Open collaborative writing with Manubot

This manuscript ([permalink](#)) was automatically generated from [greenelab/meta-review@6c53684](#) on April 5, 2019.

Authors

- **Daniel S. Himmelstein**
  0000-0002-3012-7446 · dhimmel · dhimmel
  Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania

- **David R. Slochower**
  0000-0003-3928-5050 · slochower · drslochower
  Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California, San Diego

- **Venkat S. Malladi**
  0000-0002-0144-0564 · vsmalladi · katatonikkat
  Department of Bioinformatics, University of Texas Southwestern Medical Center; Bioinformatics Core Facility, University of Texas Southwestern Medical Center

- **Dongbo Hu**
  0000-0003-2606-3969 · dongbohu · dongbohu
  Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania

- **Vincent Rubinetti**
  0000-0002-4655-3773 · vincerubinetti · vincerubinetti
  Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania

- **Casey S. Greene**
  0000-0001-8713-9213 · cgreene · GreeneScientist
  Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania

- **Anthony Gitter**
  0000-0002-5324-9833 · agitter · anthonygitter
  Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison; Morgridge Institute for Research

Abstract

Open, collaborative research is a powerful paradigm that can immensely strengthen the scientific process by integrating broad and diverse expertise. However, traditional research and multi-author writing processes break down at scale. New tools and workflows that rely on automation can catch errors and ensure that contributions are fairly tracked in massively collaborative research. We present techniques for overcoming challenges of open research, with special emphasis on manuscript writing. These include approaches for managing distributed authors and our new software, named Manubot, for automating citation and many other aspects of manuscript building.

Introduction

The internet enables science to be shared in real-time at a low cost to a global audience. This development has decreased the barriers to making science open, while supporting new massively collaborative models of research. However, the scientific community requires tools whose workflows encourage openness. Manuscripts are the cornerstone of scholarly communication, but drafting and publishing manuscripts has traditionally relied on proprietary or offline tools that do not support *open scholarly writing*, in which anyone is able to contribute and the contribution history is preserved and public. We introduce Manubot, a new tool and infrastructure for authoring scholarly manuscripts in the open, and report how it was instrumental for the collaborative project that led to its creation.

Based on our experience leading a recent open review [1], we discuss the advantages and challenges of open collaborative writing, a form of crowdsourcing [2]. Our review manuscript [3] was code-named the Deep Review and surveyed deep learning's role in biology and precision medicine, a research area undergoing explosive growth. We initiated the Deep Review in August 2016 by creating a GitHub repository (https://github.com/greenelab/deep-review) to coordinate and manage contributions. GitHub is a platform designed for collaborative software development that is adaptable for collaborative writing. From the start, we made the GitHub repository public under a Creative Commons Attribution 4.0 International (CC BY 4.0) license. We encouraged anyone interested to contribute by proposing changes or additions. Although we invited some specific experts to participate, most authors discovered the manuscript organically through conferences or social media, deciding to contribute without solicitation. In total, the Deep Review attracted 36 authors, who were not determined in advance, from 20 different institutions in less than two years.

The Deep Review and other studies that subsequently adopted the Manubot platform were unequivocal successes bolstered by the collaborative approach. However, inviting wide authorship brought many technical and social challenges such as how to fairly distribute credit, coordinate the scientific content, and collaboratively manage extensive reference lists. The manuscript writing process we developed using the Markdown language, the GitHub platform, and our new Manubot tool for automating manuscript generation addresses these challenges.

Manubot contribution workflow

There are many existing collaborative writing platforms. Some provide "what you see is what you get" (WYSIWYG) interfaces and support Microsoft Word or similar document formats. Others are LaTeX-based systems for technical writing [4], such as Overleaf and Authorea. These tools ideally offer version control, multiple permission levels, or other functionality to support multi-author document editing. Existing platforms work well for editing text and are widely used for scholarly writing. However, they lack features that are important for open writing, such as managing a collaborative manuscript and attributing precise credit among many authors (Table 1). None of these platforms offer the ability to address thematically related changes together and enable multiple authors to iteratively refine proposed changes.

Table 1: Collaborative writing platforms. A summary of features that differentiate Manubot from existing collaborative writing platforms. We assessed features on June 15, 2018 using the free version of each platform and updated our assessment on April 3, 2019 to add Overleaf v2 and the features in the bottom three rows. Some platforms offer additional features through a paid subscription or software.
1) Additional functionality, such as bibliography management and tracking changes, is available by editing the Word document stored in OneDrive with the paid Word desktop application. 2) Conversations about modifications take place on the document as comments, annotations, or unsaved chats. There is no integrated forum for discussing and editing

revisions. 3) In some circumstances, Overleaf Git commits are not modular. Edits made by distinct authors may be attributed to a single author. The GitHub Sync feature attributes all edits to the project owner.

Feature	Manubot	Authorea + BibTeX	Overleaf v2 + BibTeX	Google Docs + Paperpile	Word Online[1]	Markdown on GitHub
Multi-author editing	Yes	Yes	Yes	Yes	Yes	Yes
Propose changes	Yes	No	No	Yes	No	Yes
Continuous integration testing	Yes	No	No	No	No	No
Multi-participant conversation for changes	Yes	No[2]	No[2]	No[2]	No[2]	Yes
Character-level provenance for text	Yes	No (versions tracked by day)	No[3]	Requires manual inspection of history	Not after changes are accepted	Yes
Bibliography management	Yes	Yes	Yes	Yes	No, requires the Word desktop application	No
Citation by identifier	Yes	No	No	No	No	No
Editing software	Any text editor	Web interface	Web interface	Web interface	Web interface	Any text editor
Document format	Markdown	LaTeX	LaTeX	Proprietary	Proprietary	Markdown
Templating	Yes	Yes	Yes	No	No	No
Technical expertise required	Yes	Yes	Yes	No	No	Yes
WYSIWYG mode	No	Rich text available	Rich text available	Yes	Yes	Preview rendered Markdown
Inline comments	Yes using Hypothesis	Yes	Yes	Yes	Yes	No
Viewing changes	Diff of manuscript source	Highlight changes	Compare labeled versions	Highlight changes	No	Diff of manuscript source

Manubot's collaborative writing workflow adopts standard software development strategies that enable any contributor to edit any part of the manuscript but enforce discussion and review of all proposed changes. The GitHub platform supports organizing and editing the manuscript. Manubot projects use GitHub *issues* for organization, opening a new issue for each discussion topic. For example, in a review manuscript like the Deep Review, this includes each primary paper under consideration. Within a paper's issue, contributors summarize the research, discuss it (sometimes with participation from the original authors), and assess its relevance to the review. In a primary

research article, issues can instead track progress on specific figures or subsections of text being drafted. Issues serve as an open to-do list and a forum for debating the main messages of the manuscript.

GitHub and the underlying Git version control system [5,6] also structure the writing process. The official version of the manuscript is *forked* by individual contributors. A contributor then adds and revises files, grouping these changes into *commits*. When the changes are ready to be reviewed, the series of commits are submitted as a *pull request* through GitHub, which notifies other authors of the pending changes. GitHub's review interface allows anyone to comment on the changes, globally or at specific lines, asking questions or requesting modifications, as described by Johnny Israeli [7]. Conversations during review can reference other pull requests, issues, or authors, linking the relevant people and content, as illustrated in Figure 1. Reviewing batches of revisions that focus on a single theme is more efficient than independently discussing isolated comments and edits and helps maintain consistent content and tone across different authors and reviewers. Once all requested modifications are made, the manuscript maintainers, a subset of authors with elevated GitHub permissions, formally approve the pull request and merge the changes into the official version. The process of writing and revising material can be orchestrated through GitHub with a web browser or a local text editor.
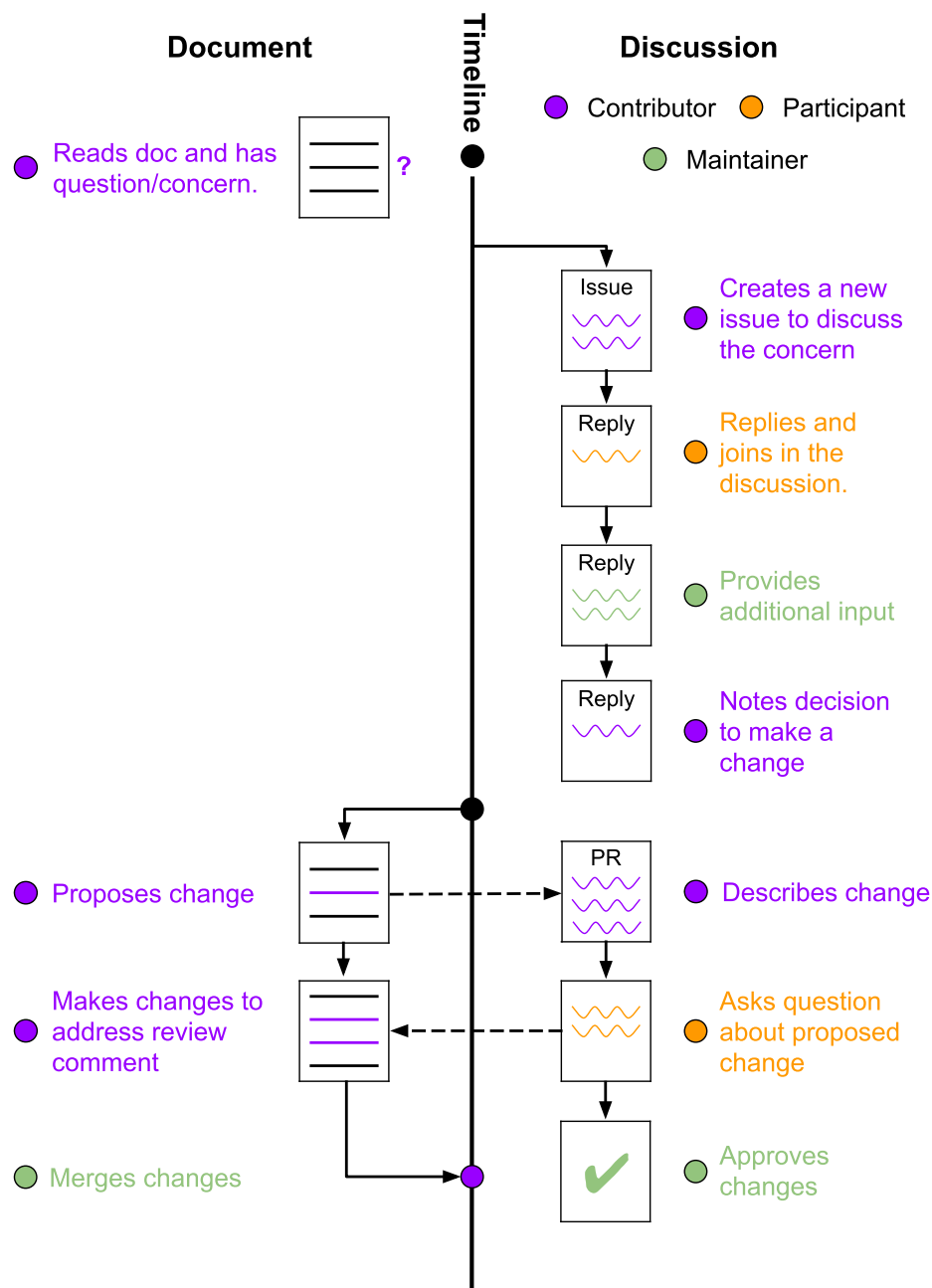
Figure 1: Manubot editing workflow. Any reader can contribute to a Manubot manuscript by proposing a change through a pull request. This example involves three people: a manuscript Maintainer, an existing project Contributor, and an additional Participant in the discussion. Manuscript text is shown in solid lines on the left of the timeline and discussion on GitHub is shown by squiggly lines to the right of the timeline. The Contributor opens a GitHub issue to discuss a manuscript modification. The Maintainer and the Participant provide feedback in the issue, and the Maintainer recommends creating a GitHub pull request to update the text. The Contributor creates the pull request. It is reviewed by the Maintainer and the Participant, and the Contributor updates the pull request in response. Once the pull request is approved, the Maintainer merges the changes into the official version of the manuscript.

The Deep Review issue and pull request on protein-protein interactions demonstrate this process in practice. A new contributor identified a relevant research topic that was missing from the review manuscript with examples of how the literature would be summarized, critiqued, and integrated into the review. A maintainer confirmed that this was a desirable topic and referred to related open issues. The contributor made the pull request, and two maintainers and another participant made recommendations. After four rounds of reviews and pull request edits, a maintainer merged the changes.

We found that this workflow was an effective compromise between fully unrestricted editing and a more heavily-structured approach that limited the authors or the sections they could edit. In addition, authors are associated with their commits, which makes it easy for contributors to receive credit for their work. Figure [2] and the GitHub contributors page summarize all edits and commits from each author, providing aggregated information that is not available on other collaborative writing platforms. Because the Manubot writing process tracks the complete history through Git commits, it enables detailed retrospective contribution analysis. These pull request and contribution tracking examples both come from Deep Review, the largest Manubot project to date, but illustrate the general principles of transparency and collaboration that are shared by all open Manubot manuscripts.
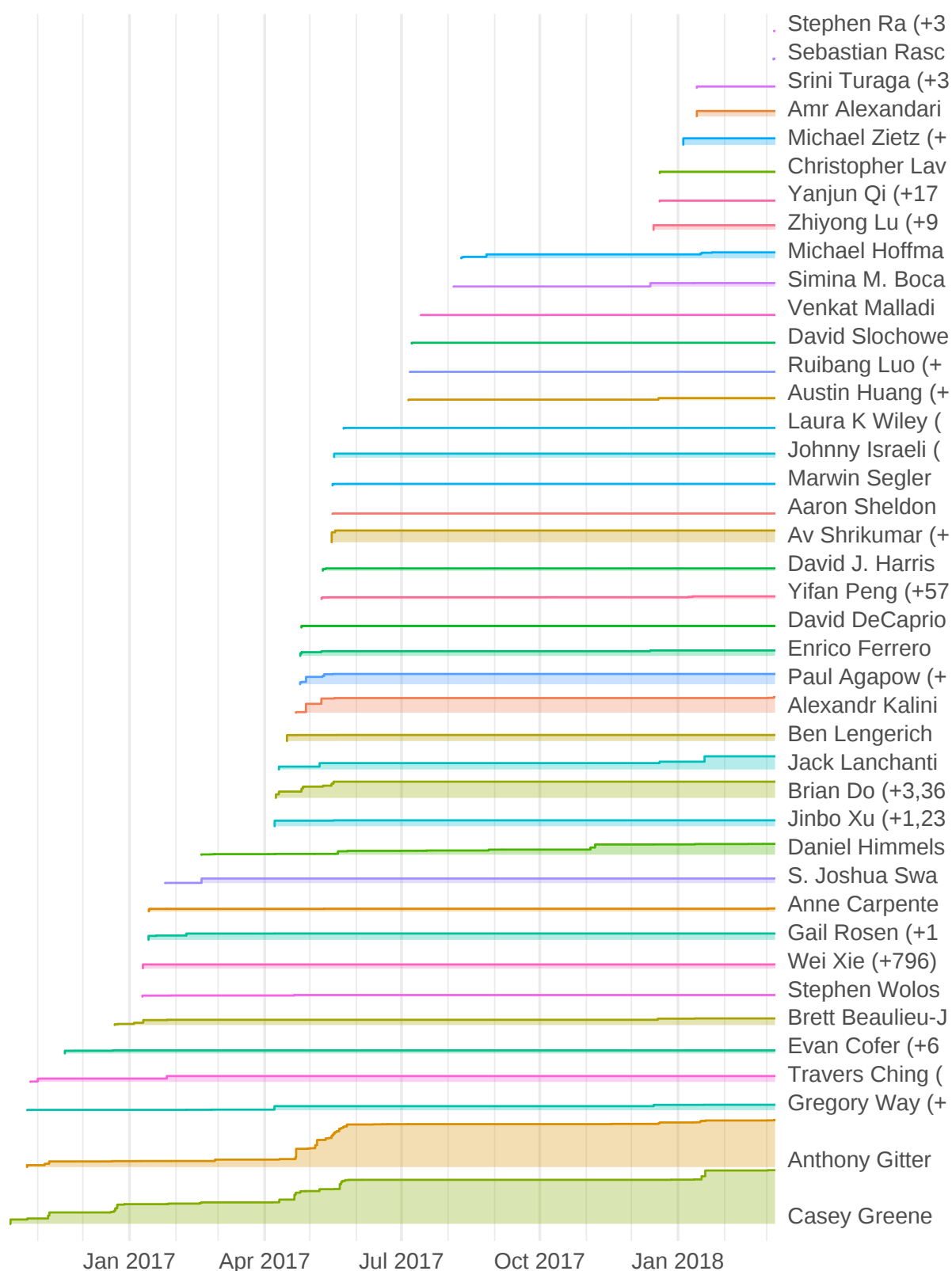
Figure 2: Deep Review contributions by author over time. The total words added to the Deep Review by each author is plotted over time (final values in parentheses). These statistics were extracted from Git commit diffs of the manuscript's Markdown source. This figure reveals the composition of written contributions to the manuscript at every point in its history. The Deep Review was initiated in August 2016, and the first complete manuscript was released as a preprint [8] in May 2017. While the article was under review, we continued to maintain the project and accepted new contributions. The preprint was updated in January 2018, and the article was accepted by the journal in March 2018 [3]. As of March 06, 2019, the Deep Review repository accumulated 755 Git commits, 317 merged pull requests, 609 issues, and 819 GitHub stars. The notebook to generate this figure can be interactively launched using Binder [9], enabling users to explore alternative visualizations or analyses of the source data.

Manubot features

Manubot is a system for writing scholarly manuscripts via GitHub. For each manuscript, there is a corresponding Git repository. The `master` branch of the repository contains all of the necessary inputs to build the manuscript. Specifically, a `content` directory contains one or more Markdown files that define the body of the manuscript as well as a metadata file to set information such as the title, authors, keywords, and language. Figures can be hosted in the `content/images` subdirectory or elsewhere and specified by URL. Repositories contain scripts and other assets that define how to build and deploy the manuscript. Many of these operations are delegated to the `manubot` [Python package](#) or other dependencies such as Pandoc, which converts between document formats, and Travis CI, which builds the manuscript in the cloud. Manubot pieces together many existing standards and technologies to encapsulate a manuscript in a repository and automatically generate outputs.

Markdown

With Manubot, manuscripts are written as plain-text Markdown files. The [Markdown standard](#) itself provides limited yet crucial formatting syntax, including the ability to embed images and format text via bold, italics, hyperlinks, headers, inline code, codeblocks, blockquotes, and numbered or bulleted lists. In addition, Manubot relies on extensions from [Pandoc Markdown](#) to enable citations, tables, captions, and equations specified using the popular TeX math syntax. Markdown with Pandoc extensions supports most formatting options required for scholarly writing [10], but currently lacks the ability to cross-reference and automatically number figures, tables, and equations. For this functionality, Manubot includes the `pandoc-xnos` suite of Pandoc filters. A list of formatting options officially supported by Manubot, at the time of writing, is viewable as [raw markdown](#) and the corresponding [rendered HTML](#).

By virtue of its readable markup syntax, Markdown is well suited for version control using Git. Markdown treats a single line break between text as a space, and requires two-or-more consecutive line breaks to denote a new paragraph. For optimal tracking of Markdown files with Git, we recommend placing each sentence on its own line. This convention allows Git to display diffs on a per sentence basis while avoiding unnecessary reflows associated with line wrapping and allowing easy rearrangement of sentences.

Citation by identifier

Manubot includes an additional layer of citation processing, currently unique to the system. All citations point to a standard identifier, for which Manubot automatically retrieves bibliographic metadata. Table [2](#) presents the supported identifiers and example citations before and after Manubot processing. Authors can optionally define citation tags to provide short readable alternatives to the citation identifiers. Metadata is exported to [Citation Style Language (CSL) JSON Data Items](#), an open standard that is widely supported by reference managers [11,12]. However, sometimes external resources provide Manubot with invalid CSL Data, which can cause errors with downstream citation processors, such as [pandoc-citeproc](#). Therefore, Manubot removes invalid fields according to the [CSL Data specification](#). In cases where automatic retrieval of metadata fails or produces incorrect references — which is most common for URL citations — users can manually provide the correct CSL JSON. Manual CSL JSON also supports references without standard identifiers, such as print-only newspaper articles.

Table 2: Citation types supported by Manubot. Manubot allows users to cite different types of persistent identifiers, as shown in this table. Metadata source indicates the primary resource used to retrieve bibliographic metadata. For certain identifier types, additional metadata sources are queried should the primary fail. For example, when translation-server ISBN lookup fails, Manubot tries Wikipedia's Citoid service followed by the isbnlib Python package. When translation-server URL lookup fails, Manubot then tries Greycite [13]. Raw citations enable citing works when no supported persistent identifiers exist, but require that the user specifies the metadata. Finally, authors may optionally map a named tag to any of the supported identifier types. In this example, the tag `avasthi-preprints` represents the DOI identifier `doi:10.7554/eLife.38532`.

Identifier	Metadata source	Example citation	Processed citation
Digital Object Identifier (DOI)	DOI Content Negotiation	`doi:10.1098/rsif.2017.0387`	[3]
shortDOI	DOI Proxy Server REST API	`doi:10/gddkhn`	[3]
PubMed Identifier (PMID)	NCBI E-utilities	`pmid:25851694`	[14]
PubMed Central Identifier (PMCID)	NCBI Literature Citation Exporter	`pmcid:PMC4719068`	[2]
arXiv ID	arXiv API	`arxiv:1502.04015v1`	[15]
International Standard Book Number (ISBN)	Zotero translation-server	`isbn:9780262517638`	[16]
Web address (URL)	Zotero translation-server	`url:https://lgatto.github.io/open-and-open/`	[17]
Wikidata ID	Zotero translation-server	`wikidata:Q56458321`	[18]
Raw	Provided by user	`raw:dongbo-conversation`	[19]
Tag	Source for tagged identifier	`tag:avasthi-preprints`	[20]

Manubot formats bibliographies according to a CSL style specification. Styles define how references are constructed from bibliographic metadata, controlling layout details such as the maximum number of authors to list per reference. Manubot's default style emphasizes titles and electronic (rather than print) identifiers and applies numeric-style citations [21]. Alternatively, users can also choose from thousands of predefined styles or build their own [22]. As a result, adopting the specific bibliographic format required by a journal usually just requires specifying the style's source URL in the Manubot configuration.

Format conversion

Manubot uses Pandoc to convert manuscripts from Markdown to HTML, PDF, and optionally DOCX outputs. Pandoc supports conversion between additional formats — such as LaTeX, AsciiDoc, EPUB, and JATS — offering Manubot users broad interoperability. Journal Article Tag Suite (JATS) is a standard XML format for scholarly articles that is used by publishers, archives, and text miners [23,24, 25]. Pandoc's JATS support provides an avenue to integrate Manubot with the larger JATS ecosystem. In the future, journals may begin accepting submissions in JATS. But for now, Manubot's DOCX output is usually sufficient for journal submissions that require an editable source document. Otherwise, authors generally use the PDF output (e.g. for preprint or initial journal submissions). The primary Manubot output is HTML intended to be viewed in a web browser. Accordingly, manuscripts natively support JavaScript and can thus include any web-based interactive visualization, such as those produced using Vega-Lite, Bokeh, or Plotly [26,27].

Interactive features and appearance

Manubot comes with several "plugins" that can be included in manuscripts exported as HTML. These plugins add special interactive features that enhance the user experience of viewing and reading manuscripts (Figure 3). For example, with the "tooltips" plugin enabled, when the user hovers over a link to a reference or figure, a preview of that item pops up above the link, along with controls to navigate between other mentions of that item elsewhere in the document. The build process can also accommodate different "themes", which change the general aesthetics and appearance of the exported document (for example, from a contemporary sans-serif style to a more traditional serif style). The architecture of the plugins and themes is designed to provide authors with enough flexibility to suit their particular needs and preferences.
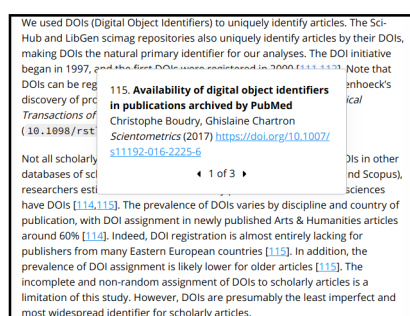
Citation tooltips
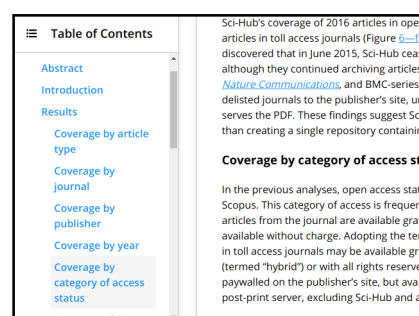
Table of contents
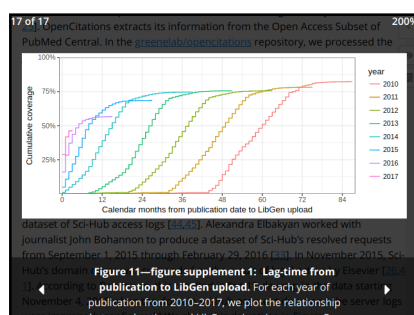
Image viewer (lightbox)
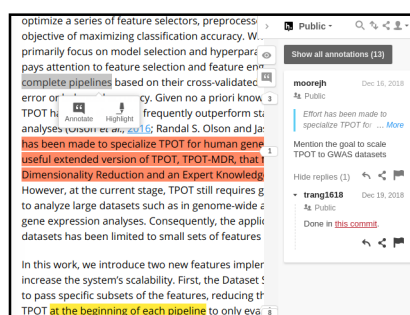
Annotations and highlights (Hypothesis)
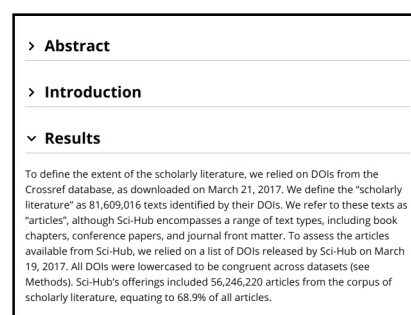
Collapsible sections
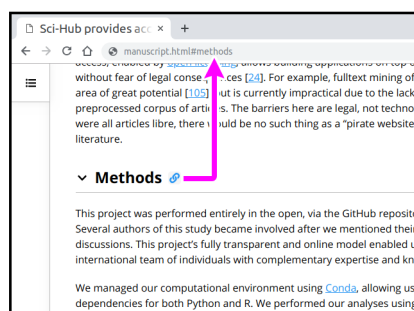
Section/figure/table anchors (permalinks)
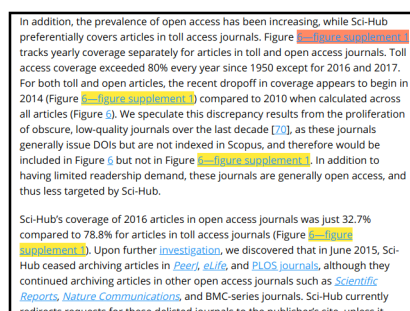
Same citation highlighting

Jump to citation

Figure 3: Examples of the various Manubot plugins, illustrating their functionality and usefulness. Screenshots were taken from existing manuscripts made with Manubot: Sci-Hub Coverage Study and TPOT-DS, available under the CC BY 4.0 License. Clarifying markups are overlaid in purple.

The Manubot "front-end" (layout, look, controls, behavior, etc.) was developed in line with current best practices and user expectations of the modern web. The plugins utilize standard technology built in to most major web browsers, allowing them to be relatively lightweight, modular, and easy to configure.

Continuous publication

Manubot performs continuous publication: every update to a manuscript's source is automatically reflected in the online outputs. The approach uses continuous integration (CI) [28,29,30], specifically via Travis CI, to monitor changes. When changes occur, the CI service attempts to generate an updated manuscript. If this process is error free, the CI service timestamps the manuscript and uploads the output files to the GitHub repository. Because the HTML manuscript is hosted using GitHub Pages, the CI service automatically deploys the new manuscript version when it pushes the updated outputs to GitHub. Using CI to build the manuscript automatically catches many common errors, such as misspelled citations, invalid formatting, or misconfigured software dependencies.

To illustrate, the source GitHub repository for this article is https://github.com/greenelab/meta-review. When this repository changes, Travis CI rebuilds the manuscript. If successful, the output is deployed back to GitHub (to dedicated `output` and `gh-pages` branches). As a result, https://greenelab.github.io/meta-review stays up to date with the latest HTML manuscript. Furthermore, versioned URLs, such as https://greenelab.github.io/meta-review/v/4b6396bcefd1b9c7ddf39c1d3f0b3eab2dd63f31/, provide access to previous manuscript versions.

Timestamping

The idea of the "priority of discovery" is important to science, and Vale and Hyman discuss the importance of both disclosure and validation [31]. In their framework, disclosure occurs when a scientific output is released to the world. However, for a manuscript that is shared as it is written, being able to establish priority could be challenging. Manubot supports OpenTimestamps to timestamp the HTML and PDF outputs on the Bitcoin blockchain. This procedure allows one to retrospectively prove that a manuscript version existed prior to its blockchain-verifiable timestamp [15,32,33,34,35]. Timestamps protect against attempts to rewrite a manuscript's history and ensure accurate histories, potentially alleviating certain authorship or priority disputes. Because all bitcoin transactions compete for limited space on the blockchain, the fees required to send a single transaction can be high. OpenTimestamps minimizes fees by encoding many timestamps into a single Bitcoin transaction, enabling the service to be free of charge [36]. Since transactions can take up to a few days to be made, Manubot initially stores incomplete timestamps and upgrades them in future continuous deployment builds. We find that this asynchronous design with timestamps precise to the day is suitable for the purposes of scientific writing.

Reproducible manuscripts

Manubot and its dependencies are free of charge and largely open source. It does rely on gratis services from two proprietary platforms: GitHub and Travis CI. Fortunately, lock-in to these services is minimal, and several substitutes already exist. Manubot provides a substantial step towards end-to-end document reproducibility, where every figure or piece of data in a manuscript can be traced back to its origin [37] and is well suited for preserving provenance. For example, figures can be specified using versioned URLs that refer to the code that created them. In addition, manuscripts can be templated, so that numerical values or tables are inserted directly from the repository that created them.

Getting started

An example repository at https://github.com/manubot/rootstock, referred to as Rootstock, demonstrates Manubot's features and serves as a template for users to write their own manuscripts

with Manubot. The [current setup process](#) includes cloning the Rootstock repository, rebranding it to the user's manuscript, and configuring continuous integration. The setup process is complex, but must only be performed once per manuscript. Incorporating new Manubot features is also possible by pulling the latest commits from Rootstock, but sometimes involves resolving Git conflicts.

Contributing to an existing manuscript is less technical and can be performed entirely through GitHub's web interface, as discussed in the [Contribution Workflow](#) section and demonstrated in Video S1. Interested readers can practice editing an existing demo manuscript at https://github.com/manubot/try-manubot.

At the 2019 *Pacific Symposium on Biocomputing*, we led a working group where 17 conference participants contributed to a [demo manuscript](#). Based on this experience, we believe most computational scholars have the expertise to contribute to a Manubot manuscript. Proficiency with Manubot requires familiarity with Markdown, Git, GitHub, and continuous integration. While these tools do present a barrier to entry, they are also highly applicable outside of Manubot and increasingly part of the standard curriculum for computational scholars. For example, Markdown is used for documenting Jupyter and R Markdown notebooks.

Existing manuscripts

Since its creation to facilitate the Deep Review, Manubot has been used to write a variety of scholarly documents. The [Sci-Hub Coverage Study](#) — performed openly on GitHub from its inception — investigated Sci-Hub's repository of pirated articles [38]. Sci-Hub [reviewed](#) the initial preprint from this study in a series of tweets, pointing out a major error in one of the analyses. Within hours, the authors used Markdown's strikethrough formatting in Manubot to cross-out the errant sentences ([commit](#), [versioned manuscript](#)), thereby alerting readers to the mistake and preventing further propagation of misinformation. One month later, a larger [set of revisions](#) explained the error in more detail and was included in a second version of the preprint. As such, continuous publishing via Manubot helped the authors address the error without delay, while retaining a public version history of the process. This Sci-Hub Coverage Study preprint was the [most viewed](#) 2017 *PeerJ Preprint*, while the Deep Review was the most viewed 2017 *bioRxiv* preprint [39]. Hence, in Manubot's first year, two of the most popular preprints were written using its collaborative, open, and review-driven authoring process.

Additional research studies in progress are being authored using Manubot, spanning the fields of [regulatory genomics](#) [40], [synthetic biology](#) [41], [climate science](#), [machine learning](#), and [data visualization](#). Manubot is also being used for documents beyond traditional journal publications, such as [research tips](#), [quality standards](#) [42], [grant proposals](#), [progress reports](#), [undergraduate research reports](#) [43], [literature reviews](#), and lab notebooks. Finally, manuscripts written with other authoring systems have been successfully ported to Manubot, including the [Bitcoin Whitepaper](#) [44] and [Project Rephetio manuscript](#) [45].

Citation utilities

The `manubot` [Python package](#) provides easy access to Manubot's citation-by-identifier infrastructure, whose functionality extends beyond just Manubot manuscripts. For example, the [Kipoi](#) model zoo for genomics [46] uses the Python API to retrieve model authors from persistent identifiers. In addition, the `manubot cite` command line utility takes a list of citations and returns either a rendered bibliography or CSL Data Items (i.e. JSON-formatted reference metadata). For example, the following command outputs a Markdown reference list for the two specified articles according to the bibliographic style of *PeerJ*:

```
manubot cite --render --format=markdown \
   --csl=https://github.com/citation-style-language/styles/raw/master/peerj.csl \
   pmid:29618526 doi:10.1038/550143a
```

Pandoc brands itself as a "universal document converter", and can, as of version 2.7, convert from any of 32 input formats to any of 51 output formats. Thanks to its versatility and active development since 2006, Pandoc enjoys a large userbase across many disciplines and applications. Its filter interface enables adding custom functionality with community-developed programs. We are [prototyping](#) a Manubot-based citation-by-identifier filter. This filter would allow Pandoc users to cite persistent identifiers as part of their existing Pandoc workflows, without requiring them to adopt other aspects of Manubot, and could help popularize citation-by-identifier at an influential scale.

Future enhancements

Manubot is still under active development, and we envision major changes in its design and dependencies going forward. Currently, manuscript repositories must contain a large number of files that do not directly contain manuscript content. While this enables a high-degree of customization, it also increases complexity. Therefore, we are investigating whether configuration files with sensible defaults could enable bare-bones repositories that contain manuscript content and little else.

In addition, to simplifying the usage, we're also looking into simplifying the setup. Presently, setup is complex because users must do advanced command-line operations to clone the Rootstock repository and configure Travis CI. While we provide detailed instructions, users often struggle replicating the long list of setup commands in an appropriate computational environment. One priority will be to automate setup to a higher degree. However, this may require switching the services Manubot uses for continuous integration (e.g. from Travis CI to GitHub Actions, CircleCI, Drone, or GitLab CI), environment management (e.g. from Conda to Docker), and repository hosting (e.g. from GitHub to GitLab). In addition to simplifying setup, such migrations may also present the opportunity to decrease dependency on proprietary services and address other Manubot shortcomings, such as the current inability to view rendered manuscripts produced by pull request builds.

Upgrading a Manubot instance is an opt-in procedure. Therefore, when we introduce fundamental changes, existing manuscripts continue to function. However, large Rootstock changes can make upgrading existing manuscripts difficult. We are happy to provide users *pro bono* assistance to upgrade or troubleshoot manuscripts (please open an [issue](#) at `manubot/rootstock` for help).

One strategy to grow Manubot usage is to identify a specific user group or use case for which Manubot can be widely adopted. For example, a journal may decide to build their publishing workflow around Manubot, such that submissions would consist of a Manubot repository. This application would be most suitable for journals that currently use GitHub for submissions and publishing, such as the *Journal of Open Source Software*. Manubot could also gain traction as the primary tool used to write collaborative manuscripts within certain communities. For example, open research projects built from voluntary contributions by geographically-distributed individuals could adopt Manubot. Likewise, Manubot may excel at enabling collaborative translation of existing manuscripts into other languages. Another application could become collaborative development of online lessons, documentation, or tutorials. Projects like [Software Carpentry](#) already host each lesson in a separate GitHub repository, and may be able to benefit from Manubot-generated permalinks to historical versions.

Authorship

Manubot does not impose any restrictions on authorship. It allows authors to adhere to the author inclusion and author ordering conventions of their field, which vary considerably across disciplines [47]. Some Manubot projects create a table in their GitHub repository to track contributors who did not commit text to the manuscript. This provides a transparent way to record contributions such as experimental research that generated data for the manuscript and discuss whether they meet that project's authorship criteria. Contribution transparency helps prevent ghostwriting [48] and is especially important in collaborative writing [49]. Although we recommend authors provide their ORCID and GitHub username, Manubot also supports pseudonyms, pseudonymous GitHub usernames, and authors without an ORCID or GitHub account.

To determine authorship for the Deep Review, we followed the International Committee of Medical Journal Editors (ICMJE) guidelines and used GitHub to track contributions. ICMJE recommends authors substantially contribute to, draft, approve, and agree to be accountable for the manuscript. We acknowledged other contributors who did not meet all four criteria, including contributors who provided text but did not review and approve the complete manuscript. Although these criteria provided a straightforward, equitable way to determine who would be an author, they did not produce a traditionally ordered author list. In biomedical journals, the convention is that the first and last authors made the most substantial contributions to the manuscript. This convention can be difficult to reconcile in a collaborative effort. Using Git, we could quantify the number of commits each author made or the number of sentences an author wrote or edited, but these metrics discount intellectual contributions such as discussing primary literature and reviewing pull requests. Therefore, we concluded that it is not possible to construct an objective system to compare and weight the different types of contributions and produce an ordered author list.

To address this issue, we generalized the concept of "co-first" authorship, in which two or more authors are denoted as making equal contributions to a paper. We defined four types of contributions [3], from major to minor, and reviewed the GitHub discussions and commits to assign authors to these categories. A randomized algorithm then arbitrarily ordered authors within each contribution category, and we combined the category-specific author lists to produce a traditional ordering. The randomization procedure was shared with the authors in advance (pre-registered) and run in a deterministic manner. Given the same author contributions, it always produced the same ordered author list. We annotated the author list to indicate that author order was partly randomized and emphasize that the order did not indicate one author contributed more than another from the same category. The Deep Review author ordering procedure illustrates authorship possibilities when all contributions are publicly tracked and recorded that would be difficult with a traditional collaborative writing platform.

Papers with hundreds or thousands of authors are on the rise, such as the article describing the experiments and data that led to the discovery of the Higgs Boson [50] (5000 authors) and the report of the Drosophila genome [51] (1000 authors). Yet *the number of people that participated in writing* those papers, as opposed to generating and analyzing the data, is not always clear and likely to be far below the number of authors [52,53]. Manubot provides the scientists involved in large collaborations the opportunity to actively participate, through a public forum, in the writing process.

Discussion

Collaborative review manuscripts

The open scholarly writing Manubot enables has particular benefits for review articles, which present the state of the art in a scientific field [54]. Literature reviews are typically written in private by an invited team of colleagues. In contrast, broadly opening the process to anyone engaged in the topic — such that planning, organizing, writing, and editing occur collaboratively in a public forum where anyone is welcome to participate — can maximize a review's value. Open drafting of reviews is especially helpful for capturing state-of-the-art knowledge about rapidly advancing research topics at the intersection of existing disciplines where contributors bring diverse opinions and expertise.

Writing review articles in a public forum allows review authors to engage with the original researchers to clarify their methods and results and present them accurately, as exemplified here. Additionally, discussing manuscripts in the open generates valuable pre-publication peer review of preprints [20] or post-publication peer review [14,55,56]. Because incentives to provide public peer review of existing literature [57] are lacking, open collaborative reviews — where authorship is open to anyone who makes a valid contribution — could help spur more post-publication peer review.

Additional collaborative writing projects

The Deep Review was not the first scholarly manuscript written online via an open collaborative process. In 2013, two dozen mathematicians created the 600-page Homotopy Type Theory book, writing collaboratively in LaTeX on GitHub [58,59]. Two technical books on cryptocurrency — Mastering Bitcoin and Mastering Ethereum — written on GitHub in AsciiDoc format have engaged hundreds of contributors. Both Homotopy Type Theory and Mastering Bitcoin continue to be maintained years after their initial publication. A 2017 perspective on the future of peer review was written collaboratively on Overleaf, with contributions from 32 authors [60]. While debate was raging over tightening the default threshold for statistical significance, nearly 150 scientists contributed to a Google Doc discussion that was condensed into a traditional journal commentary [61,62]. The greatest success to date of open collaborative writing is arguably Wikipedia, whose English version contains over 5.5 million articles. Wikipedia scaled encyclopedias far beyond any privately-written alternative. These examples illustrate how open collaborative writing can scale scholarly manuscripts where diverse opinion and expertise are paramount beyond what would otherwise be possible.

Open writing also presents new opportunities for distributing scholarly communication. Though it is still valuable to have versioned drafts of a manuscript with digital identifiers, journal publication may not be the terminal endpoint for collaborative manuscripts. After releasing the first version of the Deep Review [8], 14 new contributors updated the manuscript (Figure 2). Existing authors continue to discuss new literature, creating a living document. Manubot provides an ideal platform for perpetual reviews [63,64].

Concepts for the future of scholarly publishing extend beyond collaborative writing [65,66]. Bookdown [67] and Pandoc Scholar [10] both extend traditional Markdown to better support publishing. Examples of continuous integration to automate manuscript generation include gh-publisher and Continuous Publishing [68], which was used to produce the book Opening Science [69]. Binder [9], Distill journal articles [70], Idyll [71], and Stencila [72,73] support manuscripts with interactive graphics and close integration with the underlying code. As an open source project, Manubot can be extended to adopt best practices from these other emerging platforms.

Several other open science efforts are GitHub-based like our collaborative writing process. ReScience [74] as well as titles from Open Journals, such as the *Journal of Open Source Software* [75], rely on GitHub for peer review and hosting. Distill uses GitHub for transparent peer review and post-publication peer review [76]. GitHub is increasingly used for resource curation [77], and collaborative scholarly reviews combine literature curation with discussion and interpretation.

Limitations

There are potential limitations of our GitHub-based approach. Because our review manuscript pertained to a computational topic, most of the authors had computational backgrounds, including previous experience with version control workflows and GitHub. In other disciplines, collaborative writing via GitHub and Manubot could present a steeper barrier to entry and deter participants. In addition, Git carefully tracks all revisions to the manuscript text but not the surrounding conversations that take place through GitHub issues and pull requests. These discussions must be archived to ensure that important decisions about the manuscript are preserved and authors receive credit for intellectual contributions that are not directly reflected in the manuscript's text. GitHub supports programmatic access to issues, pull requests, and reviews so tracking these conversations is feasible in the future.

In the Deep Review, we established contributor guidelines that discussed norms in the areas of text contribution, peer review, and authorship, which we identified in advance as potential areas of disagreement. Our contributor guidelines required verifiable participation: either directly attributable changes to the text or participation in the discussion on GitHub. These guidelines did not discuss broader community norms that may have improved inclusiveness. It is also important to consider how the move to an open contribution model affects under-represented minority members of the scientific community [17]. Recent work has identified clear social norms and processes as helpful to maintaining a collaborative culture [78]. Conferences and open source projects have used codes of conduct to establish these norms [79,80]. We would encourage the maintainers of similar projects to consider broader codes of conduct for project participants that build on social as well as academic norms.

Manubot in the context of open science

Science is undergoing a transition towards openness. The internet provides a global information commons, where scholarship can be publicly shared at a minimal cost. For example, open access publishing provides an economic model that encourages maximal dissemination and reuse of scholarly articles [16,81,82]. More broadly, open licensing solves legal barriers to content reuse, enabling any type of scholarly output to become part of the commons [83,84]. The opportunity to reuse data and code for new investigations, as well as a push for increased reproducibility, has begot a movement to make all research outputs public, unless there are bonafide privacy or security concerns [85,86,87]. New tools and services make it increasingly feasible to publicly share the unabridged methods of a study, especially for computational research, which consists solely of software and data.

Greater openness in both research methods and publishing creates an opportunity to redefine peer review and the role journals play in communicating science [60]. At the extreme is real-time open science, whereby studies are performed entirely in the open from their inception [88]. Many such research projects have now been completed, benefiting from the associated early-stage peer review, additional opportunity for online collaboration, and increased visibility [45,89].

Manubot is an ideal authoring protocol for real-time open science, especially for projects that are already using an open source software workflow to manage their research. While Manubot does require technical expertise, the benefits are manyfold. Specifically, Manubot demonstrates a system for publishing that is transparent, reproducible, immediate, permissionless, versioned, automated, collaborative, open, linked, provenanced, decentralized, hackable, interactive, annotated, and free of charge. These attributes empower integrating Manubot with an ecosystem of other community-driven tools to make science as open and collaborative as possible.

Code and data availability

The source code and data for this manuscript are available at https://github.com/greenelab/meta-review and archived via Software Heritage identifier `swh:1:dir:5e644c3a487081b272b2c9b52bcd55caa89c4f85` . Source code for Manubot resides in the following repositories:

- https://github.com/manubot/manubot (GitLab mirror, archived at `swh:1:dir:1eb3b7e6d7e21239fedccf25a186af622ee6912e` , packaged on PyPI)
- https://github.com/manubot/rootstock (GitLab mirror, archived at `swh:1:dir:cfc3af2e8e1a0d9b639fdab0943731e608910731`).

Acknowledgments

Funding

References

1. **TechBlog: "Manubot" powers a crowdsourced "deep-learning" review**
Jeffrey Perkel
Naturejobs (2018-02-20) http://blogs.nature.com/naturejobs/2018/02/20/techblog-manubot-powers-a-crowdsourced-deep-learning-review/

2. **Crowdsourcing in biomedicine: challenges and opportunities**
Ritu Khare, Benjamin M Good, Robert Leaman, Andrew I Su, Zhiyong Lu
Briefings in bioinformatics (2016-01) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4719068/
DOI: 10.1093/bib/bbv021 · PMID: 25888696 · PMCID: PMC4719068

3. **Opportunities and obstacles for deep learning in biology and medicine**
Travers Ching, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, Alexandr A. Kalinin, Brian T. Do, Gregory P. Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M. Hoffman, … Casey S. Greene
Journal of The Royal Society Interface (2018-04) https://doi.org/gddkhn
DOI: 10.1098/rsif.2017.0387 · PMID: 29618526 · PMCID: PMC5938574

4. **Scientific writing: the online cooperative**
Jeffrey M. Perkel
Nature (2014-10-01) https://doi.org/gbqsnd
DOI: 10.1038/514127a · PMID: 25279924

5. **A Quick Introduction to Version Control with Git and GitHub**
John D. Blischak, Emily R. Davenport, Greg Wilson
PLOS Computational Biology (2016-01-19) https://doi.org/gbqsnf
DOI: 10.1371/journal.pcbi.1004668 · PMID: 26785377 · PMCID: PMC4718703

6. **Ten Simple Rules for Taking Advantage of Git and GitHub**
Yasset Perez-Riverol, Laurent Gatto, Rui Wang, Timo Sachsenberg, Julian Uszkoreit, Felipe da Veiga Leprevost, Christian Fufezan, Tobias Ternent, Stephen J. Eglen, Daniel S. Katz, … Juan Antonio Vizcaíno
PLOS Computational Biology (2016-07-14) https://doi.org/gbrb39
DOI: 10.1371/journal.pcbi.1004947 · PMID: 27415786 · PMCID: PMC4945047

7. **Opportunities And Obstacles For Deep Learning In Biology And Medicine**
Johnny Israeli
Towards Data Science (2017-05-31) https://towardsdatascience.com/opportunities-and-obstacles-for-deep-learning-in-biology-and-medicine-6ec914fe18c2

8. **Opportunities And Obstacles For Deep Learning In Biology And Medicine**
Travers Ching, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, Alexandr A. Kalinin, Brian T. Do, Gregory P. Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, … Casey S. Greene
Cold Spring Harbor Laboratory (2017-05-28) https://doi.org/gbpvh5
DOI: 10.1101/142760

9. **Binder 2.0 - Reproducible, interactive, sharable environments for science at scale**
Project Jupyter, Matthias Bussonnier, Jessica Forde, Jeremy Freeman, Brian Granger, Tim Head, Chris Holdgraf, Kyle Kelley, Gladys Nalvarte, Andrew Osheroff, … Carol Willing
Proceedings of the 17th Python in Science Conference (2018) https://doi.org/gfwcm6
DOI: 10.25080/majora-4af1f417-011

10. **Formatting Open Science: agilely creating multiple document formats for academic manuscripts with Pandoc Scholar**
Albert Krewinkel, Robert Winkler
PeerJ Computer Science (2017-05-08) https://doi.org/gbrb4c
DOI: 10.7717/peerj-cs.112

11. **Reference Management**
Martin Fenner, Kaja Scheliga, Sönke Bartling
Opening Science (2013-12-17) https://doi.org/gbxtc8
DOI: 10.1007/978-3-319-00026-8_8

12. **Comparison of Select Reference Management Tools**
Yingting Zhang
Medical Reference Services Quarterly (2012-01) https://doi.org/hpv
DOI: 10.1080/02763869.2012.641841 · PMID: 22289095

13. **Twenty-Five Shades of Greycite: Semantics for referencing and preservation**
Phillip Lord, Lindsay Marshall
arXiv (2013-04-26) https://arxiv.org/abs/1304.7151v1

14. **Reviewing post-publication peer review.**
Paul Knoepfler
Trends in genetics : TIG (2015-04-04) https://www.ncbi.nlm.nih.gov/pubmed/25851694
DOI: 10.1016/j.tig.2015.03.006 · PMID: 25851694 · PMCID: PMC4472664

15. **Decentralized Trusted Timestamping using the Crypto Currency Bitcoin**
Bela Gipp, Norman Meuschke, André Gernandt
arXiv (2015-02-13) https://arxiv.org/abs/1502.04015v1

16. **Open access**
Peter Suber
MIT Press (2012)
ISBN: 9780262517638

17. **Open science and open science**
Laurent Gatto
(2017-06-05) https://lgatto.github.io/open-and-open/

18. **Plan S: Accelerating the transition to full and immediate Open Access to scientific publications**
cOAlition S
(2018-09-04) https://www.wikidata.org/wiki/Q56458321

19. **Conversation with Dongbo Hu regarding how to administer a cloud server** *Greene Laboratory*
(2018-12-19)

20. **Journal clubs in the time of preprints**
Prachee Avasthi, Alice Soragni, Joshua N Bembenek
eLife (2018-06-11) https://doi.org/gdm89h
DOI: 10.7554/elife.38532 · PMID: 29889024 · PMCID: PMC5995539

21. **On author versus numeric citation styles**
Daniel Himmelstein
Satoshi Village (2018-03-12) https://blog.dhimmel.com/citation-styles/

22. **TechBlog: Create the perfect bibliography with the CSL Editor**
Jeffrey Perkel
Naturejobs (2017-05-03) http://blogs.nature.com/naturejobs/2017/05/03/techblog-create-the-perfect-bibliography-with-the-csl-editor/

23. **JATS: Journal Article Tag Suite, version 1.1**
National Information Standards Organization
(2015) http://www.niso.org/standards/z39-96-2015/

24. **Journal Article Tag Suite 1.0: National Information Standards Organization standard of journal extensible markup language**
Sun Huh
Science Editing (2014-08-18) https://doi.org/gbxtdk
DOI: 10.6087/kcse.2014.1.99

25. **NISO Z39.96-201x, JATS: Journal Article Tag Suite**
Mark H. Needleman
Serials Review (2012-09) https://doi.org/gbxtdj
DOI: 10.1080/00987913.2012.10765464

26. **Data visualization tools drive interactivity and reproducibility in online publishing**
Jeffrey M. Perkel
Nature (2018-02-01) https://doi.org/gfw6g3
DOI: 10.1038/d41586-018-01322-9 · PMID: 29388968

27. **Vega-Lite: A Grammar of Interactive Graphics**
Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, Jeffrey Heer
IEEE Transactions on Visualization and Computer Graphics (2017-01) https://doi.org/f92f32
DOI: 10.1109/tvcg.2016.2599030 · PMID: 27875150

28. **Collaborative software development made easy**
Andrew Silver
Nature (2017-10-04) https://doi.org/cdvr
DOI: 10.1038/550143a · PMID: 28980652

29. **Reproducibility of computational workflows is automated using continuous analysis**
Brett K Beaulieu-Jones, Casey S Greene
Nature Biotechnology (2017-03-13) https://doi.org/f9ttx6
DOI: 10.1038/nbt.3780 · PMID: 28288103 · PMCID: PMC6103790

30. **Developing a modern data workflow for evolving data**
Glenda M Yenni, Erica M Christensen, Ellen K Bledsoe, Sarah R Supp, Renata M Diaz, Ethan P White, SK Morgan Ernest
Cold Spring Harbor Laboratory (2018-06-12) https://doi.org/gdqbzn
DOI: 10.1101/344804

31. **Priority of discovery in the life sciences**
Ronald D Vale, Anthony A Hyman
eLife (2016-06-16) https://doi.org/gcx6gx
DOI: 10.7554/elife.16931 · PMID: 27310529 · PMCID: PMC4911212

32. **The Grey Literature — Proof of prespecified endpoints in medical research with the bitcoin blockchain**https://www.bgcarlisle.com/blog/2014/08/25/proof-of-prespecified-endpoints-in-medical-research-with-the-bitcoin-blockchain/

33. **The most interesting case of scientific irreproducibility?**
Daniel Himmelstein
Satoshi Village (2017-03-08) https://blog.dhimmel.com/irreproducible-timestamps/

34. **Bitcoin for the biological literature**
Douglas Heaven
Nature (2019-02) https://doi.org/gft5gp
DOI: 10.1038/d41586-019-00447-9 · PMID: 30718888

35. **Bitcoin: A Peer-to-Peer Electronic Cash System**
Satoshi Nakamoto
(2019-03-21) https://git.dhimmel.com/bitcoin-whitepaper/

36. **OpenTimestamps: Scalable, Trust-Minimized, Distributed Timestamping with Bitcoin**https://petertodd.org/2016/opentimestamps-announcement

37. **eLife supports development of open technology stack for publishing reproducible manuscripts online**
Emily Packer
(2017-09-07) https://elifesciences.org/for-the-press/e6038800/elife-supports-development-of-open-technology-stack-for-publishing-reproducible-manuscripts-online

38. **Sci-Hub provides access to nearly all scholarly literature**
Daniel S Himmelstein, Ariel Rodriguez Romero, Jacob G Levernier, Thomas Anthony Munro, Stephen Reid McLaughlin, Bastian Greshake Tzovaras, Casey S Greene
eLife (2018-03-01) https://doi.org/ckcj
DOI: 10.7554/elife.32822 · PMID: 29424689 · PMCID: PMC5832410

39. **2017 in news: The science events that shaped the year**
Ewen Callaway, Davide Castelvecchi, David Cyranoski, Elizabeth Gibney, Heidi Ledford, Jane J. Lee, Lauren Morello, Nicky Phillips, Quirin Schiermeier, Jeff Tollefson, … Alexandra Witze
Nature (2017-12-21) https://doi.org/chnh
DOI: 10.1038/d41586-017-08493-x · PMID: 29293246

40. **GimmeMotifs: an analysis framework for transcription factor motif analysis**
Niklas Bruse, Simon J. van Heeringen
Cold Spring Harbor Laboratory (2018-11-20) https://doi.org/gfxrkc
DOI: 10.1101/474403

41. **Plasmids for independently tunable, low-noise expression of two genes**
João P. N. Silva, Soraia Vidigal Lopes, Diogo J. Grilo, Zach Hensel

Cold Spring Harbor Laboratory (2019-01-09) https://doi.org/gfs47c
DOI: 10.1101/515940

42. **A set of common software quality assurance baseline criteria for research projects**
Pablo Orviz, Álvaro López García, Doina Cristina Duma, Giacinto Donvito, Mario David, Jorge Gomes
(2017) https://digital.csic.es/handle/10261/160086
DOI: http://dx.doi.org/10.13039/501100000780

43. **Vagelos Report Summer 2017**
Michael Zietz
Figshare (2017-08-25) https://doi.org/gbr3pf
DOI: 10.6084/m9.figshare77

44. **How I used the Manubot to reproduce the Bitcoin Whitepaper**
Daniel Himmelstein
Steem (2017-09-20) https://busy.org/@dhimmel/how-i-used-the-manubot-to-reproduce-the-bitcoin-whitepaper

45. **Systematic integration of biomedical knowledge prioritizes drugs for repurposing**
Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, Sergio E Baranzini
eLife (2017-09-22) https://doi.org/cdfk
DOI: 10.7554/elife.26726 · PMID: 28936969 · PMCID: PMC5640425

46. **Kipoi: accelerating the community exchange and reuse of predictive models for genomics**
Ziga Avsec, Roman Kreuzhuber, Johnny Israeli, Nancy Xu, Jun Cheng, Avanti Shrikumar, Abhimanyu Banerjee, Daniel S Kim, Lara Urban, Anshul Kundaje, … Julien Gagneur
Cold Spring Harbor Laboratory (2018-07-24) https://doi.org/gd24sx
DOI: 10.1101/375345

47. **A Systematic Review of Research on the Meaning, Ethics and Practices of Authorship across Scholarly Disciplines**
Ana Marušić, Lana Bošnjak, Ana Jerončić
PLoS ONE (2011-09-08) https://doi.org/fp5drd
DOI: 10.1371/journal.pone.0023477 · PMID: 21931600 · PMCID: PMC3169533

48. **What Should Be Done To Tackle Ghostwriting in the Medical Literature?**
Peter C Gøtzsche, Jerome P Kassirer, Karen L Woolley, Elizabeth Wager, Adam Jacobs, Art Gertel, Cindy Hamilton
PLoS Medicine (2009-02-03) https://doi.org/bnzbx7
DOI: 10.1371/journal.pmed.1000023 · PMID: 19192943 · PMCID: PMC2634793

49. **Ten simple rules for collaboratively writing a multi-authored paper**
Marieke A. Frassl, David P. Hamilton, Blaize A. Denfeld, Elvira de Eyto, Stephanie E. Hampton, Philipp S. Keller, Sapna Sharma, Abigail S. L. Lewis, Gesa A. Weyhenmeyer, Catherine M. O'Reilly, … Núria Catalán
PLOS Computational Biology (2018-11-15) https://doi.org/gfj8wf
DOI: 10.1371/journal.pcbi.1006508 · PMID: 30439938 · PMCID: PMC6237291

50. **Combined Measurement of the Higgs Boson Mass inppCollisions ats=7and 8 TeV with the ATLAS and CMS Experiments**
G. Aad, B. Abbott, J. Abdallah, O. Abdinov, R. Aben, M. Abolins, O. S. AbouZeid, H. Abramowicz, H.

Abreu, R. Abreu, …
Physical Review Letters (2015-05-14) https://doi.org/f3nr73
DOI: 10.1103/physrevlett.114.191803 · PMID: 26024162

51. **Drosophila Muller F Elements Maintain a Distinct Set of Genomic Properties Over 40 Million Years of Evolution**
Wilson Leung, Christopher D. Shaffer, Laura K. Reed, Sheryl T. Smith, William Barshop, William Dirkes, Matthew Dothager, Paul Lee, Jeannette Wong, David Xiong, … Sarah C. R. Elgin
G3: Genes|Genomes|Genetics (2015-03-04) https://doi.org/gfw5vf
DOI: 10.1534/g3.114.015966 · PMID: 25740935 · PMCID: PMC4426361

52. **Fruit-fly paper has 1,000 authors**
Chris Woolston
Nature (2015-05-13) https://doi.org/gckfqm
DOI: 10.1038/521263f

53. **Physics paper sets record with more than 5,000 authors**
Davide Castelvecchi
Nature (2015-05-15) https://doi.org/4sn
DOI: 10.1038/nature.2015.17567

54. **Ten Simple Rules for Writing a Literature Review**
Marco Pautasso
PLoS Computational Biology (2013-07-18) https://doi.org/gcx9dm
DOI: 10.1371/journal.pcbi.1003149 · PMID: 23874189 · PMCID: PMC3715443

55. **A Stronger Post-Publication Culture Is Needed for Better Science**
Hilda Bastian
PLoS Medicine (2014-12-30) https://doi.org/gdm9cj
DOI: 10.1371/journal.pmed.1001772 · PMID: 25548904 · PMCID: PMC4280106

56. **Post-Publication Peer Review: Opening Up Scientific Conversation**
Jane Hunter
Frontiers in Computational Neuroscience (2012) https://doi.org/gdm9cm
DOI: 10.3389/fncom.2012.00063 · PMID: 22969719 · PMCID: PMC3431010

57. **Post-publication peer review, in all its guises, is here to stay**
Michael Markie
Insights the UKSG journal (2015-07-07) https://doi.org/gdm9ck
DOI: 10.1629/uksg.245

58. **Homotopy Type Theory: Univalent Foundations of Mathematics**
The Univalent Foundations Program
Institute for Advanced Study (2013) https://homotopytypetheory.org/book/

59. **The HoTT book**
Andrej Bauer
Mathematics and Computation (2013-06-20) http://math.andrej.com/2013/06/20/the-hott-book/

60. **A multi-disciplinary perspective on emergent and future innovations in peer review**
Jonathan P. Tennant, Jonathan M. Dugan, Daniel Graziotin, Damien C. Jacques, François Waldner,

Daniel Mietchen, Yehia Elkhatib, Lauren B. Collister, Christina K. Pikas, Tom Crick, … Julien Colomb
F1000Research (2017-11-01) https://doi.org/gc5tcv
DOI: 10.12688/f1000research.12037.2 · PMID: 29188015 · PMCID: PMC5686505

61. **Nearly 100 scientists spent 2 months on Google Docs to redefine the p-value. Here's what they came up with**
Jop Vrieze
Science (2018-01-18) https://doi.org/gc5tct
DOI: 10.1126/science.aat0471

62. **Justify your alpha**
Daniel Lakens, Federico G. Adolfi, Casper J. Albers, Farid Anvari, Matthew A. J. Apps, Shlomo E. Argamon, Thom Baguley, Raymond B. Becker, Stephen D. Benning, Daniel E. Bradford, … Rolf A. Zwaan
Nature Human Behaviour (2018-02-26) https://doi.org/gcz8f3
DOI: 10.1038/s41562-018-0311-x

63. **A proposal for regularly updated review/survey articles: "Perpetual Reviews"**
David L. Mobley, Daniel M. Zuckerman
arXiv (2015-02-03) https://arxiv.org/abs/1502.01329v2

64. **Why we need the Living Journal of Computational Molecular Science**
Daniel M. Zuckerman, Michael R. Shirts, David L. Mobley
Living Journal of Computational Molecular Science (2017-08-22) https://www.livecomsjournal.org/article/2031-why-we-need-the-living-journal-of-computational-molecular-science
DOI: 10.33011/livecoms.1.1.2031

65. **The arXiv of the future will not look like the arXiv**
Alberto Pepe, Matteo Cantiello, Josh Nicholson
Authorea https://doi.org/gdqbz3
DOI: 10.22541/au.149693987.70506124

66. **TechBlog: C. Titus Brown: Predicting the paper of the future**
C. Titus Brown
Naturejobs (2017-06-01) http://blogs.nature.com/naturejobs/2017/06/01/techblog-c-titus-brown-predicting-the-paper-of-the-future/

67. **bookdown**
Yihui Xie
Chapman &Hall/CRC The R Series (2016-12-21) https://doi.org/gdqbz2
DOI: 10.1201/9781315204963

68. **Continuous Publishing**
Martin Fenner
Gobbledygook (2014-03-10) http://blog.martinfenner.org/2014/03/10/continuous-publishing/

69. **Opening Science** *Springer International Publishing* (2014) https://doi.org/gdqbzz
DOI: 10.1007/978-3-319-00026-8

70. **The Building Blocks of Interpretability**
Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, Alexander

Mordvintsev
Distill (2018-03-06) https://doi.org/gdvhz5
DOI: 10.23915/distill.00010

71. **Announcing idyll.pub**
Matthew Conlen, Andrew Osheroff
Idyll (2018-06-26) https://idyll.pub/post/announcing-idyll-pub-0a3eff0661df3446a915700d/

72. **Stencila – an office suite for reproducible research**
Michael Aufreiter, Aleksandra Pawlik, Nokome Bentley
eLife Labs (2018-07-02) https://elifesciences.org/labs/c496b8bb/stencila-an-office-suite-for-reproducible-research

73. **Introducing eLife's first computationally reproducible article** *eLife* (2019-02-20) https://elifesciences.org/labs/ad58f08d/introducing-elife-s-first-computationally-reproducible-article

74. **Sustainable computational science: the ReScience initiative**
Nicolas P. Rougier, Konrad Hinsen, Frédéric Alexandre, Thomas Arildsen, Lorena A. Barba, Fabien C.Y. Benureau, C. Titus Brown, Pierre de Buyl, Ozan Caglayan, Andrew P. Davison, … Tiziano Zito
PeerJ Computer Science (2017-12-18) https://doi.org/gcx5kf
DOI: 10.7717/peerj-cs.142

75. **Journal of Open Source Software (JOSS): design and first-year review**
Arfon M. Smith, Kyle E. Niemeyer, Daniel S. Katz, Lorena A. Barba, George Githinji, Melissa Gymrek, Kathryn D. Huff, Christopher R. Madan, Abigail Cabunoc Mayes, Kevin M. Moerman, … Jacob T. Vanderplas
PeerJ Computer Science (2018-02-12) https://doi.org/gc5sjf
DOI: 10.7717/peerj-cs.147

76. **Distill Update 2018**
Distill Editors
Distill (2018-08-14) https://doi.org/gfbzs9
DOI: 10.23915/distill.00013

77. **The appropriation of GitHub for curation**
Yu Wu, Na Wang, Jessica Kropczynski, John M. Carroll
PeerJ Computer Science (2017-10-09) https://doi.org/gb3bxk
DOI: 10.7717/peerj-cs.134

78. **Innovating Collaborative Content Creation: The Role of Altruism and Wiki Technology**
Christian Wagner, Pattarawan Prasarnphanich
2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07) (2007) https://doi.org/b6vqgx
DOI: 10.1109/hicss.2007.277

79. **Code of conduct evaluations** *Geek Feminism Wiki* http://geekfeminism.wikia.com/wiki/Code_of_conduct_evaluations

80. **Contributor Covenant: A Code of Conduct for Open Source Projects**
Coraline Ada Ehmke
(2014) https://www.contributor-covenant.org/

81. **The academic, economic and societal impacts of Open Access: an evidence-based review**
Jonathan P. Tennant, François Waldner, Damien C. Jacques, Paola Masuzzo, Lauren B. Collister, Chris. H. J. Hartgerink
F1000Research (2016-09-21) https://doi.org/gbqrbc
DOI: 10.12688/f1000research.8460.3 · PMID: 27158456 · PMCID: PMC4837983

82. **How open science helps researchers succeed**
Erin C McKiernan, Philip E Bourne, C Titus Brown, Stuart Buck, Amye Kenall, Jennifer Lin, Damon McDougall, Brian A Nosek, Karthik Ram, Courtney K Soderberg, … Tal Yarkoni
eLife (2016-07-07) https://doi.org/gbqsng
DOI: 10.7554/elife.16800 · PMID: 27387362 · PMCID: PMC4973366

83. **The Legal Framework for Reproducible Scientific Research: Licensing and Copyright**
Victoria Stodden
Computing in Science & Engineering (2009-01) https://doi.org/b7tskf
DOI: 10.1109/mcse.2009.19

84. **Legal confusion threatens to slow data science**
Simon Oxenham
Nature (2016-08-03) https://doi.org/bndt
DOI: 10.1038/536016a · PMID: 27488781

85. **Enhancing reproducibility for computational methods**
V. Stodden, M. McNutt, D. H. Bailey, E. Deelman, Y. Gil, B. Hanson, M. A. Heroux, J. P. A. Ioannidis, M. Taufer
Science (2016-12-08) https://doi.org/gbr42b
DOI: 10.1126/science.aah6168 · PMID: 27940837

86. **The case for open computer programs**
Darrel C. Ince, Leslie Hatton, John Graham-Cumming
Nature (2012-02-22) https://doi.org/hqg
DOI: 10.1038/nature10836 · PMID: 22358837

87. **The Open Knowledge Foundation: Open Data Means Better Science**
Jennifer C. Molloy
PLoS Biology (2011-12-06) https://doi.org/g3b
DOI: 10.1371/journal.pbio.1001195 · PMID: 22162946 · PMCID: PMC3232214

88. **This revolution will be digitized: online tools for radical collaboration**
C. Patil, V. Siegel
Disease Models & Mechanisms (2009-04-30) https://doi.org/fvjhcj
DOI: 10.1242/dmm.003285 · PMID: 19407323 · PMCID: PMC2675795

89. **Publishing the research process**
Daniel Mietchen, Ross Mounce, Lyubomir Penev
Research Ideas and Outcomes (2015-12-17) https://doi.org/f3mn7d
DOI: 10.3897/rio.1.e7547