A Beginner's Guide to Conducting Reproducible Research

Jesse M. Alston[1,2] and Jessica A. Rick[1,3]

[1]Program in Ecology, University of Wyoming, Laramie, WY USA

[2]Department of Zoology and Physiology, University of Wyoming, Laramie, WY USA

[3]Department of Botany, University of Wyoming, Laramie, WY USA

Mailing address: 1000 E. University Dr., Laramie, WY 82072 USA

Email addresses: jalston@uwyo.edu, jrick@uwyo.edu

Abstract

Reproducible research is widely acknowledged as an important tool for improving science and reducing harm from the "replication crisis", yet research in most fields within biology remains largely irreproducible. In this article, we make the case for why all research should be reproducible, explain why research is often not reproducible, and offer a simple framework that researchers can use to make their research more reproducible. Researchers can increase the reproducibility of their work by improving data management practices, writing more readable code, and increasing use of the many available platforms for sharing data and code. While reproducible research is often associated with a set of advanced tools for sharing data and code, reproducibility is just as much about maintaining work habits that are already widely acknowledged as best practices for research. Increasing reproducibility will increase rigor, trustworthiness, and transparency while benefiting both practitioners of reproducible research and their fellow researchers.

Keywords: data management, data repository, software, open science, replication

Introduction

Replication is a fundamental tenet of science, but there is increasing fear among scientists that too few scientific studies can be replicated. This has been termed the "replication crisis" (Ioannidis 2005; Schooler 2014). Scientific papers often include inadequate detail to enable reproduction (Haddaway and Verhoeven 2015; Archmiller et al. 2020), many attempted replications of well-known scientific studies have failed in a wide variety of disciplines (Bohannon 2015; Hewitt 2012; Moonesinghe et al. 2007; Open Science Collaboration 2015), and rates of paper retractions are increasing (Cokol et al. 2008; Steen et al. 2013). Because of this, researchers are working to develop new ways for researchers, research institutions, research funders, and journals to overcome this problem (Peng 2011; Sandve et al. 2013; Stodden et al. 2013; Fiedler et al. 2012).

Because replicating studies with new independent data is expensive, rarely published in high-

25 impact journals, and sometimes even methodologically impossible, "reproducible research" is

26 often suggested as a method for increasing our ability to assess the validity and rigor of scientific

27 results (Peng 2011). Research is reproducible when others can reproduce scientific results given

28 only the original data, code, and documentation (Essawy et al. 2020). This commentary describes

29 basic requirements for such reproducibility in biological research. In it, we make the case for why

30 all research should be reproducible, explain why research is often not reproducible, and present

31 a simple three-part framework all researchers can use to make their research more reproducible.

32 These principles are applicable to researchers working in all types of biological research with data

33 sets of all sizes and levels of complexity.

34 Why Do Reproducible Research?

35 Reproducible research is a by-product of careful attention to detail throughout the research pro-

36 cess, and allows researchers to ensure that they can repeat the same analysis multiple times with

37 the same results, at any point in that process. Because of this, researchers who conduct repro-

38 ducible research are the primary beneficiaries of this practice. In addition, reproducible research

39 benefits others in the scientific community. Sharing data, code, and detailed research methods

40 and results leads to faster progress in methodological development and innovation because re-

41 search is more accessible to more scientists (Mislan et al. 2016; Parr and Cummings 2005; Roche

42 et al. 2015). In this way, everyone involved benefits from research being conducted reproducibly.

43 Reproducible research benefits those who do it

44 Reproducible research helps researchers remember how and why they performed specific analyses

45 during the course of a project. This enables easier explanation of work to collaborators, supervi-

46 sors, and reviewers, and it allows collaborators to conduct supplementary analyses more quickly

47 and more efficiently.

48 Second, reproducible research enables researchers to quickly and simply alter analyses and

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

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

Fourth, conducting reproducible research is a strong indicator to fellow researchers of rigor, trustworthiness, and transparency in scientific research. This can increase the quality and speed of peer review, because reviewers can directly access the analytical process described in a manuscript. Peer reviewers' work becomes easier and they may be able to answer methodological questions without asking the authors. It also protects researchers from accusations of research misconduct due to analytical errors, because it is unlikely that researchers would openly share fraudulent code and data with the rest of the research community. In addition, reviewers can check whether code matches with methods described in the text of a manuscript, to make sure that authors correctly performed the analyses as described. Finally, it increases the probability that errors are caught during the peer-review process, decreasing the likelihood of corrections or retractions after publication.

Finally, reproducible research increases paper citation rates (Piwowar et al. 2007; McKiernan et al. 2016) and allows other researchers to cite code and data in addition to publications. This enables a given research project to have more impact than it would if the data or methods were hidden from the public. For example, researchers can re-use code from a paper with similar methods and organize their data in the same manner as the original paper, then cite code from the original paper in their manuscript. Another researcher may conduct a meta-analysis on the phenomenon described in the two research papers, and thus use and cite both the two papers

76   and the data from those papers in their meta-analysis. Papers are more likely to be cited in these

77   re-use cases if full information about data and analyses are available (Whitlock 2011; Culina et al.

78   2018).

Reproducible research benefits the research community

80   Reproducible research allows others to learn from your work. Scientific research has a steep

81   learning curve, and allowing others to access data and code gives them a head start on per-

82   forming similar analyses. For example, junior researchers can use code shared with the research

83   community by more senior researchers to learn how to perform advanced analyses. This allows

84   junior researchers to conduct research that is more rigorous from the outset, rather than having

85   to spend months or years trying to figure out "best practices" through trial and error. Modifying

86   existing resources can also save time and effort for experienced researchers—even experienced

87   coders can modify existing code much faster than they can write code from scratch. Sharing

88   code thus allows experienced researchers to perform similar analyses more quickly.

89   Second, reproducible research allows others to understand and reproduce a researcher's work.

90   Allowing others to access data and code makes it easier for other scientists to perform follow-up

91   studies to increase the strength of evidence for the phenomenon of interest. It also increases

92   the likelihood that similar studies are compatible with one another, and that all of these studies

93   can provide evidence in support of or in opposition to a concept. In addition, sharing data and

94   code increases the utility of these studies for meta-analyses that are important for generalizing

95   and contextualizing the findings of studies on a topic. Meta-analyses in ecology and evolutionary

96   biology are often hindered by incompatibility of data between studies, or lack of documentation for

97   how those data were obtained (Stewart 2010; Culina et al. 2018). Well-documented, reproducible

98   findings enhance the likelihood that data can be used in future meta-analyses (Gerstner et al.

99   2017).

100   Third, reproducible research allows others to protect themselves from your mistakes. Mistakes

101   happen in science. Allowing others to access data and code gives them a better chance to critically

102 analyze the work, which can lead to coauthors or reviewers discovering mistakes during the revision

103 process, or other scientists discovering mistakes after publication. This prevents mistakes from

104 compounding over time and provides protection for collaborators, research institutions, funding

105 organizations, journals, and others who may be affected when such mistakes happen.

Barriers to Reproducible Research

107 There are a number of reasons that most research is not reproducible. Rapidly developing tech-

108 nologies and analytical tools, novel interdisciplinary approaches, unique ecological study systems,

109 and increasingly complex data sets and research questions hinder reproducibility, as does pressure

110 on scientists to publish novel research quickly. This multitude of barriers can be simplified into

111 four primary themes: (1) complexity, (2) technological change, (3) human error, and (4) concerns

112 over intellectual property rights. Each of these concerns can contribute to making research less

113 reproducible and can be valid in some scenarios. However, each of these factors can also be ad-

114 dressed easily via well-developed tools, protocols, and institutional norms concerning reproducible

115 research.

116 **Complexity.** — Science is difficult, and scientific research requires specialized (and often pro-

117 prietary) knowledge and tools that may not be available to everyone who would like to reproduce

118 research. For example, analyses of genomic data require researchers to possess a vast base of

119 knowledge about statistical methodologies and the molecular architecture of DNA, and genomic

120 analyses are therefore difficult to reproduce for those with limited knowledge of the subject. Some

121 analyses may require high-performance computing clusters that use several different programming

122 languages and software packages, or that are designed for specific hardware configurations. Other

123 analyses may be performed using proprietary software programs such as SAS statistical software

124 (SAS Institute Inc., Cary, NC, USA) or ArcGIS (Esri, Redlands, CA, USA) that require expensive

125 software licenses. Lack of knowledge, lack of institutional infrastructure, and lack of funding all

126 make research less reproducible. However, most of these issues can be mitigated fairly easily.

127 Researchers can cite primers on complex subjects or analyses to reduce knowledge barriers. They

128 can also thoroughly annotate analytical code with comments explaining each step in an analysis,

129 or provide extensive documentation on research software. Using open software (when possible)

130 makes research more accessible for other researchers as well.

131 **Technological change.** — Hardware and software both change over time, and they often

132 change quickly. When old tools become obsolete, research becomes less reproducible. For exam-

133 ple, reproducing research performed in 1960 using that era's computational tools would require a

134 completely new set of tools today. Even research performed just a few years ago may have been

135 conducted using software that is no longer available or is incompatible with other software that

136 has since been updated. One minor update in a piece of software used in one minor analysis in

137 an analytical workflow can render an entire project less reproducible. However, this too can be

138 mitigated by using established tools in reproducible research. Careful documentation of versions

139 of software used in analyses is a baseline requirement that anyone can meet. There are also more

140 advanced tools that can help overcome such challenges in making research reproducible, including

141 software containers, which are described in further detail below.

142 **Human error.** — Though fraudulent research is often cited as reason to make research

143 more reproducible (e.g., Ioannidis 2005; Laine et al. 2007; Crocker and Cooper 2011), many

144 more innocent reasons exist as to why research is often difficult to reproduce (e.g., Elliott 2014).

145 People forget small details of how they performed analyses. They fail to describe data collection

146 protocols or analyses completely despite their best efforts and multiple reviewers checking their

147 work. They perform sloppy analyses because they just want to be done with a project that feels

148 like it is taking forever to complete. Science is performed by fallible humans, and a wide variety

149 of common events can render research less reproducible.

150 While not all of these challenges can be avoided by performing research reproducibly, a well-

151 documented research process can guard against small errors and sloppy analyses. For example,

152 carefully recording details such as when and where data were collected, what decisions were made

153 during data collection, and what labeling conventions were used can make a huge difference in

making sure that those data can later be used appropriately or re-purposed. Unintentional errors often occur during the data wrangling stage of a project, and these can be mitigated by keeping multiple copies of data to prevent data loss, carefully documenting the process for converting raw data into clean data, and double-checking a small test set of data before manipulating the data set as a whole.

Intellectual property rights. — Researchers often hesitate to share data and code because doing so may allow other researchers to use data and code incorrectly or unethically. Other researchers may use publicly available data without notifying authors, leading to incorrect assumptions about the data that result in invalid analyses. Researchers may use publicly available data or code without citing the original data owners or code writers, who then do not receive proper credit for gathering expensive data or writing time-consuming code. Researchers may want to conceal data from others so that they can perform new analyses on those data in the future without worrying about others scooping them using the shared data. Rational self-interest can lead to hesitation to share data and code via many pathways. However, new tools for sharing data and code are making it easier for researchers to receive credit for doing so and to prevent others from using their data during an embargo period.

A Three-Step Framework for Conducting Reproducible Research

Conducting reproducible research is not exceedingly difficult, nor does it require encyclopedic knowledge of esoteric research tools and protocols. Whether they know it or not, most researchers already perform much of the work required to make research reproducible. To clarify this point, we outline below some basic steps toward making research more reproducible in three stages of a research project: (1) before data analysis, (2) during analysis, and (3) after analysis. We discuss practical tips that anyone can use, as well as more advanced tools for those who would like to move beyond basic requirements (Table 1). Most readers will recognize that reproducible

179 research largely consists of widely accepted best practices for scientific research, and that striving
180 to meet a reasonable benchmark of reproducibility is both more valuable and more attainable
181 than researchers may think.

Before data analysis: data storage and organization

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

187   First, data should be backed up at every stage of the research process and stored in multiple
188 locations. This includes raw data (e.g., physical data sheets or initial spreadsheets), clean analysis-
189 ready data (i.e., final data sets), and steps in between. Because it is entirely possible that
190 researchers unintentionally alter or corrupt data while cleaning it up, raw data should always
191 be kept as a back up. It is good practice to scan and save data sheets or lab notebook pages
192 associated with a data set to ensure that these are kept paired with the digital data set. Ideally,
193 different copies should be stored in different locations and using different storage media (e.g.,
194 paper copies *and* an external hard drive *and* cloud storage) to minimize risk of data loss from any
195 single cause. Computers crash, hard drives are misplaced and stolen, and servers are hacked—
196 researchers should not leave themselves vulnerable to those events.

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

205   It is often useful to transform data into a 'tidy' format (Wickham 2014) when cleaning up and

206 standardizing raw data. Tidy data are in long format (i.e., variables in columns, observations in

207 rows), have consistent data structure (e.g., character data are not mixed with numeric data for a

208 single variable), and have informative and appropriately formatted headers (e.g., reasonably short

209 variable names that do not include problematic characters like spaces, commas, and parentheses).

210 Data in this format are easy to manipulate, model, and visualize during analysis.

211   Metadata explaining what was done to clean up the data and what each of the variables

212 means should be stored along with the data. Data are useless unless they can be interpreted

213 (Roche et al. 2015); metadata is how we maximize data interpretability across potential users.

214 At a minimum, all data sets should include informative metadata that explains how and why data

215 were collected, what variable names mean, whether a variable consists of raw or transformed

216 data, and how observations are coded. Metadata should be placed in a sensible location that

217 pairs it with the data set it describes. A few rows of metadata above a table of observations

218 within the same file may work in some cases, or a paired text file can be included in the same

219 directory as the data if the metadata must be more detailed. In the latter case, it is best to stick

220 with a simple `.txt` file for metadata to maximize portability.

221   Finally, researchers should organize files in a sensible, user-friendly structure and make sure

222 that all files have informative names. It should be easy to tell what is in a file or directory from

223 its name, and a consistent naming protocol (e.g., ending the filename with the date created or

224 version number) provides even more information when searching through files in a directory. A

225 consistent naming protocol for both directories and files also makes coding simpler by placing

226 data, analyses, and products in logical locations with logical names. It is often more useful to

227 organize files in small blocks of similar files, rather than having one large directory full of hundreds

228 of files. For example, Noble (2009) suggests organizing computational projects within a main

229 directory for each project, with sub-directories for the manuscript (`doc/`), data files (`data/`),

230 analyses (`scripts/` or `src/`), and analysis products (`results/`) within that directory. While

231 this specific organization scheme may differ for other types of research, keeping all of the research

232 products and documentation for a given project organized in this way makes it much easier to

233 find everything at all stages of the research process, and to archive it or share it with others once

234 the project is finished.

235   Throughout the research process, from data acquisition to publication, version control can

236 be used to record a project's history and provide a log of changes that have occurred over the

237 life of a project or research group. Version control systems record changes to a file or set of files

238 over time so that you can recall specific versions later, compare differences between versions of

239 files, and even revert files back to previous states in the event of mistakes. Many researchers

240 use version control systems to track changes in code and documents over time. The most

241 popular version control system is `Git`, which is often used via hosting services such as `GitHub`,

242 `GitLab`, and `BitBucket` (Table 1). These systems are relatively easy to set up and use, and they

243 systematically store snapshots of data, code, and accompanying files throughout the duration of

244 a project. Version control also enables a specific snapshot of data or code to be easily shared,

245 so that code used for analyses at a specific point in time (e.g., when a manuscript is submitted)

246 can be documented, even if that code is later updated.

During analysis: best coding practices

248 When possible, all data wrangling and analysis should be performed using coding scripts—as

249 opposed to using interactive or point-and-click tools—so that every step is documented and

250 repeatable by yourself and others. Code both performs operations on data and serves as a log

251 of analytical activities. Because of this second function, code (unlike point-and-click programs)

252 is inherently reproducible. Most errors are unintentional mistakes made during data wrangling or

253 analysis, so having a record of these steps ensures that analyses can be checked for errors and

254 are repeatable on future data sets. If operations are not possible to script, then they should be

255 well-documented in a log file that is kept in the appropriate directory.

256   Analytical code should be thoroughly annotated with comments. Comments embedded within

257 code serve as metadata for that code, substantially increasing its usefulness. Comments should

258 contain enough information for an informed stranger to easily understand what the code does,

259 but not so much that sorting through comments is a chore. Code comments can be tested for

260 this balance by a friend who is knowledgeable about the general area of research but is not a

261 project collaborator. In most scripting languages, the first few lines of a script should include a

262 description of what the script does and who wrote it, followed by small blocks that import data,

263 packages, and external functions. Data cleaning and analytical code then follows those sections,

264 and sections are demarcated using a consistent protocol and sufficient comments to explain what

265 function each section of code performs.

266   Following a clean, consistent coding style makes code easier to read. Many well-known

267 organizations (e.g., RStudio, Google) offer style guidelines for software code that were developed

268 by many expert coders. Researchers should take advantage of these while keeping in mind that all

269 style guides are subjective to some extent. Researchers should work to develop a style that works

270 for them. This includes using a consistent naming convention (e.g., `camelCase` or `snake_case`)

271 to name objects and embedding meaningful information in object names (e.g., using "`_mat`" as

272 a suffix for objects to denote matrices or "`_df`" to denote data frames). Code should also be

273 written in relatively short lines and grouped into blocks, as our brains process narrow columns of

274 data more easily than longer ones (Martin 2009). Blocks of code also keep related tasks together

275 and can function like paragraphs to make code more comprehensible.

276   There are several ways to prevent coding mistakes and make code easier to use. First,

277 researchers should automate repetitive tasks. For example, if a set of analysis steps are being

278 used repeatedly, those steps can be saved as a function and loaded at the top of the script. This

279 reduces the size of a script and eliminates the possibility of accidentally altering some part of a

280 function so that it works differently in different locations within a script. Similarly, researchers can

281 use loops to make code more efficient by performing the same task on multiple values or objects

282 in series (though it is also important to note that nesting too many loops inside one another can

283 quickly make code incomprehensible). A third way to reduce mistakes is to reduce the number of

284 hard-coded values that must be changed to replicate analyses on an updated or new data set. It

285 is often best to read in the data file(s) and assign parameter values at the beginning of a script,

286 so that those variables can then be used throughout the rest of the script. When operating on

287 new data, these variables can then be changed once at the beginning of a script rather than

288 multiple times in locations littered throughout the script.

289 Because incompatibility between operating systems or program versions can inhibit the re-

290 producibility of research, the current gold standard for ensuring that analyses can be used in

291 the future is to create a software container, such as a `Docker` (Merkel 2014) or `Singularity`

292 (Kurtzer et al. 2017) image. Containers are lightweight, standalone, portable environments that

293 contain the entire computing environment used in an analysis: software, all of its dependencies,

294 libraries, binaries, and configuration files, all bundled into one package. Containers can then be

295 archived or shared, allowing them to be used in the future, even as packages, functions, or libraries

296 change over time. If creating a software container is infeasible or a larger step than readers are

297 willing to take, it is important to thoroughly report all software packages used, including version

298 numbers.

After data analysis: finalizing results and sharing

300 After the steps above have been followed, it is time for the step most people associate with

301 reproducible research: sharing research with others. As should be clear by now, sharing the data

302 and code is far from the only component of reproducible research; however, once Steps 1 and

303 2 above are followed, it becomes the easiest step. All input data, scripts, program versions,

304 parameters, and important intermediate results should be made publicly and easily accessible.

305 Various solutions are now available to make data sharing convenient, standardized, and accessible

306 in a variety of research areas. There are many ways to do this, several of which are described

307 below.

308 Just as it is better to use scripts than interactive tools in analysis, it is better to produce

309 tables and figures directly from code than to manipulate these using Adobe Illustrator, Microsoft

310 Powerpoint, or other image editing programs. A large number of errors in finished manuscripts

311 come from not remembering to change *all* relevant numbers or figures when a part of an analysis

312 changes, and this task can be incredibly time-consuming when revising a manuscript. Truly repro-

313 ducible figures and tables are created directly with code and integrated into documents in a way

314 that allows automatic updating when analyses are re-run, creating a "dynamic" document. For

315 example, documents written in LaTeX and `markdown` incorporate figures directly from a directory,

316 so a figure will be updated in the document when the figure is updated in the directory (see

317 Xie 2015 for a much lengthier discussion of dynamic documents). Both LaTeX and `markdown`

318 can also be used to create presentations that can incorporate live-updated figures when code or

319 data change, so that presentations can be reproducible as well. If using one of these tools is too

320 large a leap, then simply producing figures directly from code—instead of adding annotations and

321 arranging panels post-hoc—can make a substantial difference in increasing the reproducibility of

322 these products.

323 Beyond creating dynamic documents, it is possible to make data wrangling, analysis, and

324 creation of figures, tables, and manuscripts a "one-button" process using GNU Make (`https:`

325 `//www.gnu.org/software/make/`). GNU Make is a simple, yet powerful tool that can be used

326 to coordinate and automate command-line processes, such as a series of independent scripts. For

327 example, a `Makefile` can be written that will take the input data, clean and manipulate it, analyze

328 it, produce figures and tables with results, and update a LaTeX or `markdown` manuscript document

329 with those figures, tables, and any numbers included in the results. Setting up research projects

330 to run in this way takes some time, but it can substantially expedite re-analyses and reduce

331 copy-paste errors in manuscripts.

332 Currently, code and data that can be used to replicate research are often found in the sup-

333 plementary material of journal articles. Some journals (e.g., *eLife*) are even experimenting with

334 embedding data and code in articles themselves. However, this is not a fail-safe method of

335 archiving data and analyses: supplementary materials can be lost if a journal switches publish-

336 ers or when a publisher changes its website. In addition, research is only reproducible if it can

337 be accessed, and many papers are published in journals that are locked behind paywalls that

make them inaccessible to many researchers (Desjardins-Proulx et al. 2013; McKiernan et al. 2016; Alston 2019). To increase access to publications, authors can post pre-prints of final (but pre-acceptance) versions of manuscripts on a pre-print server, or post-prints of manuscripts on post-print servers. There are several widely used pre-print servers (see Table 1 for three examples), and libraries at many research institutions host post-print servers.

Similarly, data and code shared on personal websites are only available as long as websites are maintained, and can be difficult to transfer when researchers migrate to another domain or website provider. Materials archived on personal websites are also often difficult for other scientists to find, as they are not usually linked to the published research and lack a permanent digital object identifier (DOI). To make research accessible to everyone, it is therefore better to use tools like data and code repositories than personal websites.

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

When data, code, software, and products of a research project are archived together, these are termed a "research compendium" (Gentleman and Lang 2007). Research compendia are increasingly common, although standards for what is included in research compendia differ between scientific fields. They provide a standardized and easily recognisable way to organize the digital

365 materials of a research project, which enables other researchers to inspect, reproduce, and extend

366 research (Marwick et al. 2018).

367 In particular, the Open Science Framework (OSF; `http://osf.io/`) is a project management

368 repository that goes beyond the repository features of `Dryad`, `Zenodo`, and `Figshare` to integrate

369 and share components of a research project using collaborative tools. The goal of the OSF is to

370 enable research to be shared at every step of the scientific process—from developing a research

371 idea and designing a study, to storing and analyzing collected data and writing and publishing

372 reports or papers (Sullivan et al. 2019). OSF is integrated with many other reproducible research

373 tools, including widely used pre-print servers, version control software, and publishers.

Conclusions

375 While many researchers associate reproducible research primarily with a set of advanced tools

376 for sharing research, reproducibility is just as much about simple work habits as the tools used

377 to share data and code. We ourselves are not perfect reproducible researchers—we do not use

378 all the tools mentioned in this commentary all the time and often fail to follow our own advice

379 (almost always to our regret). Nevertheless, we recognize that reproducible research is a process

380 rather than a destination and work hard to consistently increase the reproducibility of our work.

381 We encourage others to do the same. Researchers can make strides toward a more reproducible

382 research process by simply thinking carefully about data management and organization, coding

383 practices, and processes for making figures and tables (e.g., Fig. 1). Time and expertise must

384 be invested in learning and adopting these tools and tips, and this investment can be substantial.

385 Nevertheless, we encourage our fellow researchers to work toward more open and reproducible

386 research practices so we can all enjoy the resulting improvements in work habits, collaboration,

387 scientific rigor, and trust in science.

Acknowledgements

Data Accessibility

There was no data or code used in this manuscript, but some resources for getting started with reproducible work flows can be found at http://www.github.com/jessicarick/resources.

Authors' Contributions

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

Author biographical

Jesse Alston is a Ph.D. candidate in the Program in Ecology and Department of Zoology and Physiology at the University of Wyoming. Jessica Rick is a Ph.D. candidate in the Program in Ecology and Department of Botany at the University of Wyoming.

References

Alston JM. 2019. Open access principles and practices benefit conservation. Conservation Letters 12: e12672.

Archmiller AA, Johnson AD, Nolan J, Edwards M, Elliott LH, Ferguson JM, Iannarilli F, Vélez J,

406   Vitense K et al. 2020. Computational reproducibility in The Wildlife Society's flagship journals.

407   The Journal of Wildlife Management n/a. Publisher: John Wiley & Sons, Ltd.

408   Bohannon J. 2015. Many psychology papers fail replication test. Science 349: 910–911.

409   Cokol M, Ozbay F and Rodriguez-Esteban R. 2008. Retraction rates are on the rise. EMBO

410   reports 9: 2–2.

411   Crocker J and Cooper ML. 2011. Addressing scientific fraud. Science 334: 1182–1182.

412   Culina A, Crowther TW, Ramakers JJC, Gienapp P and Visser ME. 2018. How to do meta-analysis

413   of open datasets. Nature Ecology & Evolution 2: 1053–1056.

414   Desjardins-Proulx P, White EP, Adamson JJ, Ram K, Poisot T and Gravel D. 2013. The case for

415   open preprints in biology. PLOS Biology 11: e1001563.

416   Elliott DB. 2014. The impact factor: a useful indicator of journal quality or fatally flawed?

417   Ophthalmic and Physiological Optics 34: 4–7.

418   Essawy BT, Goodall JL, Voce D, Morsy MM, Sadler JM, Choi YD, Tarboton DG and Malik

419   T. 2020. A taxonomy for reproducible and replicable research in environmental modelling.

420   Environmental Modelling & Software page 104753.

421   Fiedler AK, Landis DA and Arduser M. 2012. Rapid shift in pollinator communities following

422   invasive species removal. Restoration Ecology 20: 593–602.

423   Gentleman R and Lang DT. 2007. Statistical analyses and reproducible research. Journal of

424   Computational and Graphical Statistics 16: 1–23.

425   Gerstner K, Moreno-Mateos D, Gurevitch J, Beckmann M, Kambach S, Jones HP and Seppelt

426   R. 2017. Will your paper be used in a meta-analysis? Make the reach of your research broader

427   and longer lasting. Methods in Ecology and Evolution 8: 777–784.

428 Haddaway NR and Verhoeven JTA. 2015. Poor methodological detail precludes experimental
429   repeatability and hampers synthesis in ecology. Ecology and Evolution 5: 4451–4454.

430 Hewitt JK. 2012. Editorial policy on candidate gene association and candidate gene-by-
431   environment interaction studies of complex traits. Behavior Genetics 42: 1–2.

432 Ioannidis JPA. 2005. Why most published research findings are false. PLOS Medicine 2: e124.

433 Kurtzer GM, Sochat V and Bauer MW. 2017. Singularity: scientific containers for mobility of
434   compute. PLOS ONE 12: e0177459.

435 Laine C, Goodman SN, Griswold ME and Sox HC. 2007. Reproducible research: moving toward
436   research the public can really trust. Annals of Internal Medicine 146: 450.

437 Martin RC. 2009. Clean code: a handbook of agile software craftsmanship. Prentice Hall, Upper
438   Saddle River, NJ, USA.

439 Marwick B, Boettiger C and Mullen L. 2018. Packaging data analytical work reproducibly using
440   R (and friends). The American Statistician 72: 80–88.

441 McKiernan EC, Bourne PE, Brown CT, Buck S, Kenall A, Lin J, McDougall D, Nosek BA, Ram
442   K et al. 2016. How open science helps researchers succeed. eLife 5: e16800.

443 Merkel D. 2014. Docker: lightweight Linux containers for consistent development and deployment.
444   Linux Journal 2014: 2:2.

445 Mislan KAS, Heer JM and White EP. 2016. Elevating the status of code in ecology. Trends in
446   Ecology & Evolution 31: 4–7.

447 Moonesinghe R, Khoury MJ and Janssens ACJW. 2007. Most published research findings are
448   false—but a little replication goes a long way. PLOS Medicine 4: e28.

449 Noble WS. 2009. A quick guide to organizing computational biology projects. PLOS Computa-
450   tional Biology 5: e1000424.

451 Nosek BA, Alter G, Banks GC, Borsboom D, Bowman SD, Breckler SJ, Buck S, Chambers CD,
452 Chin G et al. 2015. Promoting an open research culture. Science 348: 1422–1425.

453 Open Science Collaboration. 2015. Estimating the reproducibility of psychological science. Science
454 349: aac4716.

455 Parr CS and Cummings MP. 2005. Data sharing in ecology and evolution. Trends in Ecology &
456 Evolution 20: 362–363.

457 Peng RD. 2011. Reproducible research in computational science. Science 334: 1226–1227.

458 Piwowar HA, Day RS and Fridsma DB. 2007. Sharing detailed research data is associated with
459 increased citation rate. PLOS ONE 2: e308.

460 Roche DG, Kruuk LEB, Lanfear R and Binning SA. 2015. Public data archiving in ecology and
461 evolution: how well are we doing? PLoS Biology 13: e1002295.

462 Sandve GK, Nekrutenko A, Taylor J and Hovig E. 2013. Ten simple rules for reproducible
463 computational research. PLOS Computational Biology 9: e1003285.

464 Schooler JW. 2014. Metascience could rescue the 'replication crisis'. Nature 515: 9–9.

465 Steen RG, Casadevall A and Fang FC. 2013. Why has the number of scientific retractions
466 increased? PLOS ONE 8: e68397.

467 Stewart G. 2010. Meta-analysis in applied ecology. Biology Letters 6: 78–81.

468 Stodden V, Guo P and Ma Z. 2013. Toward reproducible computational research: an empirical
469 analysis of data and code policy adoption by journals. PLOS ONE 8: e67111.

470 Sullivan I, DeHaven A and Mellor D. 2019. Open and reproducible research on Open Science
471 Framework. Current Protocols Essential Laboratory Techniques 18: e32.

472 Whitlock M, McPeek M, Rausher M, Rieseberg L and Moore A. 2010. Data archiving. The
473 American Naturalist 175: 145–146.

474 Whitlock MC. 2011. Data archiving in ecology and evolution: best practices. Trends in Ecology

475   & Evolution 26: 61–65.

476 Wickham H. 2014. Tidy data. Journal of Statistical Software 59.

477 Xie Y. 2015. Dynamic Documents with R and knitr. CRC Press.

478 # 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
Data and Code Management			
Version control			
GitHub	Y[a]	N	https://github.com
BitBucket	Y[a]	N	https://bitbucket.com
GitLab	Y[a]	Y	https://www.gitlab.com
Make			
GNU Make	Y	Y	https://www.gnu.org/software/make/
Software containers and virtual machines			
Docker	Y	Y	https://docker.com
Singularity	Y[a]	Y	https://syslabs.io
Oracle VM VirtualBox	Y	Y	https://virtualbox.org
Sharing Research			
Preprint Servers			
ArXiv	Y		https://arxiv.org/
bioRxiv	Y		https://www.biorxiv.org/
EcoEvoRxiv	Y		https://ecoevorxiv.org/
Manuscript creation			
Overleaf	Y[a]	Y	https://overleaf.com
TeXstudio	Y	Y	https://www.texstudio.org/
Rstudio	Y	Y	https://rstudio.org
Data Repositories			
Dryad	N		https://datadryad.org/
Figshare	Y[a]		https://figshare.com/
Zenodo	Y		https://zenodo.org/
Open Science Framework	Y		https://osf.io/

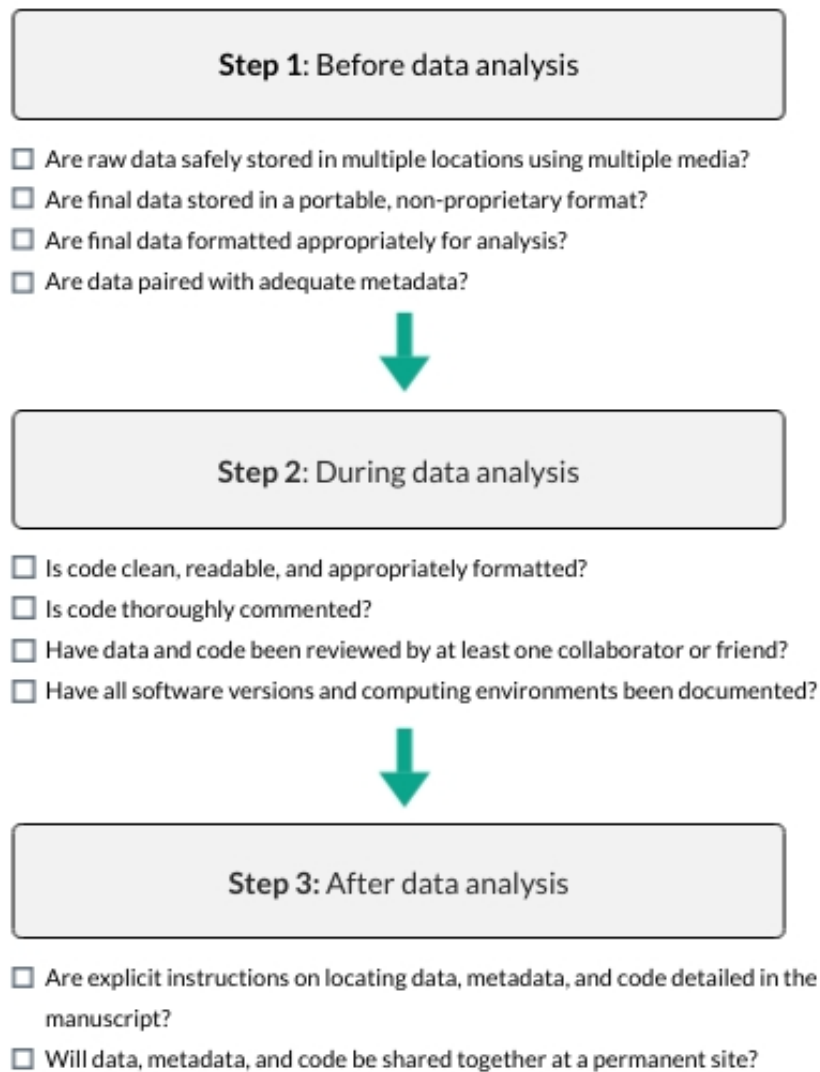[a] free to use, but paid premium options with more features are available

479 # Figures

Figure 1: A ten-point checklist to guide researchers toward greater reproducibility in their research. Researchers should give careful thought before, during, and after analysis to ensure reproducibility of their work.