication of scientific results should be as open and transparent as possible, and propose that computationally reproducible manuscripts are the ‘gold standard’ in realising this goal. The tools to produce them are now freely available, and we hope that the training materials we developed during this project (see Appendix A) will help others to acquire the necessary skills to make their own work reproducible in this way.

Author Contact

Corresponding author, Daniel H. Baker, email: daniel.baker@york.ac.uk
ORCID IDs:
DHB: 0000-0002-0161-443X
KJH: 0000-0002-5738-6843
BPAQ: 0000-0001-8595-4433
FGS: 0000-0002-4982-8023
ELW-E: 0000-0002-8926-4672

Acknowledgments

We thank all of the study authors who contributed their work to our pilot study. We also wish to thank Dr Lincoln Colling for showing us the Github Actions workflow.

Conflict of Interest and Funding

The authors declare no current conflicts of interest, but reserve the right to offer the services described in the manuscript on a commercial basis in the future. Supported by an Enhancing Research Culture award from Research England (via the University of York).

Author Contributions

DHB: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing - original draft, and Writing - review & editing. **MB**: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, and Writing - review & editing. **KJH**: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, and Writing - review & editing. **BPAQ**: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, and Writing - review & editing. **FGS**: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, and Writing - review & editing. **ELW-E**: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, and Writing - review & editing.

Open Science Practices

This article earned the Open Materials badge for making the materials openly available. This is a narrative paper and as such was not reproduced nor eligible for other badges. The entire editorial process, including the open reviews, is published in the online supplement.

References

Abbasi, K. (2023). A commitment to act on data sharing. *BMJ*, p1609. https://doi.org/10.1136/bmj.p1609

Baker, D. H. (2021). Statistical analysis of periodic data in neuroscience. *Neurons, Behavior, Data analysis, and Theory*, *5*(3), 1–18. https://doi.org/10.51628/001c.27680

Baker, D. H., Vilidaite, G., & Wade, A. R. (2021). Steady-state measures of visual suppression. *PLOS Computational Biology*, *17*(10), e1009507. https://doi.org/10.1371/journal.pcbi.1009507

Baxter, N., & Hobson, H. (2024). The role of emotional factors in face processing abilities in autism spectrum conditions. *Research in Autism Spectrum Disorders*, *115*, 102400. https://doi.org/10.1016/j.rasd.2024.102400

Brennan, R. A., Enock, F. E., & Over, H. (2024). Attribution of undesirable character traits, rather than trait-based dehumanization, predicts punishment decisions. *R Soc Open Sci*, *11*(7), 240087. https://doi.org/10.1098/rsos.240087

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., & Riddell, A. (2017). *Stan*: A Probabilistic Programming Language. *Journal of Statistical Software*, *76*(1). https://doi.org/10.18637/jss.v076.i01

Crüwell, S., Apthorp, D., Baker, B. J., Colling, L., Elson, M., Geiger, S. J., Lobentanzer, S., Monéger, J., Patterson, A., Schwarzkopf, D. S., Zaneva, M., & Brown, N. J. L. (2023). What's in a Badge? A Computational Reproducibility Investigation of the Open Data Badge Policy in One Issue of *Psychological Science*. *Psychological Science*, *34*(4), 512–522. https://doi.org/10.1177/09567976221140828

de Bruin, A., Kressel, H., & Hemmings, D. (2023). A comparison of language control while switching within versus between languages in younger and older adults. *Sci Rep*, *13*(1), 16740. https://doi.org/10.1038/s41598-023-43886-1

Esteban, O., Markiewicz, C. J., Blair, R. W., Moodie, C. A., Isik, A. I., Erramuzpe, A., Kent, J. D., Goncalves, M., DuPre, E., Snyder, M., Oya, H., Ghosh, S. S., Wright, J., Durnez, J., Poldrack, R. A., & Gorgolewski, K. J. (2019). fMRIPrep: A robust preprocessing pipeline for functional MRI. *Nature Methods*, *16*(1), 111–116. https://doi.org/10.1038/s41592-018-0235-4

Fiske, S. (2016). A call to change science's culture of shaming. *Observer*, *29*(9), 5–11. https://www.psychologicalscience.org/observer/a-call-to-change-sciences-culture-of-shaming

Fox, J., & Weisberg, S. (2019). *An R Companion to Applied Regression* (Third Edition). Sage. https://socialsciences.mcmaster.ca/jfox/Books/Companion/

Gorgolewski, K. J., Auer, T., Calhoun, V. D., Craddock, R. C., Das, S., Duff, E. P., Flandin, G., Ghosh, S. S., Glatard, T., Halchenko, Y. O., Handwerker, D. A., Hanke, M., Keator, D., Li, X., Michael, Z., Maumet, C., Nichols, B. N., Nichols, T. E., Pellman, J., . . . Poldrack, R. A. (2016). The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Scientific Data*, *3*(1), 160044. https://doi.org/10.1038/sdata.2016.44

Grigoryan, L., Ponizovskiy, V., Weißflog, M. I., Osin, E., & Lickel, B. (2024). Guilt, shame, and antiwar action in an authoritarian country at war. *Political Psychology*, pops.12985. https://doi.org/10.1111/pops.12985

Hansford, K. J., Baker, D. H., McKenzie, K. J., & Preston, C. E. (2024). Multisensory processing and proprioceptive plasticity during resizing illusions. *Experimental Brain Research*, *242*, 451–462. https://doi.org/10.1007/s00221-023-06759-7

Hardwicke, T. E., Bohn, M., MacDonald, K., Hembacher, E., Nuijten, M. B., Peloquin, B. N., deMayo, B. E., Long, B., Yoon, E. J., & Frank, M. C. (2021). Analytic reproducibility in articles receiving open data badges at the journal *Psychological Science* : An observational study. *Royal Society Open Science*, *8*(1), 201494. https://doi.org/10.1098/rsos.201494

Imbens, G. W., & Angrist, J. D. (1994). Identification and Estimation of Local Average Treatment Effects. *Econometrica*, *62*(2), 467. https://doi.org/10.2307/2951620

Larkin, F., Oostenbroek, J., Lee, Y., Hayward, E., Fernandez, A., Wang, Y., Mitchell, A., Li, L. Y., & Meins, E. (2024). A smartphone app effectively facilitates mothers mind-mindedness: A randomized controlled trial. *Child Dev*, *95*(3), 831–844. https://doi.org/10.1111/cdev.14039

Lee, Y. J., Meins, E., & Larkin, F. (2023). Parental Mentalization Across Cultures: Mind-mindedness and Parental Reflective Functioning in British and South Korean Mothers. *PsyArXiv*. https://doi.org/10.31234/osf.io/qx9mh

Mak, M. H. C., O'Hagan, A., Horner, A. J., & Gaskell, M. G. (2023). A registered report testing the effect of sleep on Deese-Roediger-McDermott false memory: Greater lure and veridical recall but fewer intrusions after sleep. *Royal Society Open Science*, *10*(12), 220595. https://doi.org/10.1098/rsos.220595

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, *21*(9), 1281–1289. https://doi.org/10.1038/s41593-018-0209-y

Meese, T. S., & Baker, D. H. (2023). Object Image Size Is a Fundamental Coding Dimension in Human Vision: New Insights and Model. *Neuroscience*, *514*, 79–91. https://doi.org/10.1016/j.neuroscience.2023.01.025

Nuijten, M. B., & Wicherts, J. M. (2024). Implementing statcheck during peer review is related to a steep decline in statistical-reporting inconsistencies. *Advances in Methods and Practices in Psychological Science*, *7*(2). https://doi.org/10.1177/25152459241258945

Obels, P., Lakens, D., Coles, N. A., Gottfried, J., & Green, S. A. (2020). Analysis of Open Data and Computational Reproducibility in Registered Reports in Psychology. *Advances in Methods and Practices in Psychological Science*, *3*(2), 229–237. https://doi.org/10.1177/2515245920918872

Peikert, A., & Brandmaier, A. M. (2021). A Reproducible Data Analysis Workflow. *Quantitative and Computational Methods in Behavioral Sciences*, *1*, e3763. https://doi.org/10.5964/qcmb.3763

Rouder, J. N., & Haaf, J. M. (2018). Power, Dominance, and Constraint: A Note on the Appeal of Different Design Traditions. *Advances in Methods and Practices in Psychological Science*, *1*(1), 19–26. https://doi.org/10.1177/2515245917745058

Segala, F. G., Bruno, A., Martin, J. T., Aung, M. T., Wade, A. R., & Baker, D. H. (2023). Different rules for binocular combination of luminance flicker in cortical and subcortical pathways. *eLife*, *12*, RP87048. https://doi.org/10.7554/eLife.87048

Venables, W., & Ripley, B. (2002). *Modern Applied Statistics with S* (Fourth Edition). Springer. https://www.stats.ox.ac.uk/pub/MASS4/

Wiebels, K., & Moreau, D. (2021). Leveraging Containers for Reproducible Psychological Research. *Advances in Methods and Practices in Psychological Science*, *4*(2), 25152459211017853. https://doi.org/10.1177/25152459211017853

Wolen, A., Hartgerink, C., Hafen, R., Richards, B., Soderberg, C., & York, T. (2020). Osfr: An R Interface to the Open Science Framework. *Journal of Open Source Software*, *5*(46), 2071. https://doi.org/10.21105/joss.02071

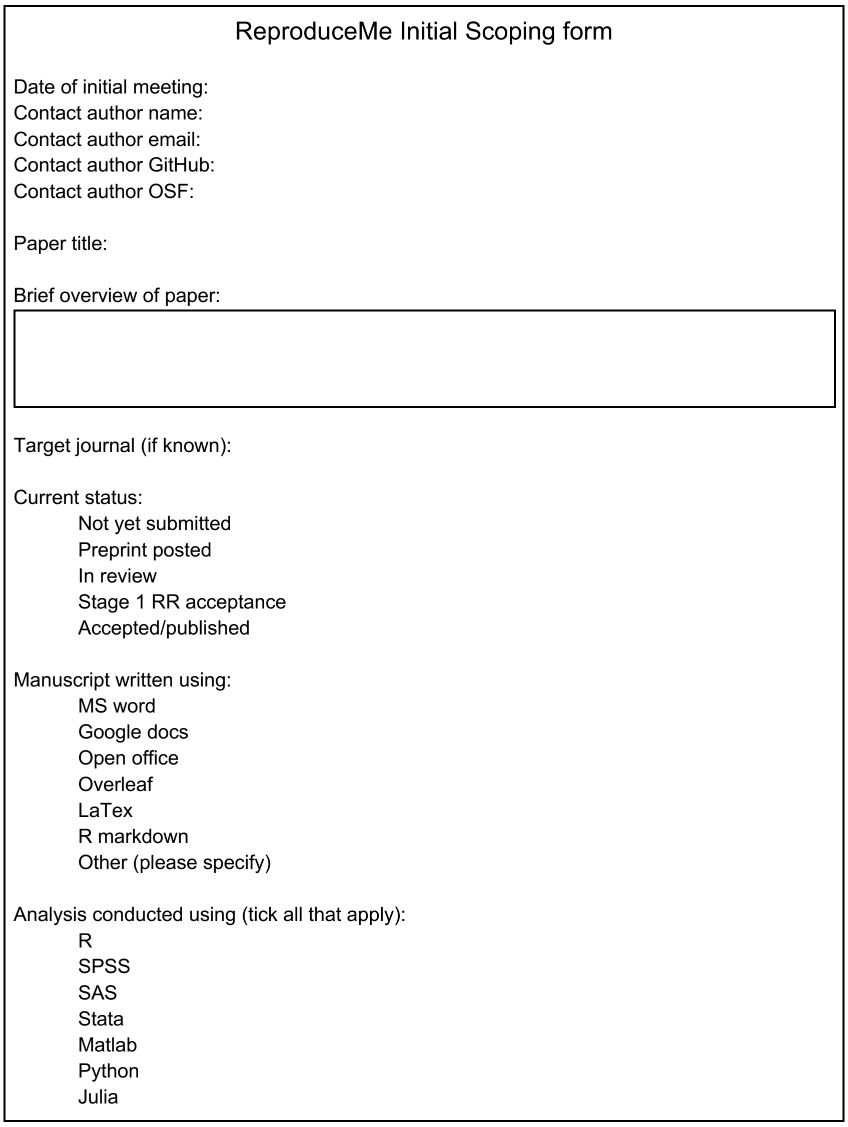
Appendices

Appendix A

The training materials produced as part of this project are available at: https://github.com/bakerdh/ReproduceMe

Appendix B

Template Onboarding form, which was completed at or after the initial meeting between authors and the project team.



ReproduceMe Initial Scoping form

Date of initial meeting:
Contact author name:
Contact author email:
Contact author GitHub:
Contact author OSF:

Paper title:

Brief overview of paper:

Target journal (if known):

Current status:
- Not yet submitted
- Preprint posted
- In review
- Stage 1 RR acceptance
- Accepted/published

Manuscript written using:
- MS word
- Google docs
- Open office
- Overleaf
- LaTex
- R markdown
- Other (please specify)

Analysis conducted using (tick all that apply):
- R
- SPSS
- SAS
- Stata
- Matlab
- Python
- Julia

JASP
Jamovi
Other (please list)

Are figures generated automatically, or assembled manually?

Data format:
CSV
XLSX
SPSS
.mat
.RData
Python
NIFTI
Other (please specify)

Approximate total data size:
0 - 100MB
100MB - 1GB
1GB - 50GB
>50GB

If data/code are already shared online, please provide the URL(s):

Do the data need to be modified (i.e. de-anonymised) before sharing?

Are there any restrictions on data sharing?

Can we convert data to a more open format if appropriate?

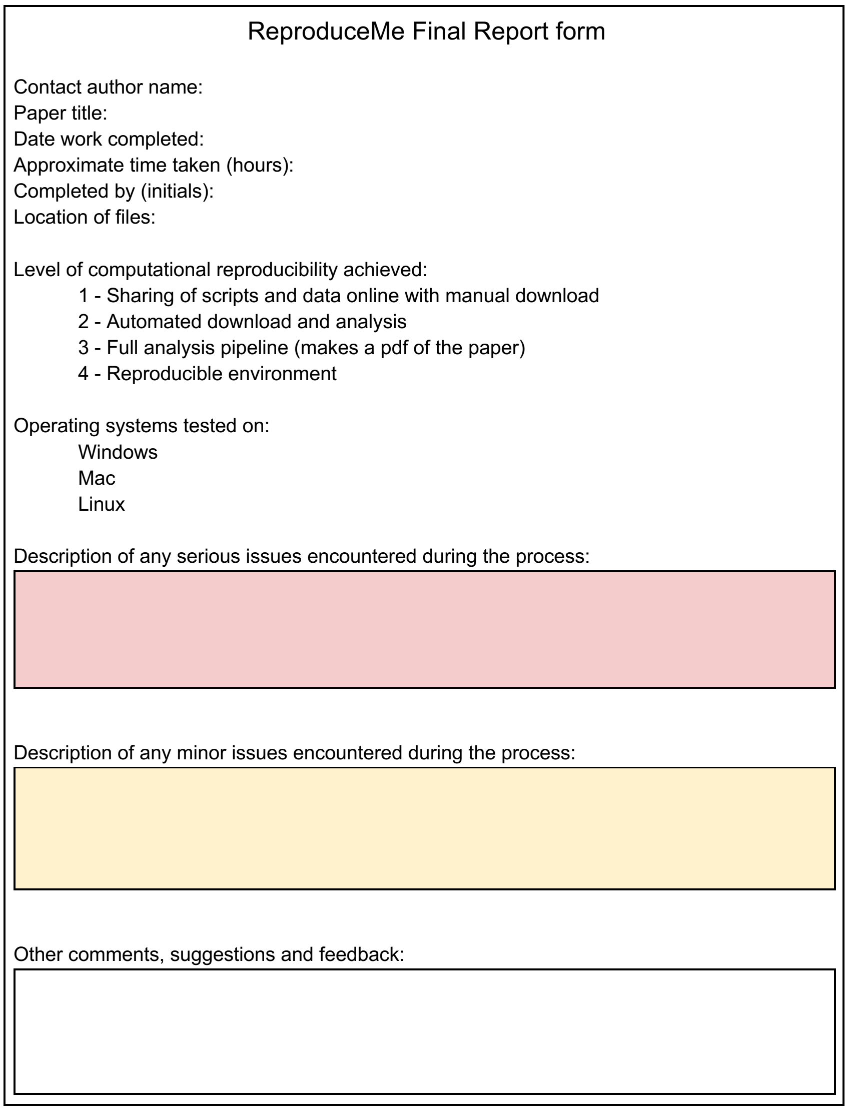
Materials provided:

Desired level of computational reproducibility:
1 - Sharing of scripts and data online with manual download
2 - Automated download and analysis
3 - Full analysis pipeline (makes a pdf of the paper)
4 - Reproducible environment

Additional comments or requests:

Appendix C

Template for the Feedback form, provided to authors when work had finished.



ReproduceMe Final Report form

Contact author name:
Paper title:
Date work completed:
Approximate time taken (hours):
Completed by (initials):
Location of files:

Level of computational reproducibility achieved:
- 1 - Sharing of scripts and data online with manual download
- 2 - Automated download and analysis
- 3 - Full analysis pipeline (makes a pdf of the paper)
- 4 - Reproducible environment

Operating systems tested on:
- Windows
- Mac
- Linux

Description of any serious issues encountered during the process:

Description of any minor issues encountered during the process:

Other comments, suggestions and feedback: