A guide to using GitHub for developing and versioning data standards and reporting formats

Robert Crystal-Ornelas[1*], Charuleka Varadharajan[1], Ben Bond-Lamberty[2], Kristin Boye[3], Madison Burrus[1], Shreyas Cholia[4], Michael Crow[5], Joan Damerow[1], Ranjeet Devarakonda[5], Kim S. Ely[6], Amy Goldman[7], Susan Heinz[5], Valerie Hendrix[4], Zarine Kakalia[1], Stephanie Pennington[2], Emily Robles[1], Alistair Rogers[6], Maegen Simmonds[1], Terri Velliquette[5], Helen Weierbach[1], Pamela Weisenhorn[8], Jessica Nicole Welch[5], Deborah A. Agarwal[4]

[1]Earth and Environmental Sciences Area, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

[2]Pacific Northwest National Laboratory, Joint Global Change Research Institute at the University of Maryland–College Park, College Park, MD 20740, USA

[3]Geochemistry and Biogeochemistry Group, SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025, USA

[4]Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA

[5]Environmental Sciences Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA

[6]Department of Environmental and Climate Sciences, Brookhaven National Laboratory, Upton, NY 11973, USA

[7]Pacific Northwest National Laboratory, Richland, WA 99345, USA

[8]Argonne National Laboratory, Lemont, IL 60439, USA

Corresponding author: Robert Crystal-Ornelas (rcrystalornelas@lbl.gov)

Key Points:

- Developing data standards on Version Control System platforms like GitHub enables collaboration and transparency.
- Many standards do not use tools for collaboration: issue tracking, licensing, and automated website hosting (GitBook or GitHub Pages).
- We make recommendations and provide templates for creating descriptive version-controlled data standard documentation on GitHub.

This article has been accepted for publication and undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process, which may lead to differences between this version and the Version of Record. Please cite this article as doi: 10.1029/2021EA001797.

Abstract

Data standardization combined with descriptive metadata facilitate data reuse, which is the ultimate goal of the Findable, Accessible, Interoperable, and Reusable (FAIR) principles. Community data or metadata standards are increasingly created through an approach that emphasizes collaboration between various stakeholders. Such an approach requires platforms for collaboration on the development process that centers on sharing information and receiving feedback. Our objective in this study was to conduct a systematic review to identify data standards and reporting formats that use version control for developing data standards and to summarize common practices, particularly in earth and environmental sciences. Of 108 data standards and reporting formats identified in our review, 32 used GitHub as the version control platform and no other platforms were used. We found no universally accepted methodology for developing and publishing data standards. Many GitHub repositories did not use key features that could help developers to gather user feedback, or to create and revise standards that build on previous work. We provide guidance for community-driven standard development and associated documentation on GitHub based on a systematic review of existing practices.

1 Introduction

Many researchers are required to submit their data to a data repository. However, such data may not reach their full potential for reuse unless they are archived in a way that enables a general user of the data to interpret and extract information. For example, in one study where researchers attempted to reuse 100 datasets stored in a long-term data repository, they found that over 60% of the datasets were unusable due to incomplete datasets or insufficient metadata (Roche et al., 2015). This data reusability problem is echoed by a growing emphasis on data stewardship through FAIR (Findable Accessible Interoperable Reusable) data principles (Wilkinson et al., 2016). While adherence to FAIR principles in earth and environmental sciences has been accelerated by funder and journal requirements, implementation of the four FAIR principles in data repositories has not seen full realization.

Community-led efforts to standardize data have emerged as one path toward ensuring that data stored in long-term repositories are well-described and consistently formatted for greater reusability (Sansone et al., 2019). Developing formal *data standards* (including common data policies, protocols, and documents) is often a rigorous international certification process that requires approval through a governing body of experts (e.g., https://www.iso.org/certification.html). In contrast, *reporting formats* or community conventions are agreed-upon terms and formatting guidelines for data and metadata sharing amongst a community of researchers that can more easily be developed and adopted without a formal certification process. Reporting formats may incorporate some elements of data standards (e.g., timestamps following the ISO 8601 standard; ISO, 2019), and may eventually become data standards if widely adopted. Data standards and reporting formats promote data reusability (Hart et al., 2016; Pasquetto et al., 2017; Zimmerman, 2008) by enabling efficient integration and interpretation of similarly formatted research data and metadata (Read et al., 2013; US EPA, 2015; Yarmey & Baker, 2013). For convenience, here we use the term "standards" to refer to formal data and metadata standards, reporting formats, and numerous other related terms (Table 1).

Education and outreach with the research community can facilitate both the creation (Sansone et al., 2019) and early adoption of community-developed data standards, as has been successfully demonstrated in the biological sciences (De Pooter et al., 2017; Galdzicki et al., 2014; Wieczorek et al. 2012). Outreach can take many forms, including annual meetings and

webinars, engaging diverse disciplines in discussion and testing, and leveraging open web platforms for potential users to preview the standard. Standards may evolve over time as user feedback is generated or new observational methodologies (e.g., sensor technology) necessitate modifications (Bezuidenhout, 2020).

For standards to be widely used by scientists, clear definitions and documentation are essential, which must describe current and past versions of the standards. Both standards and documentation can evolve with community input, which motivates the need for documenting data standards with systems that support versioning (i.e., detailed tracking of changes to multiple documents from one version to the next) and tracking user input. Such version control systems (VCS) are typically used for collaborative software development, and there are many web-based hosting services for them (e.g., GitHub, BitBucket, GitLab, CodeCommit, SourceForge). VCSs are also well suited for chronicling the collaborative development of data standards, which like software are oriented around text-based documents (Mergel, 2015; Perkel, 2016; Schneider et al., 2019). A key feature of most VCS is transparency—direct and suggested changes to content are visible. Note that throughout this paper we use the term "VCS platform" as a shorthand for a modern web-hosted software collaboration environment that combines VCS systems with code browsing and editing, issue tracking, documentation, continuous integration, and other tools for enabling software development across teams.

With over 56 million users, GitHub is one of the most popular VCS platforms (GitHub, 2020b). In addition to software development, GitHub is increasingly used as a platform for collaboration on documents, such as standards that require versioning. The software coding community on GitHub has identified four best practices researchers can take to increase a GitHub repository's visibility and reusability. The first is to create a descriptive "README" file (Lee et al., 2021), written in the markdown coding language, which helps consistently format documents on GitHub. This is the homepage of a GitHub repository, and should provide the user with details like "What the project does" and "How users can get started with the project" (GitHub, 2020a). The second is to license the code (or more generally, content) within a GitHub repository to clearly and precisely specify any conditions attached to its use and reuse (Lee et al., 2021; Stoudt et al., 2021). The third is to use GitHub for collaboration by submitting issues (i.e., making a comment on a repository) or pull requests (editing a copy of repository content and then asking the owners to "pull" the changes into existing content; Bissyandé et al., 2013).

GitHub repository owners may choose to describe their preferred methods for collaboration by creating a markdown document called CONTRIBUTING.md (Sholler et al. 2019). Lastly, GitHub-based developers may create a project webpage using services like GitHub Pages (https://pages.github.com/) that mirror some or all content within a repository to an external project website (Angulo & Aktunc, 2019; Tantisuwankul et al., 2019).

ESS-DIVE (Environmental Systems Science Data Infrastructure for a Virtual Ecosystem) is the Department of Energy's (DOE) data repository for Environmental Systems Science (ESS) research (Varadharajan et al., 2019). Starting in 2019, the ESS-DIVE team partnered with domain experts to develop data reporting formats for a suite of data types ranging from file-level and CSV metadata to domain-specific, such as soil and leaf respiration and hydrological data. While developing the data reporting formats, the domain experts solicited extensive feedback from the communities who would ultimately supply and use the data. Therefore, the documentation system needed the capability to track rounds of community feedback across multiple documents and versions. Thus, we chose GitHub as a natural VCS platform for tracking changes, comments and issues for the proposed reporting formats. In the process, we found that although version control allows for management and collaboration on standards development, there was a need for guidance on how to best leverage the broader collaborative features of VCS platforms such as GitHub for community-developed standards.

The overall objective of our research was to conduct a systematic review to inform the development of a community-driven approach for describing data standards using a VCS platform, with a focus on GitHub. Specifically, we sought to: (i) characterize the version-controlled documentation for existing data and metadata standards, (ii) identify how managers of VCS websites ask users for feedback on standards, and (iii) record whether repository managers build user-facing websites (in addition to their VCS site) for hosting version-controlled documents. In this paper, we present the results of how 32 groups developing data standards and reporting formats have organized their GitHub repositories, and provide recommendations for structuring VCS for groups taking a community-driven approach to data standard and reporting format development. We also provide a set of guidelines and example templates for managing data and metadata standards and reporting formats on GitHub and other VCS platforms.

2 Methods

2.1 Identifying VCS repositories used for managing data standards

We conducted a systematic search for groups using VCS to document data standards, data reporting formats, and ontologies (hereafter, data standards; Table 1). In September 2020, we used FAIRsharing.org's data standard search tool (https://fairsharing.org/standards/; Sansone et al., 2019) to locate existing data standards.

We retained only data standards that FAIRsharing.org classified as "ready" rather than "in development." We further filtered the database to select standards associated with the following domains: earth science (n = 65), ecology (n = 34), and environmental science (n = 31). We identified an additional 24 standards that were not captured by our initial search but were recommended by domain experts (Data used in this analysis are available in Crystal-Ornelas et al., 2021a).

2.2 Selecting relevant VCS repositories

We identified 155 potentially relevant data standards for selection. First, we removed any duplicate data standards that appeared multiple times in our search results (n = 47 duplicates). Then, we retained only standards that use GitHub as the VCS platform for actively managing documentation. We chose GitHub because it was the predominant platform (n = 60 used GitHub out of n = 108 standards reviewed). In fact, it is notable that amongst all the standards reviewed, only one organization used a different VCS platform (BitBucket). Finally, we excluded GitHub repositories that were used for simply hosting binary or non-text files (e.g., MS Word document or MS Excel spreadsheets; n = 28 excluded), and thus included only groups that used GitHub for active management and collaboration on data standards (n = 32).

2.4 Characterizing content within GitHub repositories

We visited each GitHub repository identified during our systematic search (Supplementary Information Table S1), and characterized the documents and content within each repository according to five general topics: (1) contents of the entire GitHub repository, (2) README page content, (3) preferred methods for collaboration and receiving feedback, (4) labels for tracking issues within a repository, and (5) user-facing project websites.

To characterize the content on each GitHub repository and in README files (Topics 1 and 2), we developed a set of standardized terms for content (e.g., "about section" or "recommended citation"; See Table 2 for a full list of terms and definitions) based on a pilot

screening of 10 GitHub repositories. Sometimes during the data collection process we identified a new term (e.g., "code of conduct") not previously found during the screening process and added it to our list of terms. We analyzed repository-wide content and README content separately to identify if content was more often included as part of a repository's README file (i.e., the repository's homepage) or contained in sub-folders of a repository.

We then broadly reviewed the way that each GitHub repository suggested visitors contribute revisions or updates to their version-controlled documents (Topic 3). To do this, we categorized the preferred method of collaboration for each repository as (1) issue submissions, (2) pull requests (i.e., suggesting changes directly to content/documents) and issue submissions, or (3) unclear contribution method. We then conducted a more detailed characterization of content within repositories that supports user collaboration. Similar to the methods used for Topics 1 and 2, we created a set of standardized content terms related to contributing to GitHub repositories (e.g., "issue templates" or "GitHub tutorials;" See Table 3 for a full list of terms and definitions). Then we manually identified whether repositories included the content terms or not.

We used text analysis tools to identify any GitHub issue labels that were commonly used for tracking and organizing user-submitted GitHub issues. We carried out two steps to prepare the text (i.e., "issue labels"). First, we "tokenized" all labels using the python module *re* (Python Software Foundation, 2020) to create a ready-to-analyze list of all label text. Then, we "stemmed" each label, which removes suffixes in order to enable clearer grouping of words with similar stems. For example, the labels "reviewed" and "reviewing" would be stemmed to the root "review". Then, we counted the frequency of each stemmed label and further grouped stemmed labels by visual inspection where necessary.

Lastly, we visited each GitHub repository to determine if repositories had separate project websites for those standards (e.g., https://environmentaldatainitiative.org/ or https://cfconventions.org/) and, if so, identified the service they used to create those websites.

We also recorded if the version-controlled documentation was also stored in a long-term data repository (e.g., Zenodo, Dryad, Figshare).

3 Results

Our systematic search located 108 data standards, guidelines, and reporting formats as well as ontologies in earth science, environmental science, and ecology. There was variety in the platforms used to manage standards (Table 4). In general, data standards were either hosted using GitHub (n = 60, 55%) or through the organization's website (n = 42, 39%). When the data standard documentation was hosted on an organization's website, the site often provided links to data standard documentation in PDF, RDF, or CSV format. Notably, out of the 108 data standards in our review, only 17 (18%) were published and stored in a recognized data repository.

3.1 Version control content

Most GitHub repositories (94%) convey general information on the data standard by using an "About" section (Figure 1a). Because all repositories in our review focused on data standards, reporting formats, or ontologies, the repositories often contained a file with descriptions of key terms and definitions of those terms (91%). Another frequently used documentation method was indicating the current version of the data standard using semantic versioning for tracking releases (e.g., v1.0.3; Preston-Werner, 2020). The current version of the data standard was often listed in the body of the repository's README file or by using the built-in "Release" widget (which in turn leverages the underlying Git VCS's "tag" mechanism) that is part of every GitHub repository home page.

Some less common elements of GitHub repositories include usage licenses (56%), recommended citations for the standard (31%), funding information (22%), and "Getting Started" sections (34%). Getting started content is typically different from "About" sections because it provides a table of contents to the GitHub repository complete with links and information on how to quickly make use of documents, folders, or templates within a repository.

The patterns we found throughout GitHub repositories are generally mirrored in our analysis that focused on GitHub README files (Figure 1b).

3.2 Ways to contribute to data standards

Only 18 (56%) of the repositories in our review encourage contributions, and only 10 (31%) provide detailed instructions on how to suggest changes to the documentation (Figure 2). Seven repositories provided detailed guides for contributing using GitHub recommended CONTRIBUTING.md files within their repository's root folder, or in a '.github' folder. Three repositories provided step-by-step tutorials for making contributions.

Repository managers on GitHub asked users for feedback in several different ways. Of the 32 repositories that were part of our review, 9 (28%) suggest that users submit GitHub issues to make suggestions for revising data standards. It was more common (n = 13; 41%) for repositories to allow both pull requests and issue submissions. For 10 repositories (31%) it was unclear how the data standard developers wanted users to submit feedback.

We found 208 unique labels being used to track and prioritize user-submitted issues across all 32 repositories. The maximum number of labels used by a single repository was 28, while nine repositories did not add any additional labels beyond the default set provided by GitHub. The most frequently used custom terms included in the labels were "priority" (n = 13), "class" (n = 7), "docs" (n = 8), and "term" (n = 6). These labels were often paired with other words that gave the issue label additional context (e.g., high-priority or new-term-request).

3.3 User-facing websites

Eleven repositories managed and displayed their data standards only on GitHub (e.g., https://github.com/ESIPFed/science-on-schema.org/). The rest of the standards (n = 21) were hosted on other websites in addition to GitHub. Most often, repository managers used GitHub Pages (https://pages.github.com/) to build a project website (n = 18) to mirror some or all of their documents and templates on a separate site (e.g., https://www.odm2.org/). The remaining three

repositories built separate HTML-based sites (e.g., https://environmentaldatainitiative.org/) to display a subset of files hosted on their GitHub repositories.

3.4 Documenting ESS-DIVE's data reporting formats on GitHub and on ESS-DIVE

We used the practices identified in our systematic review to create our ESS-DIVE Community Space on GitHub (https://github.com/ess-dive-community), where six teams of scientists are developing and managing data and metadata reporting formats (e.g., Bond-Lamberty et al., 2021; Damerow et al., 2021; Ely et al., 2021). We note that initial drafts of documentation were created and reviewed using other collaborative cloud-based tools (e.g., Google Sheets), then migrated to GitHub for community feedback.

To facilitate uploading data standard documentation to GitHub, we created README and GitHub Issue templates, complete with written prompts for content based on the findings of our systematic review (Templates are available for download in Crystal-Ornelas et al., 2021b). For example, all README files include a "How to Contribute" heading where each repository can link to the GitHub issue templates that help organize user feedback. In general, repositories begin with a flat file directory (i.e., with no subfolders) and then folders are created if a reporting format has several files of the same type (e.g., templates, images) within the repository. We use the documentation tool GitBook to render our GitHub repositories as project websites (e.g., https://ess-dive.gitbook.io/continuous-soil-respiration-reporting-format/). Content displayed on GitBooks are automatically updated with the most recent version of documents on GitHub, lessening the burden of keeping track of documents across multiple platforms for our repository managers.

When reporting formats are finalized, we use GitHub's "Release" feature to tag updates to the data standard documents with the semantic versioning schema MAJOR.MINOR.PATCH (Figure 3). The version numbers (e.g., v1.0.1) are assigned according to whether the changes to the formats are forward compatible, backwards compatible, or typo fixes, respectively. Once documents are tagged with a version number in GitHub, the documents can be easily

downloaded and then archived in ESS-DIVE. The archived data package is issued a DOI and the resulting citation is manually updated in the GitHub repository README file.

4 Discussion

4.1 Recommendations for using GitHub to develop data standards

Community-led data and metadata standard development and adoption require agreement across communities of researchers that work together to discuss, test, and update documentation. Based on the results of our systematic review and in light of well-established GitHub best practices from the coding community described in our introduction, we outline our recommendations for version control of data standard documentation using GitHub (Figure 3). Incorporating these recommendations may improve the usability of community-developed data standards, especially to engage scientists who are unfamiliar with GitHub yet essential for their contributions to the ongoing adoption of data standards. We note that many of our recommendations can be tracked using GitHub's "community profile" checklist that is built-in to every repository (e.g., https://github.com/ess-dive-community/essdive-leaf-gas-exchange/community).

4.1.1 File types

First, data standard developers need to decide which file types they will upload to GitHub to describe their data standards. The four main file types to choose from are binary files (e.g., Excel spreadsheets or Word documents), csv files, markdown files, and JSON or YAML files. One consideration when deciding which files to upload, is that the GitHub user interface does not allow users to easily view changes (called GitHub diffs) between versions of some of the more human-readable file formats (e.g., Excel spreadsheets or column data such as CSV files). Markdown files have the benefit of being relatively easy to modify within the GitHub user interface, and changes to markdown files can be easily tracked using GitHub diffs. Lastly, changes to JSON and YAML files will be shown clearly in GitHub diffs and can be incorporated into GitHub validation tools, but the learning curve to becoming familiar with these formats is steeper than all other file types.

4.1.2 README files

Next, we recommend that GitHub repositories contain, at a minimum, a detailed README file in the repository root, in addition to domain-specific documentation for the data and metadata standards. This README file should include the following subheadings to organize content and

support first-time users: “About”, “Getting Started”, “How to Contribute”, “License”, “Funding and Acknowledgments”, and “Recommended Citation”. Without this critical information, it is unclear how and whether data standards are able to be reused by scientists (README file template available in Crystal-Ornelas et al., 2021b).

4.1.3 Licensing

In order to facilitate collaboration within a repository, we recommend that each repository include an open-source usage license. When first initializing a GitHub repository, users can choose from a set of usage licenses that can be autogenerated as part of the repository set-up process, and then modified to suit user needs. GitHub has also created a website where users can search for and select open-source licenses: https://choosealicense.com/.

4.1.4 Versioning

We recommend semantic versioning (e.g., v1.0.1) be used to track updates to version-controlled data standard documents (Preston-Werner, 2020). By using the built-in GitHub “Release” feature, repository managers can save a snapshot of their GitHub repository at a point in time and assign a version number that aligns with semantic version conventions (Figure 3). Clear version numbers enable users to identify when they need to migrate their data to an updated standard or locate and download previous versions of the documentation. When repository managers choose to publish a “Release”, we also recommend that they archive their data standard documents in a long-term data repository. If no domain-specific archive exists, then GitHub’s integration with Zenodo can be used to instantly archive GitHub repository content and also generate a recommended citation. Some data standard developers may choose to version terms and vocabularies used within their data standard, separately from the standard itself (e.g., https://github.com/tdwg/vocab/blob/master/vms/maintenance-specification.md or https://cfconventions.org/standard_name_rules.html). Decoupling vocabulary and data standard versioning can be an effective way to communicate with users when different aspects of the standard change (e.g., specific vocabulary terms vs. supporting documentation). As community data standards are used, tested, and feedback is generated, developers should be prepared for standards to be updated and changed over time. In addition to the semantic versioning described above, we recommend that changes to data standards be documented using one or more of the following approaches: listing the latest updates in the repository’s README file, describing changes in local commits, providing pull request descriptions, referencing issue numbers in

commit or pull request messages, or creating a GitHub changelog to provide details on data standard updates.

4.1.5 Issues and contributions

We strongly recommend that collaborative development of community data standards take place through GitHub issues and pull requests rather than by other personal communications so that decisions and revisions are tracked over time, and publicly documented within a repository. Repository managers can create issue and pull requests templates to help structure user-submitted comments and edits. If data standard developers would like to include detailed contributing guidelines, we suggest creating CONTRIBUTING.md files in the root directory, which will also be indexed by the "community profile" checklist mentioned above, and linked on each issue template (https://docs.github.com/en/communities/setting-up-your-project-for-healthy-contributions/setting-guidelines-for-repository-contributors). We suggest that issues be categorized using either the set of built-in issue labels provided by GitHub or other labels specified by repository creators.

4.1.6 Documentation

We recommend that all GitHub repositories externally display the documentation on more general public-friendly project websites. In our review, we found that three of the repositories that built websites external to GitHub, did so using platforms that require manually updating content each time documents are updated. This type of long-term management of multiple documents across multiple web platforms is inefficient, error-prone, and unsustainable. Instead, we recommend using one of the many website-building platforms that seamlessly integrate with GitHub to mirror repository content on webpages and retrieve updates to documents automatically (e.g., GitHub Pages, GitBook, netlify). For data standard developers creating machine-readable standards (e.g., CSV, JSON, or YAML files), many of the website building platforms can display these machine readable formats in more human readable tables (e.g., https://github.com/tdwg/camtrap-dp/blob/main/_layouts/tables.html). Website building and updating is just one of the many tasks that can be automated using a feature called "GitHub Actions" (https://docs.github.com/en/actions). Advanced GitHub users may also consider using

automated GitHub Actions to welcome contributors to their projects on GitHub or validate contributions from the user community.

4.2 Challenges and future directions

There are three main challenges to using GitHub as the primary platform for collaboration on data and metadata standard documentation. First, by design Git and thus GitHub does not support real-time collaboration on cloud-based files (e.g., google sheets). For the six teams developing data reporting formats with ESS-DIVE, the initial documents were generally drafted in word processing or spreadsheet tools before being uploaded to GitHub. The benefit of this approach is that many contributors, some unfamiliar with Git and GitHub, could directly edit documents and suggest changes. However, it means that the earliest phases of data standard development occurred outside of the VCS platform. One solution is to save the collaborative spreadsheet as a CSV file and when updates are made, upload the CSV to the GitHub repository, tag a new release of the data standard on GitHub, and close the related GitHub issue through a commit message.

A second, albeit relatively minor, limitation is that file types that are commonly used to create data standards (e.g., Excel spreadsheets, Word documents, and even CSV files) are not easy to edit within the GitHub user interface—either because they are proprietary binary formats (Excel/Word) or columnar by nature (CSV). Computer code and markdown files are, in comparison, easy to edit within GitHub and produce human readable GitHub diffs. However, we note that if computer code or markdown files have hundreds of lines of changes between versions, users may want to view GitHub diffs using the desktop (https://desktop.github.com/) rather than the website version of GitHub. Binary files like word documents or spreadsheets must be edited offline, and then updated within a GitHub repository. This may deter contributions from users that want to view and edit documents in one location.

The third and perhaps most important challenge is related to the sometimes steep learning curve that must be overcome for scientists to feel comfortable and/or motivated to engage with content on GitHub (Isomöttönen & Cochez, 2014). Although GitHub and other organizations have developed educational tutorials geared toward first-time users (e.g., https://lab.github.com/ or Openscapes, 2021), VCS platforms in general, and GitHub specifically, remain focused on a programming user base. Thus, despite new user-friendly improvements, the GitHub learning

curve can be steep for some non-computational researchers. Features like 'GitHub Conversations' and in line commenting are examples where changes to GitHub's user interface can make somewhat complicated tasks (i.e., reviewing pull requests) more approachable. Moreover, creating project websites (i.e., using platforms like GitBook) let users unfamiliar with GitHub interact with documents through more human readable websites. However, a key feature of version control is change logs that let visitors see how version-controlled content changes over time, and these change logs are not exposed through the GitBook web interface.

Our focus on community engagement in standards development means that our systematic review did not explicitly consider the machine-readability of data and metadata standards documented on GitHub. Although machine-readability is the ultimate target, intermediate human-readable standards may be required to lower the barrier for standards adoption when developing meaningful formats and guidelines for metadata and data. Machine-readable standards are a cornerstone of FAIR data (Wilkinson et al., 2016), and multi-disciplinary teams of domain scientists, informaticists, and computer programmers are critical to bridging the gap between human and machine-readable standards. Indeed, there are templates used by some data standard developers to render machine-readable standards into human-readable templates (e.g., https://github.com/tdwg/camtrap-dp/blob/main/_layouts/tables.html rendered as https://tdwg.github.io/camtrap-dp/data/#deployments). As we move toward engaging the broader research community in adopting community-developed data standards, we see GitHub and autogenerated project websites as the platforms for responding to user feedback, posting tutorials for using standards, and versioning our supporting documents in response to the user community.

5 Conclusion

Community-developed data standards and reporting formats are a key step toward making data FAIR. VCS platforms can enable collaboration on documentation during and after the standards development process. However, our systematic review found that GitHub, and more broadly VCS platforms, are generally underused for collaboration on data and metadata standard development (30% of all standards were hosted on GitHub, and only one standard used BitBucket out of 108 reviewed). Even among the GitHub repositories, many do not use important tools for collaboration such as issue templates, issue labels, licensing information, and hosting content on project websites autogenerated from GitHub content that can enable

community discussion and feedback for improving the standards. At ESS-DIVE, we have used GitHub to enhance the development of our community data and metadata reporting formats, using the systematic review described in this paper to guide the structure and content of the ESS-DIVE Community Space on GitHub. The recommendations on VCS structure we outline here can be used by researchers developing data standards or reporting formats looking for greater community involvement in data stewardship.

Acknowledgments

RCO, CV, SC, VH, JD, MB, ZK, ER, MS, and DA were funded through the ESS-DIVE repository by the U.S. DOE's Office of Science Biological and Environmental Research under contract number DE-AC02-05CH11231. KSE and AR were supported through the US Department of Energy contract number DE-SC0012704 to Brookhaven National Laboratory. Reporting format development was supported by ESS-DIVE's Community Funds, through the Office of Biological and Environmental Research in the Department of Energy, Office of Science. The authors thank Dr. Chelle Gentemann, Dr. Peter Desmet, and one anonymous reviewer for their insightful comments on our manuscript.

Data availability statement

The code used in this analysis is available at the following GitHub repository (https://github.com/ess-dive-community/essdive-github-systematic-review).
Data (Crystal-Ornelas et al., 2021a) and example templates (Crystal-Ornelas et al., 2021b) associated with the manuscript have been archived in the ESS-DIVE data repository.

Conflicts of interest

The authors have no conflicts of interest to declare.

Author contributions

Conceptualization: RCO, CV

Data curation: RCO

Formal Analysis: RCO

Funding Acquisition: DA, CV

Project Administration: DA

Writing - original draft: RCO, CV, JD

Writing - review and editing: RCO, CV, BBL, KB, SC, MC, RD, KSE, AG, SH, VH, JD, SP, MB, ZK, ER, MS, AR, TV, HW, PW, JNW, DA

Tables and Figures

Table 1. Definitions and examples of typical terms related to data standardization.

Term	Definition	References for definition	Examples	Reference for example
Mark-up language / specification	Emphasizes machine readability and data exchange. Mark-up languages/specifications may include tools to facilitate conversion to machine-readable code, but primarily they are intended to make data or metadata machine readable.	(Jones et al., 2019; Spellman et al., 2002; Yilmaz et al., 2011)	Ecological Metadata Language; Minimum information about any sequences	(Jones et al., 2019; Yilmaz et al., 2011)
Ontology	Defined vocabularies where each term has a persistent identifier, explicit definition and documented relationships between terms. Ontologies can be separate from or within standards, reporting formats, or other data guidelines.	(Buttigieg et al., 2013; DiGiuseppe et al., 2014; Raskin et al., 2006)	Environmental Ontology (ENVO)	(Buttigieg et al., 2013)
Reporting format	Guidelines for formatting data developed by a community of researchers. These formats are generally more easily developed, adopted, and modified over time to be responsive to changing data needs. These are not governed or accredited by formal committees.	(Sansone et al., 2019)	Leaf-level Gas Exchange	(Bond-Lamberty et al., 2021; Damerow et al., 2021; Ely et al., 2021)

Schemas	Provide machine-readable structure and relationships for unique data object identification.	(Flannery et al., 2009; S.-A. Sansone et al., 2020)	DataCite metadata schema; ICAT schema	(DataCite Metadata Working Group, 2019; ICAT Project, 2020)
Standard	Agreed-upon policies and procedures for representing data. Standards are typically accredited by a large governing body.	(US EPA, 2015; Yarmey & Baker, 2013)	ISO 8601; OGC GeoTIFF Standard	(ISO, 2019)

Table 2. Categories of content included in GitHub repositories that document data standards.

GitHub repository and README file elements	Definition	Example
About	An about section is generally several sentences long, appears at the top of a README.md file, and describes the purpose of the repository.	https://github.com/EnvironmentOntology/envo#the-environment-ontology
Citation	The permanent URL and/or DOI of the reporting format	https://github.com/dagendresen/darwincore-germplasm#citation
Contribute	GitHub repository managers will sometimes provide guidelines for contributing in the repository's README file or in a separate file typically named CONTRIBUTING.md	https://github.com/opengeospatial/weather-on-the-web#contributing
Funding	List of organizations financially supporting efforts	https://github.com/NCEAS/oboe#citation-and-credits

Getting started	This section provides visitors to repository with guidance on what they can find on GitHub page, and general folder structure	https://github.com/tdwg/dwc#getting-started
History	Paragraph describing development of reporting format or ontology	https://github.com/NCEAS/eml#history
License	License information describes parameters for use of material in GitHub repository.	https://github.com/EnvironmentOntology/environmental-exposure-ontology/blob/master/LICENSE.txt
Resources	Slack channel, wiki, other groups doing similar work	https://github.com/ESIPFed/science-on-schema.org//#resources
Terms	Repositories for data reporting formats and ontologies may have files that contain required or optional vocabularies	https://github.com/EcologicalTraitData/ETS/blob/master/ETS.csv
Use case guide	Example of how reporting format or ontology can be used	https://github.com/EcologicalTraitData/ETS/blob/master/bestpractice.Rmd

Version	Release information, often provided in semantic versioning's MAJOR.MINOR.PATCH format (https://semver.org/)	https://github.com/NCEAS/eml/releases
Visual structure	Flow-chart or image file (JPG, PNG) depicting repository directory structure	https://github.com/tdwg/dwc#repo-structure

Table 3. Standardized set of terms we used to characterize the methods for collaboration identified across all GitHub repositories documenting data standards.

Terms related to collaboration	Definition	Example
Code of conduct	Document that provides guidelines for community behavior within a GitHub repository	https://github.com/cf-convention/cf-conventions/blob/master/CODE_OF_CONDUCT.md
CONTRIBUTING.md	Document providing detailed guidance on how repository managers would like users to contribute to the project	https://github.com/tdwg/dwc/blob/master/.github/CONTRIBUTING.md
GitHub tutorial	General overview of version control using git	https://github.com/ESIPFed/sweet/blob/master/CONTRIBUTING.md#how-to-work-with-us-on-github-using-git-command-line
Instructions for branching	If repositories encourage users to submit suggested changes through pull requests are their instructions for branching.	https://github.com/opengeospatial/weather-on-the-web/wiki/Propose-a-change-to-a-draft-wow-specification-document
Issue templates	Repository manager can provide templates for users submitting GitHub issues	https://github.com/tdwg/dwc/issues/new/choose

Mention contributing	Statement of how visitors can contribute to content within the GitHub repository.	https://github.com/GenomicsStandardsConsortium/mixs#purpose
Outline of workflow	A clear process by which repository managers review feedback and incorporate changes.	https://github.com/EnvironmentOntology/envo/wiki/Adding-classes-to-ENVO

Table 4. The data standards identified in our systematic review were hosted on the internet using a variety of platforms including Version Control Systems, general purpose websites, and long-term data repositories. Sample sizes indicate the number of data standards hosted on each of the platforms.

Platform for displaying data standards	Purpose	Allows collaboration?	Long-term data repository?
Agroportal (n = 1)	Data repository	X	✓
Bioportal (n = 2)	Data repository used for conveying information	X	✓
Bitbucket (n = 1)	Displaying information and providing opportunity for collaboration	✓	X
GitHub (n = 60)	Displaying data standard content and collaborating on files through version control	✓	X
INRAE (n = 1)	Data repository	X	✓
Journal article (n = 2)	Proposing usage of data standard	X	X
Organization's own website (n = 42)	Typically hosting content in various file formats (PDF, XML, RDF, CSV)	X	X

Figure 1. (A) In our analysis of the content across 32 GitHub repositories we found that greater than 90% of the repositories included both an 'about' section to describe the repository and a 'terms' list that defines essential elements of the data standard. Relatively few repositories included a table of contents in the form of a 'getting started' section (34%). Even fewer provided recommended citations for their work (31%) or funder information (22%). (B) All GitHub repositories in our systematic review contained a README file. Content of the README pages varied, but most contained an 'about' section that described the data standard. Licensing information, suggested citations, and versioning details were described in approximately 30% of README pages.

Figure 2. Just over half of the repositories in our review (56%) mention contributing to their data standards on GitHub. Fewer (n = 10) provide details on the multi-step process often involved in reviewing suggested changes to repository content all the way through publishing approved content.

Figure 3. Visual depiction of the final recommendations for version-controlled data standard documentation, how they should be versioned, and where they should be archived or hosted for reuse.

References

Angulo, M. A., & Aktunc, O. (2019). Using GitHub as a teaching tool for programming courses. In *ASEE Gulf-Southwest Section Annual Meeting 2018 Papers*. American Society for Engineering Education. Retrieved from https://repositories.lib.utexas.edu/bitstream/handle/2152/79922/using-github-as-a-teaching-tool-for-programming-courses.pdf?sequence=2

Bezuidenhout, L. (2020). Being Fair about the Design of FAIR Data Standards. *Digit. Gov.: Res. Pract.*, *1*(3), 1–7. https://doi.org/10.1145/3399632

Bissyandé, T. F., Lo, D., Jiang, L., Réveillère, L., Klein, J., & Traon, Y. L. (2013). Got issues? Who cares about it? A large scale investigation of issue trackers from GitHub. In *2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE)* (pp. 188–197). https://doi.org/10.1109/ISSRE.2013.6698918

Bond-Lamberty, B., Christianson, D. S., Crystal-Ornelas, R., Mathes, K., & Pennington, S. C. (2021). A reporting format for field measurements of soil respiration. *Ecological Informatics*. https://doi.org/10.1016/j.ecoinf.2021.101280

Buttigieg, P. L., Morrison, N., Smith, B., Mungall, C. J., Lewis, S. E., & ENVO Consortium. (2013). The environment ontology: contextualising biological and biomedical entities. *Journal of Biomedical Semantics*, *4*(1), 43. https://doi.org/10.1186/2041-1480-4-43

Crystal-Ornelas, R., Varadharajan, C., Bond-Lamberty, B., Boye, K., Cholia, S., Crow, M., et al. (2021a). Data from: "A guide to using Version Control Systems for developing and versioning data standards and reporting formats [Data set]. *Environmental Systems Science Data Infrastructure for a Virtual Ecosystem*. https://doi.org/10.15485/1780565

Crystal-Ornelas, R., Varadharajan, C., Bond-Lamberty, B., Boye, K., Cholia, S., Crow, M., et al. (2021b). Templates for developing and versioning data standards and reporting formats

using Version Control Systems [Data set]. *Environmental Systems Science Data Infrastructure for a Virtual Ecosystem*. https://doi.org/10.15485/1780564

Damerow, J., Varadharajan, C., Boye, K., Brodie, E. L., Burrus, M., Chadwick, D. K., et al. (2021). Sample identifiers and metadata to support data management and reuse in multidisciplinary ecosystem sciences. *Data Science Journal*, *20*, 1–19. https://doi.org/10.5334/dsj-2021-011

DataCite Metadata Working Group. (2019). DataCite Metadata Schema for the Publication and Citation of Research Data. Version 4.3, *DataCite*. https://doi.org/10.14454/f2wp-s162

De Pooter, D., Appeltans, W., Bailly, N., Bristol, S., Deneudt, K., Eliezer, M., et al. (2017). Toward a new data standard for combined marine biological and environmental datasets - expanding OBIS beyond species occurrences. *Biodiversity Data Journal*, *5*, e10989. https://doi.org/10.3897/BDJ.5.e10989

DiGiuseppe, N., Pouchard, L. C., & Noy, N. F. (2014). SWEET ontology coverage for earth system sciences. *Earth Science Informatics*, *7*(4), 249–264. https://doi.org/10.1007/s12145-013-0143-1

Ely, K. S., Rogers, A., Agarwal, D. A., Ainsworth, E. A., Albert, L., Ali, A., et al. (2021). A reporting format for leaf-level gas exchange data and metadata. *Ecological Informatics*, 101232. https://doi.org/10.1016/j.ecoinf.2021.101232

Flannery, D., Matthews, B., Griffin, T., Bicarregui, J., Gleaves, M., Lerusse, L., et al. (2009). ICAT: Integrating Data Infrastructure for Facilities Based Science. In *2009 Fifth IEEE International Conference on e-Science* (pp. 201–207). https://doi.org/10.1109/e-Science.2009.36

Galdzicki, M., Clancy, K. P., Oberortner, E., Pocock, M., Quinn, J. Y., Rodriguez, C. A., et al. (2014). The Synthetic Biology Open Language (SBOL) provides a community standard

for communicating designs in synthetic biology. *Nature Biotechnology*, *32*(6), 545–550. https://doi.org/10.1038/nbt.2891

GitHub. (2020a). *About READMEs*. Retrieved from https://docs.github.com/en/free-pro-team@latest/github/creating-cloning-and-archiving-repositories/about-readmes

GitHub. (2020b). *Empowering Healthy Communities*.

Hart, E. M., Barmby, P., LeBauer, D., Michonneau, F., Mount, S., Mulrooney, P., et al. (2016). Ten Simple Rules for Digital Data Storage. *PLoS Computational Biology*, *12*(10), e1005097. https://doi.org/10.1371/journal.pcbi.1005097

ICAT Project. (2020). ICAT Schema (v4.10.0). Retrieved from https://repo.icatproject.org/site/icat/server/4.10.0/schema.html

ISO. (2019). Date and Time Format (ISO Standard Number 8601-1:2019). Retrieved from https://www.iso.org/iso-8601-date-and-time-format.html

Isomöttönen, V., & Cochez, M. (2014). Challenges and Confusions in Learning Version Control with Git. In *Information and Communication Technologies in Education, Research, and Industrial Applications* (pp. 178–193). Springer International Publishing. https://doi.org/10.1007/978-3-319-13206-8_9

Jones, M., O'Brien, M., Mecum, B., Boettiger, C., Schildhauer, M., Maier, M., et al. (2019). *Ecological Metadata Language version 2.2.0*. KNB Data Repository. https://doi.org/10.5063/F11834T2

Lee, G., Bacon, S., Bush, I., Fortunato, L., Gavaghan, D., Lestang, T., et al. (2021). Barely sufficient practices in scientific computing. *Patterns (New York, N.Y.)*, *2*(2), 100206. https://doi.org/10.1016/j.patter.2021.100206

Mergel, I. (2015). Open collaboration in the public sector: The case of social coding on GitHub. *Government Information Quarterly*, *32*(4), 464–472. https://doi.org/10.1016/j.giq.2015.09.004

Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., & Group, P. (2009). Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *PLoS Medicine*, *6*(7), e1000097.

Openscapes. (2021, March 29). Openscapes Champions Lesson Series. Retrieved from https://openscapes.github.io/series/index.html

Pasquetto, I. V., Randles, B. M., & Borgman, C. L. (2017). On the Reuse of Scientific Data. *Data Science Journal*, *16*, 8. https://doi.org/10.5334/dsj-2017-008

Perkel, J. (2016). Democratic databases: science on GitHub. *Nature*, *538*(7623), 127–128. https://doi.org/10.1038/538127a

Preston-Werner, T. (2020). Semantic Versioning 2.0.0. Retrieved November 6, 2020, from https://semver.org/

Python Software Foundation. (2020). Python Language Reference (Version 3.9.1). Retrieved from https://www.python.org/

Raskin, R., Pan, M., & Mattmann, C. (2006). Enabling Semantic Interoperability for Earth Science Data. In *Proceedings of the Second IEEE International Conference on e-Science and Grid Computing*.

Read, K., Creamer, A. T., Kafel, D., Vander Hart, R. J., & Martin, E. R. (2013). Building an eScience Thesaurus for Librarians: A Collaboration Between the National Network of Libraries of Medicine, New England Region and an Associate Fellow at the National Library of Medicine. *Journal of EScience Librarianship*, *2*(2), 7. https://doi.org/10.7191/jeslib.2013.1049

Roche, D. G., Kruuk, L. E. B., Lanfear, R., & Binning, S. A. (2015). Public data archiving in ecology and evolution: how well are we doing? *PLoS Biology*, *13*(11), e1002295.

Sansone, S. A., McQuilton, P., Rocca-Serra, P., Gonzalez-Beltran, A., Izzo, M., Lister, A. L., et al. (2019). FAIRsharing as a community approach to standards, repositories and policies. *Nature Biotechnology*, *37*(4), 358–367. https://doi.org/10.1038/s41587-019-0080-8

Sansone, S.-A., McQuilton, P., Rocca-Serra, P., Gonzalez-Beltran, A., Izzo, M., & AL and Thurston M, L. (2020). FAIRsharing.org. Retrieved February 1, 2021, from https://fairsharing.org

Schneider, F. D., Fichtmueller, D., Gossner, M. M., Güntsch, A., Jochum, M., König- Ries, B., et al. (2019). Towards an ecological trait- data standard. *Methods in Ecology and Evolution / British Ecological Society*, *10*(12), 2006–2019. https://doi.org/10.1111/2041-210X.13288

Sholler, D., Steinmacher, I., Ford, D., Averick, M., Hoye, M., & Wilson, G. (2019). Ten simple rules for helping newcomers become contributors to open projects. PLoS Computational Biology, 15(9), e1007296.

Spellman, P. T., Miller, M., Stewart, J., Troup, C., Sarkans, U., Chervitz, S., et al. (2002). Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biology*, *3*(9), RESEARCH0046. https://doi.org/10.1186/gb-2002-3-9-research0046

Stoudt, S., Vásquez, V. N., & Martinez, C. C. (2021). Principles for data analysis workflows. *PLoS Computational Biology*, *17*(3), e1008770. https://doi.org/10.1371/journal.pcbi.1008770

Tantisuwankul, J., Nugroho, Y. S., Kula, R. G., Hata, H., Rungsawang, A., Leelaprute, P., & Matsumoto, K. (2019). A Topological Analysis of Communication Channels for

Knowledge Sharing in Contemporary GitHub Projects. *The Journal of Systems and Software, 158,* 110416.

US EPA. (2015, June 11). Data Standards. Retrieved January 25, 2021, from https://www.epa.gov/data-standards/learn-about-data-standards

Varadharajan, C., Cholia, S., Snavely, C., Hendrix, V., Procopiu, C., Swantek, D., et al. (2019). Launching an Accessible Archive of Environmental Data. *Eos , 100*. Retrieved from https://eos.org/science-updates/launching-an-accessible-archive-of-environmental-data

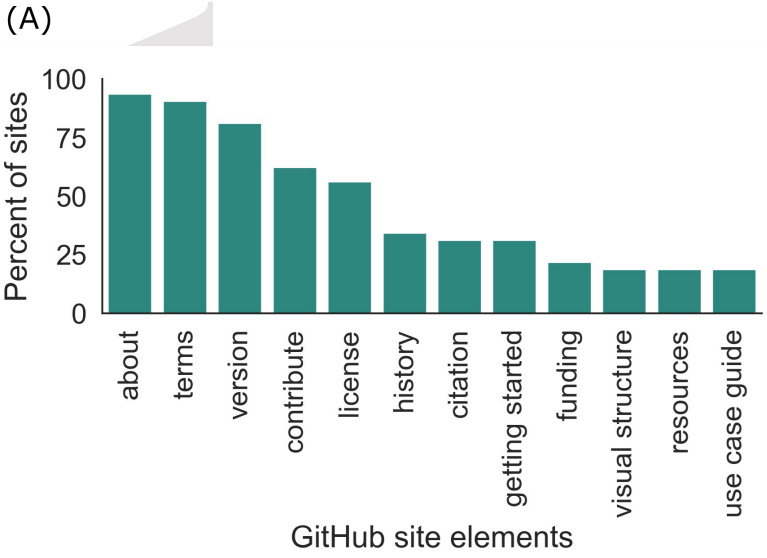
Wieczorek, J., Bloom, D., Guralnick, R., Blum, S., Döring, M., Giovanni, R., et al. (2012). Darwin Core: An Evolving Community-Developed Biodiversity Data Standard. PloS One, 7(1), e29715.

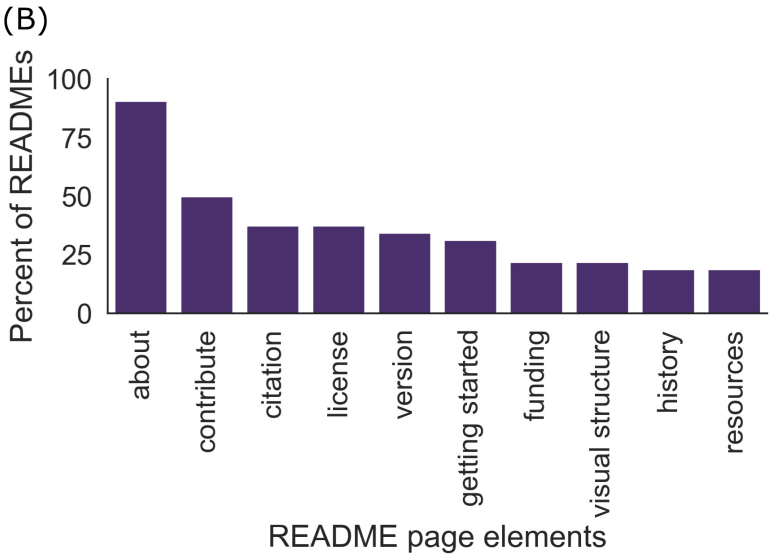
Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, *3*(1), 160018. https://doi.org/10.1038/sdata.2016.18

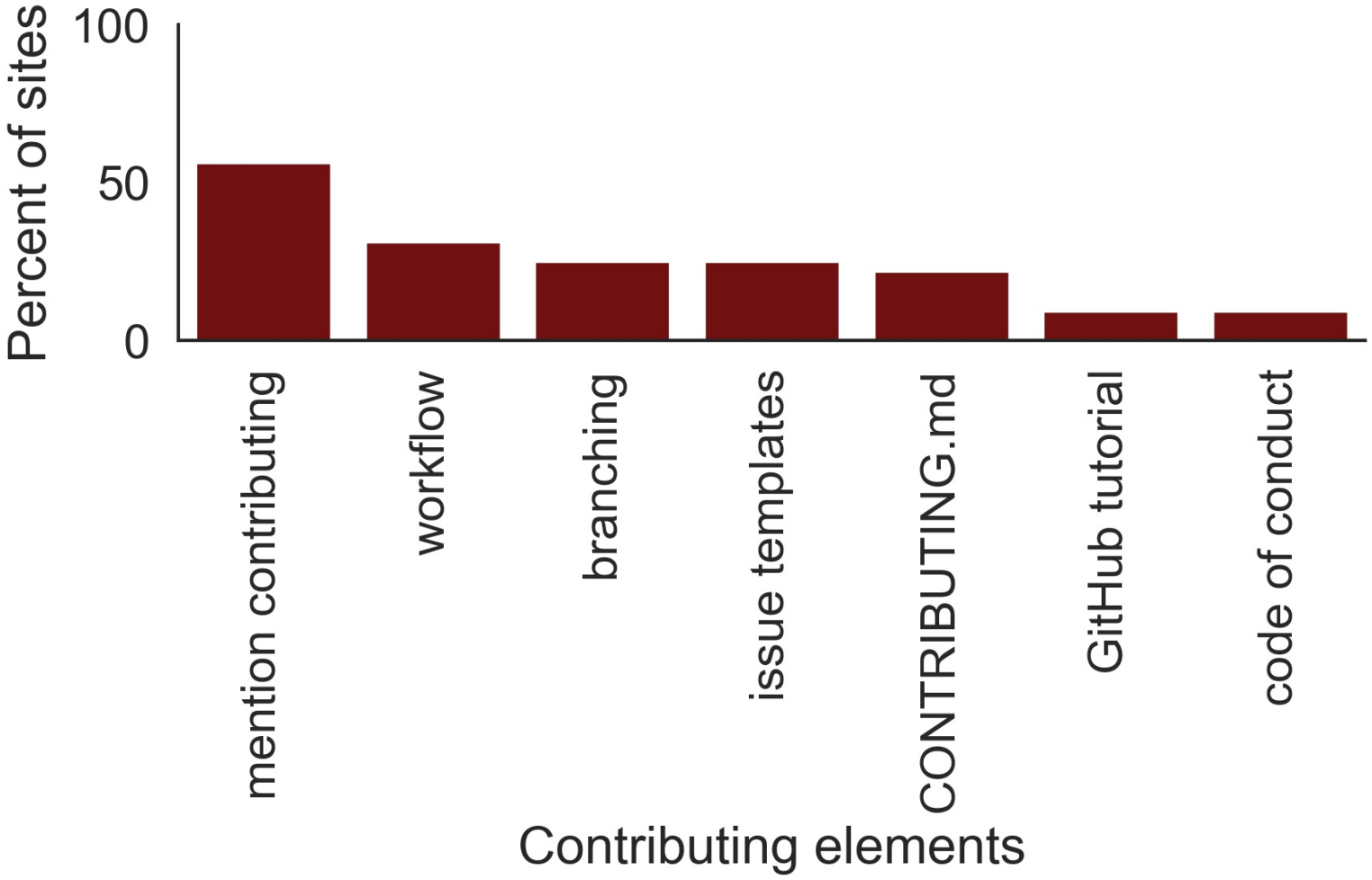
Yarmey, L., & Baker, K. (2013). Towards Standardization: A Participatory Framework for Scientific Standard-Making. *The International Journal of Digital Curation*, *8*(1). https://doi.org/10.2218/ijdc.v252

Yilmaz, P., Kottmann, R., Field, D., Knight, R., Cole, J. R., Amaral-Zettler, L., et al. (2011). Minimum information about a marker gene sequence (MIMARKS) and minimum information about any (x) sequence (MIxS) specifications. *Nature Biotechnology*, *29*(5), 415–420. https://doi.org/10.1038/nbt.1823

Zimmerman, A. S. (2008). New Knowledge from Old Data: The Role of Standards in the Sharing and Reuse of Ecological Data. *Science, Technology & Human Values*, *33*(5), 631–652. https://doi.org/10.1177/0162243907306704







GitHub Repository

README

License

Issue templates

Issue tags

Versioning data standards

X . Y . Z

Major . Minor . Patch

Documents updated, new version issued (e.g., v1.0.1)

Archiving and displaying

Version controlled documents stored in long-term archive

Updates pushed to user-facing website