

COMMENTARY

# A beginner's guide to conducting reproducible research in ecology, evolution, and conservation

Jesse M. Alston<sup>1,2\*</sup>, Jessica A. Rick<sup>1,3</sup>

<sup>1</sup>Program in Ecology, University of Wyoming, Laramie, WY, USA

<sup>2</sup>Department of Zoology and Physiology, University of Wyoming, Laramie, WY, USA

<sup>3</sup>Department of Botany, University of Wyoming, Laramie, WY, USA

**Correspondence**

Jesse Alston, University of Wyoming, 1000 E University Dr., Laramie, WY 82072

Email: jalston@uwyo.edu

**Running headline:** Reproducible research

## 1 ABSTRACT

- 2 1. Reproducible research is burgeoning in popularity among scientists worried about the "replication crisis", yet  
3 research in the fields of ecology, evolution, and conservation science remains largely irreproducible.
- 4 2. In this commentary, we make the case for why all research should be reproducible, explain why research is often  
5 not reproducible, and offer a simple framework that researchers can use to make their research more reproducible.
- 6 3. Researchers can make their work more reproducible by improving data management practices, writing more ac-  
7 cessible and readable code, and increasing use of the many available platforms for sharing data and code.
- 8 4. While reproducible research is often associated primarily with a set of advanced tools for sharing data and code,  
9 reproducibility is just as much about solidifying a set of simple work habits that are already widely acknowledged  
10 as best practices for research. Increasing the reproducibility of research in ecology, evolution, and conservation  
11 will increase rigor, trustworthiness, and transparency in these fields while benefiting both practitioners of repro-  
12 ducible research and their fellow researchers.

## 13 KEYWORDS

14 code, data management, data repository, open science, reproducible research, replication, scientific publication

## 15 INTRODUCTION

16 Replication is a fundamental tenet of science, but there is increasing fear among scientists that too few scientific  
17 studies can be replicated. This has been termed the "replication crisis" (Ioannidis, 2005; Schooler, 2014). Scientific  
18 papers often include inadequate detail to enable reproduction (Haddaway and Verhoeven, 2015), many attempted  
19 replications of well-known scientific studies have failed in a wide variety of disciplines (Bohannon, 2015; Hewitt, 2012;  
20 Moonesinghe et al., 2007; Open Science Collaboration, 2015), and rates of paper retractions are increasing (Cokol  
21 et al., 2008; Steen et al., 2013). Because of this, researchers are working to develop new ways for researchers, research  
22 institutions, research funders, and journals to overcome this problem (Peng, 2011; Sandve et al., 2013; Stodden et al.,

23 2013).

24 Because replicating studies with new independent data is expensive, rarely published in high-impact journals, and  
25 sometimes even methodologically impossible, "reproducible research" is often suggested as a method for increasing  
26 our ability to assess the validity and rigor of scientific results (Peng, 2011). Research is reproducible when others can  
27 reproduce scientific results given only the original data, code, and documentation. This commentary describes basic  
28 requirements for reproducible research in ecology, evolution, and conservation science. In it, we make the case for  
29 why all research should be reproducible, explain why research is often not reproducible, and present a simple three-  
30 part framework all researchers can use to make their research more reproducible. These principles are applicable to  
31 researchers working in all types of research with data sets of all sizes and levels of complexity.

## 32 **WHY DO REPRODUCIBLE RESEARCH?**

### 33 **Reproducible research benefits those who do it**

34 Reproducible research is a by-product of careful attention to detail throughout the research process, and allows re-  
35 searchers to ensure that they can repeat the same analysis multiple times with the same results, at any point in that  
36 process. Because of this, researchers who conduct reproducible research are the primary beneficiaries of this practice.  
37 This is true for several reasons.

38 First, reproducible research helps researchers remember how and why they performed specific analyses during  
39 the course of a project. This enables easier explanation of work to collaborators, supervisors, and reviewers, and it  
40 allows collaborators to conduct supplementary analyses more quickly and more efficiently.

41 Second, reproducible research enables researchers to quickly and simply alter analyses and figures. This is often  
42 requested by supervisors, collaborators, and reviewers across all stages of a research project, and expediting this  
43 process saves substantial amounts of time. When analyses are truly reproducible, creating a new figure may be as  
44 easy as changing one value in a line of code and re-running a script, rather than spending hours recreating a figure  
45 from scratch.

46 Third, reproducible research enables quick reconfiguration of previously conducted research tasks so that new  
47 projects that require similar tasks become much simpler and easier. Science is an iterative process, and many of the  
48 same tasks are performed over and over. Conducting research reproducibly enables researchers to re-use earlier  
49 materials (e.g., analysis code, file organization systems) to execute these common research tasks more efficiently in  
50 subsequent iterations.

51 Fourth, conducting reproducible research sends peers a strong signal of rigor, trustworthiness, and transparency.  
52 This can increase the quality and speed of peer review, because reviewers can directly access the analytical process  
53 described in a manuscript. Peer reviewers' work becomes easier and they may be able to answer methodological  
54 questions without asking the authors. It also protects researchers from accusations of research misconduct due to  
55 analytical errors, because it is unlikely that researchers would openly share fraudulent code and data with the rest of  
56 the research community. Finally, it increases the probability that errors are caught during the peer-review process,  
57 decreasing the likelihood of corrections or retractions after publication.

58 In addition to improving the analytical process, reproducible research increases paper citation rates (McKiernan  
59 et al., 2016; Piwowar et al., 2007) and allows other researchers to cite code and data in addition to publications. This  
60 enables a given research project to have more impact than it would if the data or methods were hidden from the  
61 public. For example, researchers can re-use code from a paper with similar methods and organize their data in the  
62 same manner as the original paper, then cite code from the original paper in their manuscript. Another researcher  
63 may conduct a meta-analysis on the phenomenon described in the two research papers, and thus use and cite both  
64 the two papers and the data from those papers in their meta-analysis. Papers are also more likely to be cited in these  
65 re-use cases if full information about data and analyses are available (Culina et al., 2018; Whitlock, 2011).

## 66 **Reproducible research benefits the research community**

67 Reproducible research also benefits others in the scientific community. Sharing data, code, and detailed research  
68 methods and results leads to faster progress in methodological development and innovation because research is more  
69 accessible to more scientists (Mislán et al., 2016; Parr and Cummings, 2005; Roche et al., 2015). These benefits spread

70 through the community a few different ways.

71 First, reproducible research allows others to learn from your work. Scientific research has a steep learning curve,  
72 and allowing others to access data and code gives them a head start on performing similar analyses. For example,  
73 junior researchers can use code shared with the research community by more senior researchers to learn how to  
74 perform advanced analytical workflows. This allows junior researchers to conduct research that is more rigorous  
75 from the outset, rather than having to spend months or years trying to figure out "best practices" through trial and  
76 error. Modifying existing resources can also save time and effort for experienced researchers. For example, even  
77 experienced coders can modify existing code faster than they can write code from scratch. Sharing code thus allows  
78 experienced researchers to perform similar analyses more quickly.

79 Second, reproducible research allows others to understand and reproduce a researcher's work. Allowing others  
80 to access data and code makes it easier for other scientists to perform follow-up studies to increase the strength  
81 of evidence for the same phenomenon. It also increases the likelihood that similar studies are compatible with one  
82 another, and that all of these studies can provide evidence in support of or in opposition to the same concept. In addi-  
83 tion, sharing data and code increases the utility of these studies for meta-analyses that are important for generalizing  
84 and contextualizing the findings of studies on a topic. Meta-analyses in ecology and evolutionary biology are often  
85 hindered by incompatibility of data between studies, or lack of documentation for how those data were obtained  
86 (Culina et al., 2018; Stewart, 2010). Well-documented, reproducible findings enhance the likelihood that data can be  
87 used in future meta-analyses (Gerstner et al., 2017).

88 Third, reproducible research allows others to protect themselves from your mistakes. Mistakes do happen in  
89 science. Allowing others to access data and code gives them a better chance to critically analyze the work, which  
90 can lead to coauthors discovering mistakes during the revision process or other scientists discovering mistakes after  
91 publication. This prevents mistakes from compounding over time and provides protection for collaborators, research  
92 institutions, funding organizations, journals, and others who may be affected when such mistakes happen.

## 93 BARRIERS TO REPRODUCIBLE RESEARCH

94 There are a number of valid reasons that most research is not reproducible. Rapidly developing technologies and  
95 analytical tools, novel interdisciplinary approaches, unique ecological study systems, and increasingly complex data  
96 sets and research questions hinder reproducibility, as does pressure on scientists to publish novel research quickly  
97 in high-impact journals. This multitude of barriers can be simplified into four primary themes: (1) complexity, (2)  
98 technological change, (3) human error, and (4) concerns over intellectual property rights. Each of these concerns can  
99 contribute to making research less reproducible, and all are valid to some degree. But each of these factors can also  
100 be addressed easily via well-developed tools, protocols, and institutional norms concerning reproducible research.

101 **Complexity.** – Science is difficult, and scientific research requires specialized (and often proprietary) knowledge  
102 and tools that may not be available to everyone who would like to reproduce research. For example, analyses of ge-  
103 nomic data require researchers to possess a vast base of knowledge about statistical methodologies and the molecular  
104 architecture of DNA, and genomic analyses are therefore difficult to reproduce for those with limited knowledge of  
105 the subject. Some analyses may require high-performance computing clusters that use several different programming  
106 languages and software packages, or that are designed for specific hardware configurations. Other analyses may be  
107 performed using proprietary software programs such as SAS statistical software (SAS Institute Inc., Cary, NC, USA)  
108 or ArcGIS (Esri, Redlands, CA, USA) that require expensive software licenses. Lack of knowledge, lack of institutional  
109 infrastructure, and lack of funding all make research less reproducible. However, these issues can be mitigated fairly  
110 easily. Researchers can cite primers on complex subjects or analyses to reduce knowledge barriers. They can also  
111 thoroughly annotate analytical code with comments explaining each step in an analysis, or provide extensive doc-  
112 umentation on research software. Using open software (when possible) makes research more accessible for other  
113 researchers as well.

114 **Technological change.** – Hardware and software both change over time, and they often change quickly. When old  
115 tools become obsolete, research becomes less reproducible. For example, reproducing research performed in 1960  
116 using that era's computational tools would require a completely new set of tools today. Even research performed just  
117 a few years ago may have been conducted using software that is no longer available or is incompatible with other

118 software that has since been updated. One minor update in a piece of software used in one minor analysis in an  
119 analytical workflow can render an entire project less reproducible, even if researchers follow best practices. However,  
120 this too can be mitigated by using established tools in reproducible research. Careful documentation of versions of  
121 software used in analyses is a baseline requirement that anyone can meet. There are also more advanced tools that can  
122 help overcome such challenges in making research reproducible, including software containers, which are described  
123 in further detail below.

124 **Human error.** — Though fraudulent research is often cited as reason to make research more reproducible (e.g.,  
125 Crocker and Cooper 2011; Laine et al. 2007; Ioannidis 2005), many more innocent reasons exist as to why research  
126 is often difficult to reproduce. People forget small details of how they performed analyses. They fail to describe  
127 data collection protocols or analyses completely despite their best efforts and multiple reviewers checking their work.  
128 They perform sloppy analyses because they just want to be done with a project that feels like it is taking forever to  
129 complete. Science is performed by fallible humans, and a wide variety of common events can render research less  
130 reproducible.

131 While not all of these challenges can be avoided by performing research reproducibly, a well-documented research  
132 process can guard against small errors and sloppy analyses. For example, carefully recording details such as when and  
133 where data were collected, what decisions were made during data collection, and what labeling conventions were  
134 used can make a huge difference in making sure that those data can later be used appropriately, or even re-purposed.  
135 Unintentional errors often occur during the data wrangling stage of a project, and these can be mitigated by keeping  
136 multiple copies of data to prevent data loss, carefully documenting the process for converting raw data into clean  
137 data, and double-checking a small test set of data before manipulating the data set as a whole.

138 **Intellectual property rights.** — Researchers often hesitate to share data and code because doing so may allow  
139 other researchers to use data and code incorrectly or unethically. Other researchers may use publicly available data  
140 without notifying authors, leading to incorrect assumptions about the data that cause subsequent analyses to be  
141 incorrect. Researchers may use publicly available data or code without citing the original data owners or code writers,  
142 who then do not receive proper credit for gathering expensive data or writing time-consuming code. Researchers  
143 may want to conceal data from others so that they can perform new analyses on those data in the future without

144 worrying about others scooping them using the shared data. Rational self-interest can lead to hesitation to share data  
145 and code via many pathways. However, new tools for sharing data and code are making it easier for researchers to  
146 receive credit for doing so and to prevent others from using their data during an embargo period.

## 147 **A THREE-STEP FRAMEWORK FOR CONDUCTING** 148 **REPRODUCIBLE RESEARCH**

149 Conducting reproducible research is not exceedingly difficult, nor does it require encyclopedic knowledge of obscure  
150 research tools and protocols. Whether they know it or not, most researchers already perform much of the work  
151 required to make research reproducible. To clarify this point, we outline below some basic steps toward making  
152 research more reproducible in three stages of a research project: (1) before data analysis, (2) during analysis, and (3)  
153 after analysis. We discuss practical tips that anyone can use, as well as more advanced tools for those who would  
154 like to move beyond basic requirements (Table 1). Most readers will recognize that reproducible research largely  
155 consists of widely accepted best practices for scientific research, and that striving to meet a reasonable benchmark  
156 of reproducibility is both valuable and more attainable than researchers may think.

### 157 **Before data analysis: data storage and organization**

158 Reproducibility starts in the planning stage, with sound data management practices. It does not arise simply from  
159 sharing data and code online after a project is done. It is difficult to reproduce research when data are disorganized  
160 or missing, or when it is impossible to determine where or how data originated.

161 First, data should be backed up at every stage of the research process and stored in multiple locations. This  
162 includes raw data (e.g., physical data sheets or initial spreadsheets), clean analysis-ready data (i.e., final data sets), and  
163 steps in between. Because it is entirely possible that researchers unintentionally alter or corrupt data while cleaning  
164 it up, raw data should always be kept as a back up. Depending on the data source, it is good practice to scan and save  
165 data sheets or lab notebook pages associated with a data set to ensure that these are kept paired with the digital data



166 set. Ideally, different copies should be stored in different locations and using different storage media (e.g., hard copies  
167 *and* an external hard drive *and* cloud storage) to minimize risk of data loss from any single cause. Computers crash,  
168 hard drives are misplaced and stolen, and servers are hacked—researchers should not leave themselves vulnerable to  
169 those events.

170 Digital data files should be stored in useful, flexible, portable, non-proprietary formats. Storing data digitally in a  
171 "flat" file format is almost always a good idea. Flat file formats are those that store data as plain text with one record  
172 per line (e.g., `.csv` or `.txt` files) and are the most portable formats across platforms, as they can be opened by anyone  
173 without proprietary software programs. For more complex data types, multi-dimensional relational formats such as  
174 `json`, `hdf5`, or other discipline-specific formats (e.g., `biom` and `EML`) may be appropriate. However, the complexity  
175 of these formats makes them difficult for new researchers to access and use appropriately, so it is best to stick with  
176 simpler file formats when possible.

177 It is often useful to transform data into a 'tidy' format (Wickham, 2014) when cleaning up and standardizing raw  
178 data. Tidy data are in long format (i.e., variables in columns, observations in rows), have consistent data structure (e.g.,  
179 character data are not mixed with numeric data for a single variable), and have informative and appropriately formatted  
180 headers (e.g., reasonably short variable names that do not include problematic characters like spaces, commas, and  
181 parentheses). Data in this format are easy to manipulate, model, and visualize during analysis.

182 Metadata explaining what was done to clean up the data and what each of the variables means should be stored  
183 along with the data. Data are useless unless they can be interpreted (Roche et al., 2015); metadata is how we max-  
184 imize data interpretability across potential users. At a minimum, all data sets should include informative metadata  
185 that explains how and why data were collected, what variable names mean, whether a variable consists of raw or  
186 transformed data, and how observations are coded. Metadata should be placed in a sensible location that pairs it with  
187 the data set it describes. A few rows of metadata above a table of observations within the same file may work in some  
188 cases, or a paired text file can be included in the same directory as the data if the metadata must be more detailed. In  
189 the latter case, it is best to stick with a simple `.txt` file for metadata to maximize portability.

190 Finally, researchers should organize files in a sensible, user-friendly structure and make sure that all files have  
191 informative names. It should be easy to tell what is in a file or directory from its name, and a consistent naming protocol

192 (e.g., ending the filename with the date created or version number) provides even more information when searching  
193 through files in a directory. A consistent naming protocol for both directories and files also makes coding simpler by  
194 placing data, analyses, and products in logical locations with logical names. It is often more useful to organize files  
195 in small blocks of similar files, rather than having one large directory full of hundreds of files. For example, Noble  
196 (2009) suggests organizing computational projects within a main directory for each project, with sub-directories for  
197 the manuscript (`doc/`), data files (`data/`), analyses (`scripts/` or `src/`), and analysis products (`results/`) within that  
198 directory. While this specific organization scheme may differ for other types of research, keeping all of the research  
199 products and documentation for a given project together in this way makes it much easier to find everything at all  
200 stages of the research process, and to archive it or share it with others once the project is finished.

201       Throughout the research process, from data acquisition to publication, version control can be used to record a  
202 project's history and provide a log of changes that have occurred over the life of a project or research group. Version  
203 control systems record changes to a file or set of files over time so that you can recall specific versions later, compare  
204 differences between versions of files, and even revert files back to previous states in the event of mistakes. Many  
205 researchers use version control systems to track changes in code and documents over time. The most popular version  
206 control system is `Git`, which is often used via hosting services such as `GitHub`, `GitLab`, and `BitBucket` (Table 1). These  
207 systems are relatively easy to set up and use, and they systematically store snapshots of data, code, and accompanying  
208 files throughout the duration of a project. Version control also enables a specific snapshot of data or code to be  
209 easily shared, so that code used for analyses at a specific point in time (e.g., when a manuscript is submitted) can be  
210 documented, even if that code is later updated.

## 211 **During analysis: best coding practices**

212 When possible, all data wrangling and analysis should be performed using coding scripts—as opposed to using inter-  
213 active or point-and-click tools—so that every step is documented and repeatable by yourself and others. Code both  
214 performs operations on data and serves as a log of analytical activities. Because of this second function, code (unlike  
215 point-and-click programs) is inherently reproducible. Most errors are unintentional mistakes made during data wran-

216 gling or analysis, so having a record of these steps ensures that analyses can be checked for errors and are repeatable  
217 on future data sets. If operations are not possible to script, then they should be well-documented in a log file that is  
218 kept in the appropriate directory.

219 Analytical code should be thoroughly annotated with comments. Comments embedded within code serve as  
220 metadata for that code, substantially increasing its usefulness. Comments should contain enough information for an  
221 informed stranger to easily understand what the code does, but not so much that sorting through comments becomes  
222 a chore. Code comments can be tested by a friend who is knowledgeable about the general area of research but is  
223 not a project collaborator. In most scripting languages, the first few lines of a script should include a description of  
224 what the script does and who wrote it at the top, followed by small blocks that import data, packages, and external  
225 functions. Analytical code then follows those sections, and sections should be demarcated using a consistent protocol  
226 and sufficient comments to explain what function each section of code performs.

227 Following a clean, consistent coding style makes code easier to read. Many well-known organizations (e.g., RStu-  
228 dio, Google) offer style guidelines for software code that were developed by many expert coders. Researchers should  
229 take advantage of these while keeping in mind that all style guides are subjective to some extent. Researchers should  
230 work to develop a style that works for them. This includes using a consistent naming convention (e.g., `camelCase` or  
231 `snake_case`) to name objects and embedding meaningful information in object names (e.g., using `"_mat"` as a suffix  
232 for objects to denote matrices or `"_df"` to denote data frames). Code should also be written in relatively short lines  
233 and grouped into blocks, as our brains process narrow columns of data more easily than longer ones. Blocks of code  
234 also keep related tasks together and can function like paragraphs to make code more comprehensible.

235 There are several ways to prevent coding mistakes and make code easier to use. First, researchers should au-  
236 tomate repetitive tasks. For example, if a set of analysis steps are being used repeatedly, those steps can be saved  
237 as a function and loaded at the top of the script. This reduces the size of a script and eliminates the possibility of  
238 accidentally altering some part of a function so that it works differently in different locations within a script. Similarly,  
239 researchers can use loops to make code more efficient by performing the same task on multiple values or objects  
240 in series (though researchers should keep in mind that nesting too many loops inside one another can quickly make  
241 code incomprehensible). A third way to reduce mistakes is to reduce the number of hard-coded values that must be

242 changed to replicate analyses on a different data set. It is often best to read in the data file(s) and set parameters at the  
243 beginning of a script, so that those variable names can be used throughout the rest of the script. When operating on  
244 new data, these variables can then be changed once at the beginning of a script rather than multiple times in locations  
245 littered throughout the script.

246 Because incompatibility between operating systems or program versions can inhibit the reproducibility of re-  
247 search, the gold standard for ensuring that analyses can be used in the future is to create a software container, such  
248 as a Docker (Merkel, 2014) or Singularity (Kurtzer et al., 2017) image. Containers are lightweight, standalone,  
249 portable environments that contain the entire computing environment used in an analysis: software, all of its depen-  
250 dencies, libraries, binaries, and configuration files, all bundled into one package. Containers can then be archived or  
251 shared, allowing them to be used in the future, even as packages, functions, or libraries change over time. If creating  
252 a software container is unfeasible or a larger step than readers are willing to take, it is important to thoroughly report  
253 all software packages used, including version numbers.

## 254 **After data analysis: finalizing results and sharing**

255 After the steps above have been followed, it is time for the step most people associate with reproducible research:  
256 sharing research with others. As should be clear by now, sharing the data and code is far from the only component of  
257 reproducible research; however, once Steps 1 and 2 above are followed, it becomes the easiest step. All input data,  
258 scripts, program versions, parameters, and intermediate results should be made publicly and easily accessible. Various  
259 solutions are now available to make data sharing convenient, standardized, and accessible in a variety of research areas.  
260 There are many ways to do this, several of which are described below.

261 Just as it is better to use scripts than interactive tools in analysis, it is better to produce tables and figures directly  
262 from code than to manipulate these using Powerpoint or other image editing programs. A large number of errors in  
263 finished manuscripts come from not remembering to change *all* relevant numbers or figures when a part of an analysis  
264 changes, and this task can be incredibly time-consuming when revising a manuscript. Truly reproducible figures and  
265 tables are created directly with code and integrated into documents in a way that allows automatic updating when

266 analyses are re-run, creating a "dynamic" document. For example, documents written in `LATEX` and `markdown` incor-  
267 porate figures directly from a directory, so a figure will be updated in the document when the figure is updated in  
268 the directory (see Xie 2015 for a much lengthier discussion of dynamic documents). Both `LATEX` and `markdown` can  
269 also be used to create presentations that can incorporate live-updated figures when code or data change, so that  
270 presentations can be reproducible as well. If using one of these tools is too large a leap, then simply producing figures  
271 directly from code—instead of adding annotations and arranging panels post-hoc—can make a substantial difference  
272 in increasing the reproducibility of these products.

273 Beyond creating dynamic documents, it is possible to make data wrangling, analysis, and creation of figures, ta-  
274 bles, and manuscripts a "one-button process" using `GNU Make` (<https://www.gnu.org/software/make/>). `GNU Make` is a  
275 simple, yet powerful tool that can be used to coordinate and automate command-line processes, such as a series of  
276 independent scripts. For example, a `Makefile` can be written that will take the input data, clean and manipulate it,  
277 analyze it, produce figures and create tables with results, and update a `LATEX` or `markdown` manuscript document with  
278 those figures, tables, and any numbers included in the results. Setting up research projects to run in this way takes  
279 some time, but it can substantially expedite re-analyses and reduce copy-paste errors in manuscripts.

280 Currently, code and data that can be used to replicate research are often found in the supplementary material  
281 of journal articles. Some journals (e.g., eLife) are even experimenting with embedding data and code in articles them-  
282 selves. However, this is not a fail-safe method of archiving data and analyses: supplementary materials can be lost if  
283 a journal switches publishers or when a publisher changes its website. In addition, research is only reproducible if it  
284 can be accessed, and many papers are published in journals that are locked behind paywalls that make them inacces-  
285 sible to many researchers (Alston, 2019; Desjardins-Proulx et al., 2013; McKiernan et al., 2016). To increase access to  
286 publications, authors can post pre-prints of final (but pre-acceptance) versions of manuscripts on a pre-print server,  
287 or post-prints of manuscripts on post-print servers. There are several widely used pre-print servers (see Table 1 for  
288 three examples), and libraries at many research insitutions host post-print servers.

289 Similarly, data and code shared on personal websites are only available as long as websites are maintained, and  
290 can be difficult to transfer when researchers migrate to another domain or website provider. Materials archived on  
291 personal websites are also often difficult for other scientists to find, as they are not usually linked to the published

292 research and lack a permanent digital object identifier (DOI). To make research accessible to everyone, it is therefore  
293 better to use tools like data and code repositories than personal websites.

294 Data archiving in online repositories has become more popular in recent years, a trend resulting from a combina-  
295 tion of improvements in technology for sharing data, an increase in omics-scale data sets, and an increasing number of  
296 publisher and funding organizations who encourage or mandate data archiving (Nosek et al., 2015; Whitlock, 2011;  
297 Whitlock et al., 2010). Data repositories are large databases that collect, manage, and store data sets for analysis,  
298 sharing, and reporting. Repositories may be either subject- or data-specific, or cross-disciplinary general repositories  
299 that accept multiple data types. Some are free and others require a fee for depositing data. Journals often recom-  
300 mend appropriate repositories on their websites, and these recommendations should be consulted when submitting  
301 a manuscript. Three commonly used general purpose repositories are Dryad, Zenodo, and Figshare; each of these  
302 creates a DOI that allows data and code to be citable by others. Before choosing a repository, researchers should  
303 explore commonly used options in their specific fields of research.

304 When data, code, software, and products of a research project are archived together, these are termed a "research  
305 compendium" (Gentleman and Lang, 2007). Research compendia are increasingly common, although standards for  
306 what is included differ between scientific fields. They provide a standardized and easily recognisable way to organize  
307 the digital materials of a research project, which enables other researchers to inspect, reproduce, and extend research  
308 (Marwick et al., 2018).

309 In particular, the Open Science Framework (OSF; <http://osf.io/>) is a project management repository that goes  
310 beyond the repository features of Dryad/Zenodo/Figshare to integrate and share components of a research project  
311 using collaborative tools. The goal of the OSF is to enable research to be shared at every step of the scientific process—  
312 from developing a research idea and designing a study, to storing and analyzing collected data and writing and publish-  
313 ing reports or papers (Sullivan et al., 2019). OSF is integrated with many other reproducible research tools, including  
314 widely used pre-print servers, version control software, and publishers.

## 315 CONCLUSIONS

316 While many researchers associate reproducible research primarily with a set of advanced tools for sharing research,  
317 reproducibility is just as much about simple work habits as the tools used to share data and code. We ourselves  
318 are not perfect reproducible researchers—we do not use all the tools mentioned in this commentary all the time and  
319 often fail to follow our own advice. Nevertheless, we recognize that reproducible research is a process rather than a  
320 destination and work hard to consistently increase the reproducibility of our work. We encourage others to do the  
321 same. Researchers can make strides toward a more reproducible research process by simply thinking carefully about  
322 data management and organization, coding practices, and processes for making figures and tables (e.g. Table 2). Time  
323 and expertise must be invested in learning and adopting these tools and tips, and this investment can be substantial.  
324 Nevertheless, we encourage our fellow researchers to work toward more open and reproducible research practices  
325 so we can all enjoy the resulting improvements in work habits, collaboration, scientific rigor, and trust in science.

## 326 ACKNOWLEDGEMENTS

327 Many thanks to J.G. Harrison and B.J. Rick for providing helpful comments on an early version of this manuscript and  
328 to C.A. Buerkle for inspiring this project during his Computational Biology course at the University of Wyoming.

## 329 DATA ACCESSIBILITY

330 There was no data or code used in this manuscript. Resources for getting started with reproducible work flows can  
331 be found at <http://www.github.com/jessicarick/resources>.

## 332 **AUTHORS' CONTRIBUTIONS**

333 Both authors contributed equally to this project. JA conceived the idea and both authors jointly wrote the manuscript,  
334 contributed critically to drafts, and gave final approval for publication.

## 335 **REFERENCES**

- 336 Alston, J. M. (2019) Open access principles and practices benefit conservation. *Conservation Letters*, **12**, e12672.
- 337 Bohannon, J. (2015) Many psychology papers fail replication test. *Science*, **349**, 910–911.
- 338 Cokol, M., Ozbay, F. and Rodriguez-Esteban, R. (2008) Retraction rates are on the rise. *EMBO reports*, **9**, 2–2.
- 339 Crocker, J. and Cooper, M. L. (2011) Addressing scientific fraud. *Science*, **334**, 1182–1182.
- 340 Culina, A., Crowther, T. W., Ramakers, J. J. C., Gienapp, P. and Visser, M. E. (2018) How to do meta-analysis of open datasets.  
341 *Nature Ecology & Evolution*, **2**, 1053–1056.
- 342 Desjardins-Proulx, P., White, E. P., Adamson, J. J., Ram, K., Poisot, T. and Gravel, D. (2013) The case for open preprints in  
343 biology. *PLOS Biology*, **11**, e1001563.
- 344 Gentleman, R. and Lang, D. T. (2007) Statistical analyses and reproducible research. *Journal of Computational and Graphical*  
345 *Statistics*, **16**, 1–23.
- 346 Gerstner, K., Moreno-Mateos, D., Gurevitch, J., Beckmann, M., Kambach, S., Jones, H. P. and Seppelt, R. (2017) Will your  
347 paper be used in a meta-analysis? Make the reach of your research broader and longer lasting. *Methods in Ecology and*  
348 *Evolution*, **8**, 777–784. \_eprint: <https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210X.12758>.
- 349 Haddaway, N. R. and Verhoeven, J. T. A. (2015) Poor methodological detail precludes experimental repeatability and hampers  
350 synthesis in ecology. *Ecology and Evolution*, **5**, 4451–4454.
- 351 Hewitt, J. K. (2012) Editorial policy on candidate gene association and candidate gene-by-environment interaction studies of  
352 complex traits. *Behavior Genetics*, **42**, 1–2.
- 353 Ioannidis, J. P. A. (2005) Why most published research findings are false. *PLOS Medicine*, **2**, e124.



- 354 Kurtzer, G. M., Sochat, V. and Bauer, M. W. (2017) Singularity: scientific containers for mobility of compute. *PLOS ONE*, **12**,  
355 e0177459.
- 356 Laine, C., Goodman, S. N., Griswold, M. E. and Sox, H. C. (2007) Reproducible research: moving toward research the public  
357 can really trust. *Annals of Internal Medicine*, **146**, 450.
- 358 Marwick, B., Boettiger, C. and Mullen, L. (2018) Packaging data analytical work reproducibly using R (and friends). *The American*  
359 *Statistician*, **72**, 80–88.
- 360 McKiernan, E. C., Bourne, P. E., Brown, C. T., Buck, S., Kenall, A., Lin, J., McDougall, D., Nosek, B. A., Ram, K., Soderberg, C. K.,  
361 Spies, J. R., Thaney, K., Updegrave, A., Woo, K. H. and Yarkoni, T. (2016) How open science helps researchers succeed.  
362 *eLife*, **5**, e16800.
- 363 Merkel, D. (2014) Docker: lightweight Linux containers for consistent development and deployment. *Linux Journal*, **2014**, 2:2.
- 364 Mislán, K. a. S., Heer, J. M. and White, E. P. (2016) Elevating the status of code in ecology. *Trends in Ecology & Evolution*, **31**,  
365 4–7.
- 366 Moonesinghe, R., Khoury, M. J. and Janssens, A. C. J. W. (2007) Most published research findings are false—but a little repli-  
367 cation goes a long way. *PLOS Medicine*, **4**, e28.
- 368 Noble, W. S. (2009) A quick guide to organizing computational biology projects. *PLOS Computational Biology*, **5**, e1000424.
- 369 Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., Buck, S., Chambers, C. D., Chin, G., Chris-  
370 tensen, G., Contestabile, M., Dafoe, A., Eich, E., Freese, J., Glennerster, R., Goroff, D., Green, D. P., Hesse, B., Humphreys,  
371 M., Ishiyama, J., Karlan, D., Kraut, A., Lupia, A., Mabry, P., Madon, T., Malhotra, N., Mayo-Wilson, E., McNutt, M., Miguel,  
372 E., Paluck, E. L., Simonsohn, U., Soderberg, C., Spellman, B. A., Turitto, J., VandenBos, G., Vazire, S., Wagenmakers, E. J.,  
373 Wilson, R. and Yarkoni, T. (2015) Promoting an open research culture. *Science*, **348**, 1422–1425.
- 374 Open Science Collaboration (2015) Estimating the reproducibility of psychological science. *Science*, **349**, aac4716.
- 375 Parr, C. S. and Cummings, M. P. (2005) Data sharing in ecology and evolution. *Trends in Ecology & Evolution*, **20**, 362–363.
- 376 Peng, R. D. (2011) Reproducible research in computational science. *Science*, **334**, 1226–1227.
- 377 Piwowar, H. A., Day, R. S. and Fridsma, D. B. (2007) Sharing detailed research data is associated with increased citation rate.  
378 *PLOS ONE*, **2**, e308.

- 379 Roche, D. G., Kruuk, L. E. B., Lanfear, R. and Binning, S. A. (2015) Public data archiving in ecology and evolution: how well are  
380 we doing? *PLoS Biology*, **13**, e1002295.
- 381 Sandve, G. K., Nekrutenko, A., Taylor, J. and Hovig, E. (2013) Ten simple rules for reproducible computational research. *PLOS*  
382 *Computational Biology*, **9**, e1003285.
- 383 Schooler, J. W. (2014) Metascience could rescue the 'replication crisis'. *Nature*, **515**, 9–9.
- 384 Steen, R. G., Casadevall, A. and Fang, F. C. (2013) Why has the number of scientific retractions increased? *PLOS ONE*, **8**,  
385 e68397.
- 386 Stewart, G. (2010) Meta-analysis in applied ecology. *Biology Letters*, **6**, 78–81.
- 387 Stodden, V., Guo, P. and Ma, Z. (2013) Toward reproducible computational research: an empirical analysis of data and code  
388 policy adoption by journals. *PLOS ONE*, **8**, e67111.
- 389 Sullivan, I., DeHaven, A. and Mellor, D. (2019) Open and Reproducible Research on Open Science Framework. *Current Protocols*  
390 *Essential Laboratory Techniques*, **18**, e32.
- 391 Whitlock, M., McPeck, M., Rausher, M., Rieseberg, L. and Moore, A. (2010) Data archiving. *The American Naturalist*, **175**,  
392 145–146.
- 393 Whitlock, M. C. (2011) Data archiving in ecology and evolution: best practices. *Trends in Ecology & Evolution*, **26**, 61–65.  
394 Publisher: Elsevier.
- 395 Wickham, H. (2014) Tidy data. *Journal of Statistical Software*, **059**.
- 396 Xie, Y. (2015) *Dynamic Documents with R and knitr*. CRC Press.

**397 TABLES**

**TABLE 1** A list of advanced tools commonly used for reproducible research, aggregated by function. This list is not intended to be comprehensive, but should serve as a good starting point for those interested in moving beyond basic requirements.

	Free	Open Source	Website
<b>Data and Code Management</b>			
Version control			
GitHub	Y*	N	<a href="https://github.com">https://github.com</a>
BitBucket	Y*	N	<a href="https://bitbucket.com">https://bitbucket.com</a>
GitLab	Y*	Y	<a href="https://www.gitlab.com">https://www.gitlab.com</a>
Make			
GNU Make	Y	Y	<a href="https://www.gnu.org/software/make/">https://www.gnu.org/software/make/</a>
Software containers and virtual machines			
Docker	Y	Y	<a href="https://docker.com">https://docker.com</a>
Singularity	Y*	Y	<a href="https://syslabs.io">https://syslabs.io</a>
Oracle VM VirtualBox	Y	Y	<a href="https://virtualbox.org">https://virtualbox.org</a>
<b>Sharing Research</b>			
Preprint Servers			
ArXiv	Y		<a href="https://arxiv.org/">https://arxiv.org/</a>
bioRxiv	Y		<a href="https://www.biorxiv.org/">https://www.biorxiv.org/</a>
EcoEvoRxiv	Y		<a href="https://ecoevorxiv.org/">https://ecoevorxiv.org/</a>
Manuscript creation			
Overleaf	Y*	Y	<a href="https://overleaf.com">https://overleaf.com</a>
TeXstudio	Y	Y	<a href="https://www.texstudio.org/">https://www.texstudio.org/</a>
Rstudio	Y	Y	<a href="https://rstudio.org">https://rstudio.org</a>
Data Repositories			
Dryad	N		<a href="https://datadryad.org/">https://datadryad.org/</a>
Figshare	Y*		<a href="https://figshare.com/">https://figshare.com/</a>
Zenodo	Y		<a href="https://zenodo.org/">https://zenodo.org/</a>
Open Science Framework	Y		<a href="https://osf.io/">https://osf.io/</a>

\* free to use, but paid premium options with more features are available

**TABLE 2** Ten questions to ask before sharing data and code publicly.

- 
1. Are raw data safely stored in multiple locations using multiple media?
  2. Are final data stored in a portable, non-proprietary format?
  3. Are final data formatted appropriately for analysis?
  4. Are data paired with adequate metadata?
  5. Is code clean, readable, and appropriately formatted?
  6. Is code thoroughly commented?
  7. Have data and code been reviewed by at least one collaborator or friend?
  8. Have all software versions and computing environments been documented?
  9. Are explicit instructions on locating data, metadata, and code detailed in the manuscript?
  10. Will data, metadata, and code be shared together at a permanent site?
-