The Geological Society of America
Special Paper 558

Revealing Earth science code and data-use practices using the Throughput Graph Database

Andrea K. Thomer*
University of Arizona, School of Information, P.O. Box 210076, Harvill Building, Tucson, Arizona 85721, USA

Morgan F. Wofford*
Michael C. Lenard*
University of Michigan, School of Information, 105 S. State Street, Ann Arbor, Michigan 48109, USA

Socorro Dominguez Vidana*
Data Scientist, Vancouver, British Columbia, Canada

Simon J. Goring*
University of Wisconsin–Madison, Department of Geography, Science Hall, 550 N. Park Street, Madison, Wisconsin 53706, USA

ABSTRACT

The increased use of complex programmatic workflows and open data within the Earth sciences has led to an increase in the need to find and reuse code, whether as examples, templates, or code snippets that can be used across projects. The "Throughput Graph Database" project offers a platform for discovery that links research objects by using structured annotations. Throughput was initially populated by scraping GitHub for code repositories that reference the names or URLs of data archives listed on the Registry of Research Data Repositories (https://re3data.org). Throughput annotations link the research data archives to public code repositories, which makes data-relevant code repositories easier to find.

Linking code repositories in a queryable, machine-readable way is only the first step to improving discoverability. A better understanding of the ways in which data is used and reused in code repositories is needed to better support code reuse. In this paper, we examine the data practices of Earth science data reusers through a classification of GitHub repositories that reference geology and paleontology data archives. A typology of seven reuse classes was developed to describe how data were used within a code repository, and it was applied to a subset of 129 public code repositories on GitHub. Code repositories could have multiple typology assignments. Data use for Software Development dominated (n = 44), followed by Miscellaneous Links to Data Archives (n = 41), Analysis (n = 22), and Educational (n = 20) uses. GitHub repository features show some relationships to the assigned typologies, which indicates that these characteristics may be leveraged to systematically predict a code repository's category or discover potentially useful code repositories for certain data archives.

*Emails: athomer@arizona.edu; mwofford@umich.edu; mclenard@umich.edu; sedv8808@gmail.com; goring@wisc.edu

Thomer, A.K., Wofford, M.F., Lenard, M.C., Dominguez Vidana, S., and Goring, S.J., 2023, Revealing Earth science code and data-use practices using the Throughput Graph Database, *in* Ma, X., Mookerjee, M., Hsu, L., and Hills, D., eds., Recent Advancement in Geoinformatics and Data Science: Geological Society of America Special Paper 558, p. 147–159, https://doi.org/10.1130/2022.2558(10).

INTRODUCTION

Data in the Earth sciences have never been more open. Yet, this openness presents new obstacles for using data. The sheer scale and volume of open Earth science data often requires programmatic access to data archives (via an Application Program Interface [API]), complex computational pipelines, and new methods of making research reproducible (Belhajjame et al., 2012a; Goble and De Roure, 2009; Hey, 2009). Code publication and open source software development have been proposed as ways to support Earth science data access and analysis, but while there are numerous repositories meant to facilitate code sharing, finding relevant code for a new research project is not always easy (Stall et al., 2018). Code is scattered through a wide range of platforms, including general purpose data archives such as Figshare, Dryad, or Pangaea; code-specific platforms such as BitBucket, GitLab, or GitHub; paper supplements and appendices; and personal webpages and blogs. Additionally, standards for software sharing and citation have lagged behind the need for software discovery (Barnes, 2010; Du et al., 2021; Howison and Bullard, 2016), which renders code harder to find and means that individuals who write and share code may not get the credit they deserve for their work.

The fragmentation of resources makes code reuse challenging, particularly for researchers without extensive programming skills, students still developing programming skills, and researchers conducting work in a new discipline. Information organization has great power to shape scholarly communication and practices (Thomer and Wickett, 2020). Data reuse, including the use of data archives, often requires significant first-hand, tacit knowledge of a data set and its context of production or provenance (Pasquetto et al., 2019; Zimmerman, 2007, 2008). Researchers who are not well versed in a particular discipline or specific data ecosystem are at a disadvantage when reusing data. This increases the risk of marginalized researchers and students being left behind as Earth scientists increasingly incorporate complex programmatic workflows, computation, and open data into their work. In other sciences that have become more computational, gender disparities tend to worsen. For instance, women comprise over 50% of those working in the field of biology but only 20% in computer science; computational biology is somewhere in between (Bonham and Stefan, 2017). Diversity across a number of metrics is low in the Earth sciences (Berhe and Ghezzehei, 2021), and proactive steps must be taken to ensure that structural shifts in the discipline do not result in greater inequity.

There is a clear need to improve the discoverability of code and data resources in the Earth sciences. We see particular potential in infrastructure to improve associations between code repositories and the data archives they query. Code repositories (like those on GitHub: https://github.com) are one of the primary ways that code is shared or published. However, GitHub's search is not tailored to Earth science use cases, and it is largely a plain text search tool rather than semantic. By semantically "linking" code repositories to the data archives they query and augmenting those links with Earth science–specific metadata, we can greatly improve discoverability and usability.

Improving the discoverability of code is the primary goal of the Throughput database project. Throughput links existing, but scattered, resources via an annotation graph and thereby reduces the "time to science." Throughput is a sort of meta-database with a focus on supporting interdisciplinary work and that of novice researchers. It stores information about other data sets and databases and is populated by annotations that can be used to link data sets to other relevant material, provide contexts or corrections, and otherwise help document the often tacit knowledge needed to reuse data.

This chapter begins with a brief overview of the obstacles to effectively reusing code in the sciences. We then describe our efforts to make code more discoverable via (1) the Throughput database and (2) the development of use-based metadata (as coined by Lynnes et al., 2020) to make code repositories easier to find within Throughput. Here, we define use-based metadata as *keywords that describe the purpose of a code repository with respect to a data archive or data from a data archive.* We develop a typology of the ways that GitHub developers use Earth science data archives. As one may expect, code repositories referencing Earth science data archives are most commonly used for software development or data analysis. However, we also find that a significant number of code repositories use data archives for educational purposes or simply include links to Earth science data archives. We conclude by describing our plans to apply these categories as metadata within the Throughput database: first, by manual annotation, and eventually, by automatic classification via machine learning approaches. Making data and code archives easier to find and use is an important next step for geoinformatics, and this paper is intended to facilitate that work. The use of Throughput as a tool to discover and thematically organize links between data and code repositories also serves a secondary purpose: to provide support for better software citation practices and to help data managers share emerging use cases for their data resources, and thereby increase their impact through links to existing code repositories.

CODE SHARING AND REUSE

Code publication is critical to supporting scientific reproducibility (Barnes, 2010; Davison, 2012; Ince et al., 2012; Peng, 2011; Stodden et al., 2013, 2016), particularly for complex computational pipelines that are challenging to reproduce independently (Belhajjame et al., 2012a; Goble and De Roure, 2009). Software citation guidelines (Fox et al., 2021; Katz and Chue Hong, 2018; Smith et al., 2016) and platforms such as Zenodo and the Open Science Framework are meant to make it easier to create a persistent reference to a codebase and therefore facilitate code reuse. Within the Earth sciences, there have been several efforts to facilitate code sharing and reuse, such as the NASA Earth Science Data Systems (ESDS) Software Reuse Portal (Downs et al., 2006; Gerard et al., 2007; Marshall et al., 2010).

Despite these efforts, the widespread reuse of code is still nascent, and challenges remain in making code findable, citable, and reusable. Even those at large centers such as NASA have found themselves re-developing similar software pipelines rather than using existing software (Mattmann et al., 2011); similar trends have been observed in highly computational fields such as bioinformatics (Duck et al., 2016). Despite the development of citation guidelines, software citations in papers remain diverse and unstandardized. They are variously cited within references, the main text of a paper, acknowledgments, or linked in supplemental materials (Du et al., 2021; Howison and Bullard, 2016). Additionally, code itself can be shared in a variety of ways. Platforms like GitHub, Bitbucket, and GitLab are used for collaborative development and sharing, though none of these platforms guarantees long-term archiving. Software journals, such as the *Journal of Open Source Software* (https://joss.theoj.org/), provide developers with a peer-reviewed space in which to document their contributions more fully and link to persistent code repositories. Some computational workflows can be captured and then shared in a re-executable manner via technologies such as Taverna and Kepler, which essentially record the transformations and computations performed on a data set so that they can be re-executed (Ludäscher et al., 2006), albeit with known limitations (Belhajjame et al., 2012b; Thomer et al., 2018).

Despite these challenges, code reuse is critical for the Earth sciences. Over the past several decades, significant time, funding, and labor have been invested in the development of Earth science data infrastructure. This includes data archives such as EarthChem, Pangaea, the Paleobiology Database, and more. Publication archives tailored for text mining were developed, such as GeoDeepDive (Peters et al., 2017). Data compilations emerged, such as COHMAP (COHMAP MEMBERS, 1988; Wright and Bartlein, 1993), MIOMAP (Carrasco et al., 2007), FAUNMAP (Graham and Lundelius, 1994), and eventually the Neotoma Paleoecology Database (Williams et al., 2018). Other developments include the creation of persistent identifier minting services such as IGSN (http://www.igsn.org/); the emergence of communities developing software, cyberinfrastructure, and interoperable data standards, such as LinkedEarth (http://linked.earth); and the iterative creation and refinement of best practices for data collection, analysis, and curation (Gil et al., 2016). Over the past decade in particular, initiatives such as EarthCube (https://earthcube.org) have resulted in considerable efforts to create data sharing infrastructures in the Earth sciences. Not only do these infrastructures represent incalculable hours of labor, but the data they contain represent an incomparable source of longitudinal observations about our planet and even beyond. Use and reuse of these data is simply not optional for those working in many fields.

Many of these infrastructures are best accessed through computational methods, such as calls to an online API, queries to a database, or through custom packages in R or Python. However, computational access isn't always straightforward, and the modes of accessing and integrating data vary wildly across data resources. The technical skill needed to access each of these resources presents a real barrier to participation in our field. Facilitating code reuse (and therefore facilitating access to other resources) is critical to increasing participation in the Earth sciences by those with less access to community support (Goring et al., 2020). Beyond technical skill, individuals close to a particular project, repository, or data set often have the most complete understanding of that resource and are in the best position to reuse content (Pasquetto et al., 2019). Reuse becomes more challenging with distance, which is measured in time or "social distance," between the originator and subsequent users. Distance creates a barrier to using and reusing data and resources for those outside of a project, institution, or cultural clique. We can improve the use of resources by those marginalized by existing "invisible colleges" (De Solla Price and Beaver, 1966) if we make the connections between data and analytic resources explicit, which effectively reduces the social distance between individuals and data creators.

Throughput: A New Approach to Facilitating Code and Data Reuse

With the Throughput database, we seek to ease the challenge of resource access by using a graph database to link research objects in the geosciences using structured annotations (a guiding use case for this project is described in Box 1). A key

Box 1. A Use Case for the Throughput Code Cookbook.

An early-career researcher is interested in understanding how extreme events in regional hydrology have affected soil development and ecosystem structure at multiple temporal scales. This interdisciplinary research project may require data from a paleoecological database, a hydrology resource, modern weather sensor networks, and a regional soils database. To ensure the analysis is FAIR (*sensu* Wilkinson et al., 2016), she may wish to directly download the data using an R script, transform spatial projections, and perform statistical analysis to understand how features are related over time.

The Throughput Database would let this researcher search for code repositories that link soils databases and hydrology data archives. The researcher can then use these repositories as a resource for accessing and transforming the data rather than reinventing each step in the workflow. This would allow her to make progress on her project more quickly—and could lead to a citation for the researcher who originally developed the code.

goal for Throughput is to link publicly available analytic code to data sets (and data archives), publications, websites, grants, and other user-contributed information. Throughput is designed to engage and support FAIR principles (Wilkinson et al., 2016). The database leverages the use of *findable* resources through web standards that make resources *accessible* by machine or human through an *interoperable* platform and web framework that supports a federated approach to provide a set of resources, including open source software and documentation, which support the *reuse* of project elements and concepts and support the reuse of data across the web by capturing richer metadata through annotations.

Throughput aims to address particular challenges in managing scientific workflows that often fall on early-career researchers, researchers at institutions that are not primary research facilities, and disciplinary groups with long-term funding that are managing data that lay outside core data archives. Challenges faced by these groups include (1) a lack of credit for data (or script) generation and reuse, (2) lack of technical knowledge around interdisciplinary workflows or new technical tools, (3) an inability to access contextual information for records with missing or incomplete metadata, and (4) lack of access to secondary data or analytic results associated with publications beyond an individual's core discipline or personal network.

Capturing relationships between distributed research products and data about these relationships is fundamental to resolving code and data reuse challenges. Throughput offers a means to connect and augment data objects and to manage relationships among objects, which supports the activities of data users and data generators. Users can access complete metadata around a particular data product by viewing textual annotations and examining related resources. Data generators, who may be managing complex, long-term projects, will be better able to manage data spread across various repositories (e.g., a project may include museum specimens, secondary fossil data, geochemical data, and ancient DNA records). One key demonstration case has been the annotation and linking of tephra data from the same volcanic event samples in SESAR (www.geosamples.org) with geochemistry data in EarthChem, field data in StraboSpot, or sediment core context in OpenCore, and with code used for laboratory data processing or computational models of volcanic eruptions or statistical matching of geochemistry (Kuehn et al., 2021). However, Throughput has broad capacity to create and surface links between research products.

The Throughput database has multiple potential user groups. In particular, the links between code repositories and data archives described here are intended to be of use to early-career researchers and interdisciplinary researchers working on data-intensive projects. The Code Cookbook (https://throughputdb.com) provides a searchable interface that allows individuals to discover data-leveraging code repositories by subject, keyword, or database name. By enriching repositories with rich annotations, we aim to transform them into easily reused "recipes" for working with Earth science data.

Populating Throughput

Throughput uses a Python script to access data archives (also referred to as *data catalogs* or *data repositories*) registered within the Registry of Research Data Repositories (re3data; Witt et al., 2019). The data archives were connected to code repositories in GitHub, BitBucket, and GitLab using each platform's API via a Python script via the *requests* package (https://github.com/psf/requests). Data in the Throughput Annotation Database is managed using a Neo4j graph database with a data model based on the W3C Annotation model (https://www.w3.org/TR/annotation-model/). Each individual code repository is linked to one or more data archives by virtue of either mentioning or including code elements from the data archive. The link between a code repository and a data archive creates an "annotation" element within the database that is searchable and contains provenance information about how the annotation was added (Fig. 1). As of 17 May 2021, Throughput contains information about more than 74,000 code repositories linked to more than 2400 data archives.

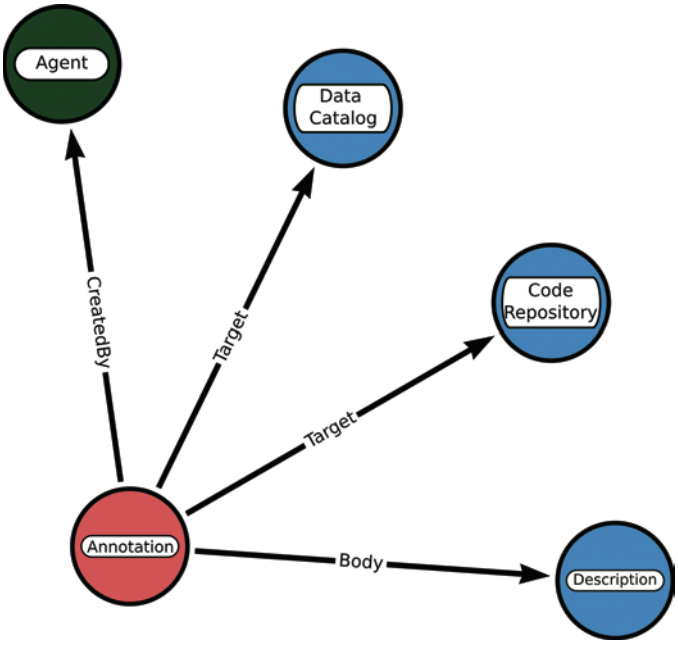

Figure 1. A representation of an annotation within Throughput is shown, which connects a code repository to a data archive. The central Annotation node has two targets, the Code Repository and the data archive (labeled here as "Data Catalog" because of the use of the https://schema.org/DataCatalog object type within the database). The Body relationship provides a connection to a plain text Description. The Agent is the individual who is the Creator of the Annotation. Any node within the graph can be a Target for another Annotation and can ultimately generate a large graph of objects within the database.

Developing a Use-Based Metadata Typology Using Throughput

Developing and applying a metadata typology (a classification system) that indicates how research data is used within a code repository will make code repositories easier to find and reuse within Throughput. This typology is not meant to pass judgement on the utility or quality of the code within the repositories but rather to add descriptive metadata that qualify the relationship between the code and the data archive referenced.

We reviewed a subset of GitHub repositories linked in Throughput to develop this typology, first looking over code repositories generally to identify particular use types, then applying and refining the classes of data use/reuse on a subset of Earth science–related code repositories. We selected code repositories that referenced data archives using the re3data subject heading "Geology and Paleontology." We used Neo4j's Cypher language to query the Throughput Database in February 2021 and retrieved 1144 GitHub repositories in total. We removed 288 GitHub repositories from the sampling frame due to missing metadata, leaving 856 GitHub repositories as our sample.

Within re3data, only 38 of the 75 data archives with the subject heading "Geology and Paleontology" were referenced by code repositories indexed by Throughput. The distribution of linked code repositories with respect to data archives shows a Pareto distribution (linked code repositories: $\bar{x} = 40$; $\tilde{x} = 14$; Fig. 2). Linked code repositories disproportionately use data from, or otherwise reference, the U.S. Geological Survey (USGS) Earthquake Hazards Program, the Paleobiology Database, Dryad, and Pangaea, with a "long tail," in which the remaining 34 data archives are referenced.

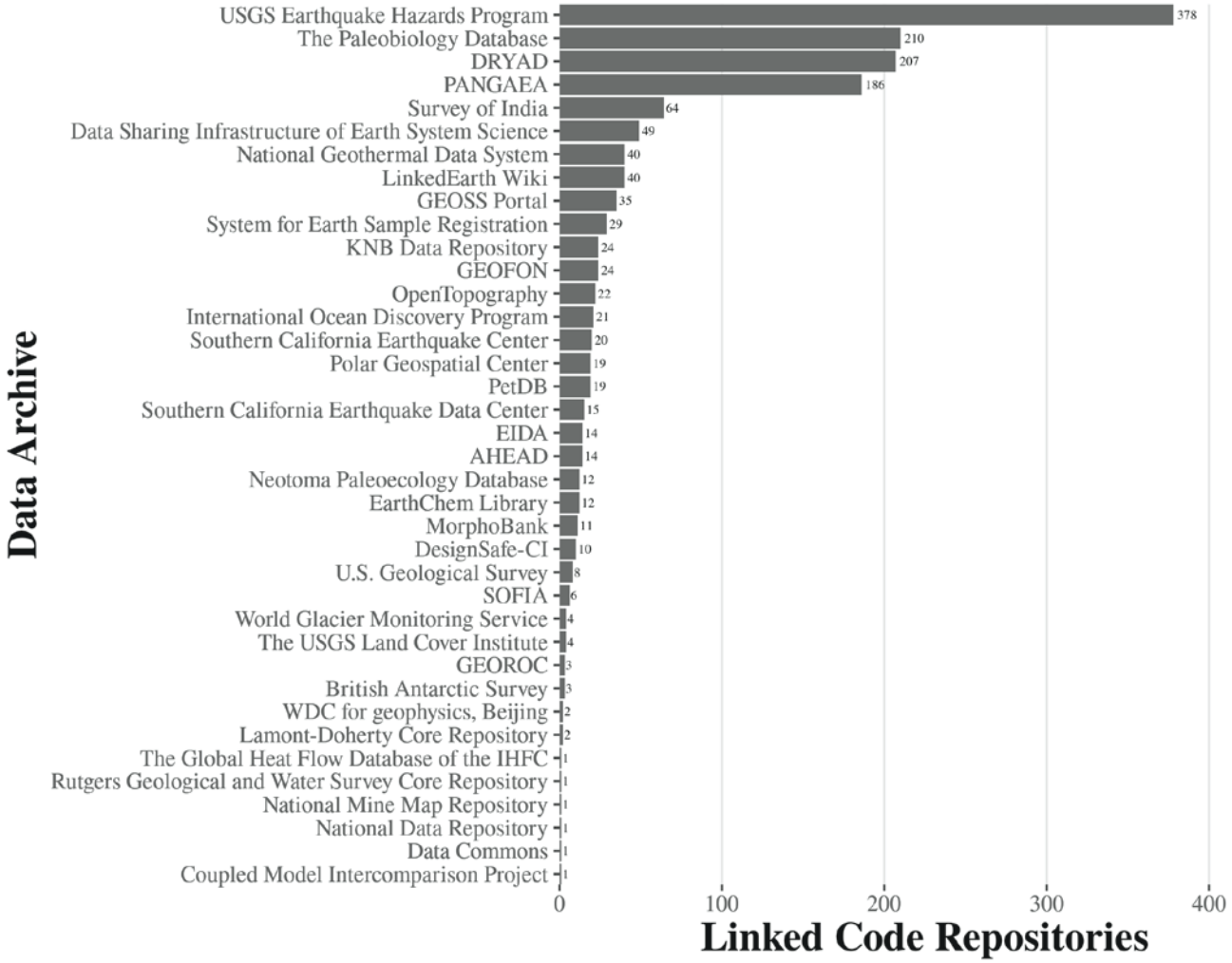

Figure 2. Bar graph shows the number of code repositories associated with individual data archives listed in the Research Resource Registry (https://re3data.org) as having "Geology and Paleontology" subject matter. The majority of data archives are referenced by fewer than 30 code repositories each. Almost half (37 of 75) of data archives with "Geology and Paleontology" as a subject were not referenced by code repositories and are not shown in this figure.

TABLE 1. DATA USE-TYPE TYPOLOGY EMPLOYED FOR THE CONTENT ANALYSIS OF CODE REPOSITORIES WITHIN THROUGHPUT, LINKED TO DATA ARCHIVES REPORTED THROUGH THE REGISTRY OF RESEARCH REPOSITORIES (https://re3data.org)

Use type	Description
Original Analysis	The code repository pulls data from an archive as a primary source for analysis or data transformation within a code repository.
Educational	The repository includes educational and instructional materials that make use of a data archive.
Software Development	The repository uses data and/or code from data archives to build freestanding tools of any sort, including libraries, plugins, frameworks, etc.
Storage	The code repository stores copies of data from data archives.
Miscellaneous Links in Articles	The repository contains articles that link to data archives, rather than any specific link to the archive itself, and does not show any other use of the data archive.
Miscellaneous Links to Data Archive Websites	The repository links to a data archive's homepage or another informational page but does not show any other use of the data archive.
Can't Categorize/Not Enough Information	404 errors (URL exists in publication or DOI but fails to link to a current public code repository) or lacks sufficient information to categorize.

Types of Data Archive Use in GitHub Repositories

We assigned *one or more* of the data-use categories (Table 1) to each code repository by examining the ways data archives or their data were used or referenced within the files of that code repository. For instance, a repository may store data (category: *Storage*) in addition to analyzing them (category: *Original Analysis*). We determined a repository's type(s) by reading README files (a text file that appears when an individual first examines the code repository online) and other documentation (when available), as well as reviewing the code itself to understand how and why it used data from a data archive. These categories were initially derived from prior research that described different types of data reuse (Coady et al., 2017; Federer, 2019; Gregory et al., 2020; Kalliamvakou et al., 2016, 2014; Pasquetto et al., 2017), but they were refined through iterative content analysis (Mayring, 2000; Pickering, 2004) of 5% of the 856 repositories that were found. We then classified an additional 10% of repositories. In total, we classified 129 repositories.

The final typology of use includes five major categories: Original Analysis, Educational, Software Development, Storage, and Miscellaneous Links. Categories are summarized in Table 1 and described in detail in the following subsections. The category *Miscellaneous Links* includes two subtypes. We also include a separate category for repositories that could not be categorized, either due to a lack of documentation or because the repository was no longer accessible. *Miscellaneous Links* uses were only applied when no other significant use for a specific data archive was found (i.e., *software development*).

After classifying our subsample of 129 code repositories, we found that repositories most commonly use data archives in support of *Software Development* (Fig. 3). Because we allow for multiple data-use classifications to be applied to a single repository, we identified 148 use types for the 129 code repositories examined. In the following subsections, we describe the kinds of data archive use we observed in greater detail and comment on how these code repositories may be useful for researchers.

Software Development

The most common category of data archive use is *Software Development* (n = 44). The *Software Development* category refers to the use of data and code from data archives to create freestanding tools and plugins. The 44 Software Development repositories in our sample might use the research data to pilot software tools, develop APIs for the database, or perform other application-oriented tasks. Not all software projects in this category were directly related to academic Earth science research. One GitHub creator used tidal data from the Center for Operational Oceanographic Products and Services to design an application that presents worldwide tidal forecasts (https://github.com/just6979/tide-catcher). However, the majority of the tools in the sample were created for researchers in the fields of geology and paleontology. Examples include an application that matches data from the Paleobiology Database to data from the GeoDeepDive Database (https://github.com/ItoErika/ePANDDA_app); the dggridR application, which uses earthquake data from the USGS Earthquake Hazards Program and creates discrete global grids that partition the surface of the Earth for R to compute spatial statistics (https://github.com/cran/dggridR); and the CoordinateCleaner Application, a tool for automated flagging of spatial and temporal errors common to paleontological collection data including data from the Paleobiology Database (https://github.com/ropensci/CoordinateCleaner).

Reuse potential: In many ways, the software and tools represented in these types of repositories are precisely what we hoped to find and link within the Throughput database. The tools are directly useful for novice researchers hoping to work with data, and software developers working on similar problems could use the repository as a template for working with the data sources.

Miscellaneous Links to Data Archive Websites

The second most common (n = 41) type of code repository is those that do not use any data from an archive directly; instead, they include a data archive's URL as an example of a

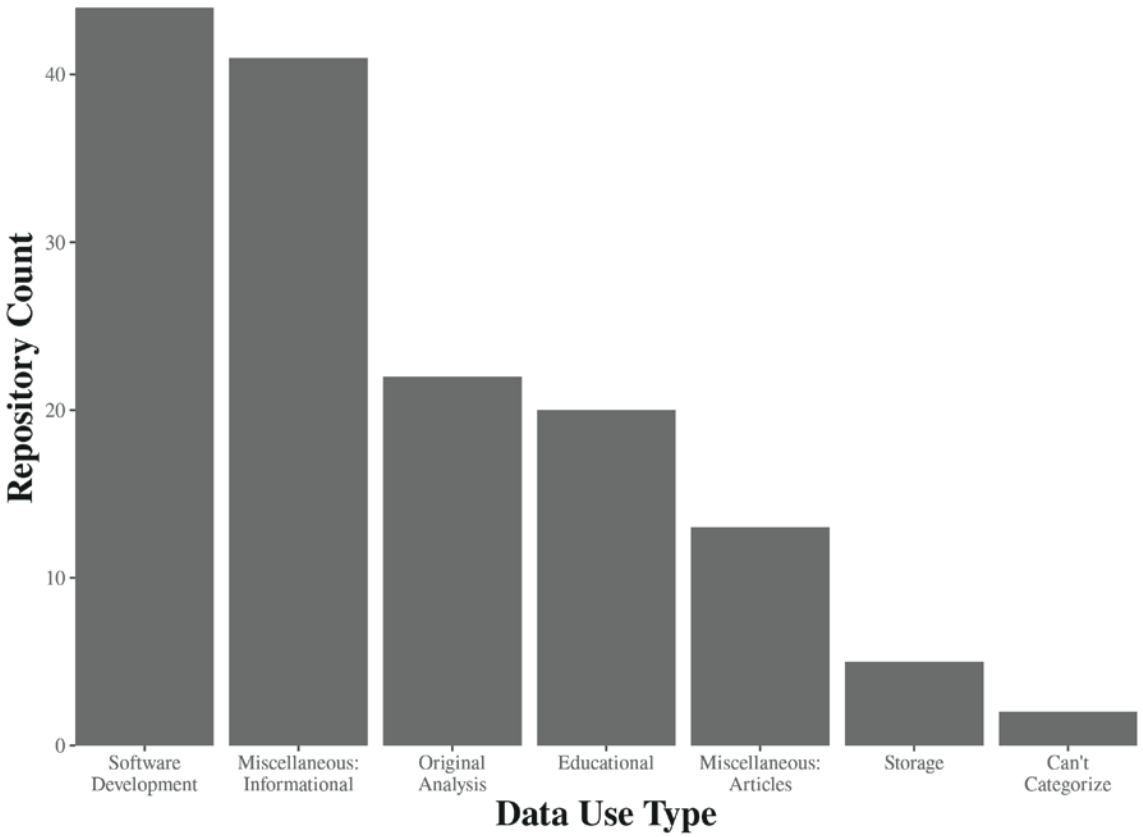

Figure 3. Distribution of data archive use types based on the sample of 129 code repositories selected for manual analysis is shown. While a small subset of code repositories could not be categorized because of a lack of information, the majority of code repositories used information from data archives either in the course of Software Development or had Miscellaneous Informational links to the data archives. Repositories meant primarily for Educational uses and Original Analysis showed lower counts but were still well represented.

place to find open data (e.g., https://github.com/biol355/biol355.github.io) or as an example of a public API (https://github.com/apibird/public-apis). Many of the repositories tagged with this category include links that appear to be directly scraped from other online registries, which cause them to be indexed by Throughput. These scraped links include lists of data archives (the Paleobiology Database, Dryad, etc.) as well as their metadata (URL, subject, content type purpose, number of data sets, etc.), but otherwise, these repositories provide little context for data use or reuse.

Reuse potential: The code in these repositories is likely less useful to Earth science researchers. However, data archive managers may be interested in these references as evidence of the broader impact of their archives. They also could serve as evidence of where data archives are gaining traction and in what communities. Some repositories in this category could also allow Throughput users to discover similar data archives and tools that are potentially useful for their goals. This category presents a challenge to the automated scraping of repositories; however, since scraping may detect these repositories, they are likely of little use to researchers who intend to use Throughput, for example, as a source of information.

Original Analysis

Original Analysis repositories (n = 22) include custom data analysis pipelines that were not designed to be generically reusable (but nevertheless, they are often valuable for people looking for code to reuse). These repositories include code meant to supplement published journal articles. For instance, one repository contains R code to examine what influences the presence and abundance of fossils in a geologic unit using data from the Paleobiology Database. The repository's README indicates that the code is a compendium to an article published in *The American Statistician* (https://github.com/psmits/notfossil). *Original Analysis* repositories also include code created by students for theses or class projects. For example, the repository https://github.com/psmits/dissertation includes analysis files that use Paleobiology Database data of North American and European occurrences of marine genera and sedimentary cover to reveal differences between fossil range-duration relationships.

Reuse potential: Repositories in this category likely have high reuse potential. While *Original Analysis* repositories are typically created to support a single study's reproducibility rather than to create generically reusable code, these scripts are nevertheless important points of reference for similar analysis, and the initial stages of obtaining and cleaning data for statistical analysis are likely to be similar across studies.

Educational

Educational repositories include tutorials, manuals, assignment instructions, lecture slides, and other pedagogical materials. Repositories (n = 20 in our sample) were classified as *Educational* if they actively used a data archive's code or data. Instructional resources that solely link to a data archive's homepage were coded as *Miscellaneous Links to Data Archive Websites* rather than educational.

A typical *Educational* repository contains class materials using a data archive's resources. For instance, one GitHub repository contains instructional materials for a graduate course in advanced paleoecology that uses data from the Neotoma Paleoecology Database in R (https://github.com/WilliamsPaleoLab/Geography52). *Educational* uses also include non-class-related tutorials, such as the Arctic Data Center's tutorials on cleaning data from the Knowledge Network for Biocomplexity (KNB) Data Repository (https://github.com/jenniferschmidt/arctic-training-repo). We note that some educational resources that draw on Earth science data archives were not necessarily in support of Earth science classes but rather generic data science courses in need of large amounts of data.

Certain data archives have high repository counts as a result of education/outreach materials that involve copying (cloning or forking) the original repository to a new personal repository for each workshop or course participant. These copied repositories do not add significantly to information heterogeneity for the data archive since they are effectively a subset of the original repository. For instance, the USGS Earthquake Hazards Program website is linked by a large number of similar repositories based on the original educational materials (e.g., https://github.com/felzek/leaftlet).

Reuse potential: Educational repositories could include useful code for a variety of contexts and could be directly reusable by instructors who are building their own curricula or simply learning on their own. Again, these repositories could be used as evidence of impact by data archive managers and curators.

Miscellaneous Links in Articles

The repositories in this subcategory do not use data from an archive but instead include collections of articles that reference particular data archives (n = 13 in our sample). For example, a GitHub creator developed a web/Android application for optical character recognition that recognizes text in a digital image (https://github.com/maximz/Photon). Data used to train this software tool included Wikipedia articles that link to the USGS Earthquake Hazards Program.

Reuse potential: Like the repositories that contain *Links to Data Archive Websites,* the code in these repositories is likely less useful to Earth science researchers but could be helpful as a broader indicator of data archive impact.

Data Storage

Data storage repositories (n = 5) include copies of data from data archives; this often is in tandem with *Software Development* or *Original Analysis* uses. For example, one repository aggregates several Earth science and biology datasets (https://github.com/hurwitzlab/planet-microbe-datapackages) as part of a pipeline that transforms data sets into data packages that conform to the Frictionless Data Standard. The resulting data are used in a larger project that integrates oceanographic, environmental, and physicochemical data layers. Still, the specific repository is used only to store the data in an online code repository.

Reuse potential: When *Data Storage* coincides with *Original Analysis*, Throughput users could utilize the repository for reproducibility tests or as a codebase. When storage accompanies *Software Development*, users may utilize the tools for their own data analysis or as a codebase.

FUTURE DIRECTIONS

Steps toward Automatic Classification of Code Repositories

While the use-based annotations described above can certainly be added by hand to Throughput (as we did with the 129 repositories we reviewed), a more scalable approach would use machine learning or other computational methods to automatically categorize a code repository. As a final step of our analysis, we explored whether there were statistically significant differences in code repository features across the data archive use types. For instance, do *Original Analysis* code repositories have longer READMEs? Are *Educational* repositories bookmarked more with "stars" (GitHub's method of bookmarking a repository)?

GitHub repository features can be divided into two groups: *user activity metrics* and *documentation metrics*. *User activity metrics* count how often GitHub users edit, contribute to, and interact with the repositories, including the number of "forks" (copies made for reuse, separate from the original repository owner), number of stars (bookmarks of the repository created by users), number of branches (internal copies of the repository that are owned by the original creator, generally used for feature development), and total commits (total number of aggregate changes to the repository codebase). *Documentation metrics* represent how much metadata and explanatory content are made available; documentation is important in making a repository reusable, regardless of type. We selected the following features to compare: the number of badges (graphic figures added by a creator to summarize repository stats, language, or content), number of characters used in the repository's description, number of README headings, and README character count. Secondary

documentation may exist on external websites, including project homepages, external documentation wikis, or hosted documentation sites such as ReadTheDocs (https://readthedocs.org). Discovering links to external documentation within a code repository is of high value, but complex, and therefore is excluded from this analysis.

For each use category, we determined basic descriptive statistics for all repository features. If a repository had multiple data-use types, its metadata metrics were included in each category. We removed one code repository from the *Links to data archive websites* category in our analysis because it was an extreme outlier in the metadata fields of user activity (more than 25× IQR above the third quartile for total commits; 1875 × IQR above the third quartile for the number of stars). We used means rather than medians for reporting because of the prevalence of zeros within many variables. We then performed ANOVA on each metadata field to test for any statistically significant difference between data archive use types with more than 10 repositories to conform to ANOVA's assumptions. If the ANOVA showed significant results, we planned to use the Tukey Test to determine where the differences lie. Table 2 presents the means of the numerical GitHub user activity and documentation metadata fields separated by data archive use type; Figure 4 presents these means as a rose diagram.

While we did not find statistically significant differences among our categories, we found that *Original Analysis* repositories tend to have lower user activity metrics than the often similar *Software development* category (Fig. 4). The *Original Analysis* category has the lowest number of issues, the second lowest number of stars and forks, and the third lowest number of branches and total commits. *Software development* repositories, meanwhile, show the highest number of branches, the second highest number of total commits and stars, and the third highest number of forks and issues. Some of these differences likely result from the work arrangements inherent in these projects; software development projects are highly collaborative, whereas repositories for one-time analyses are more likely to be written by one person.

The lack of statistical significance was not an unexpected result due to the small sample size of coded GitHub repositories. We need to expand our coded GitHub repositories and measured features to understand what, potentially, makes a given use type distinct. In the future, the relatively simple user activity and documentation metrics we used to assess repositories here (length of README and number of stars) can be significantly expanded. Factors to examine include the programming languages used within a repository, the number of files, directory structure, breadth of contributor networks, and other elements. A broader suite of repository features may make automated data-use typology classification possible. We plan to explore this going forward. A broader suite of metadata would support the identification of elements in the content metadata that appear to be more important in classifying and, ultimately, discovering these code resources. The practice of scraping various code repositories has already helped to identify some key features—for example, the presence of grant numbers and properly formatted references to particular databases. A broader scale metadata analysis would help support the clarification of best practices, in particular for educational and analytic code repositories.

Implementation and Further Development in Throughput

Until automatic classification approaches are refined, we have added a widget in the Throughput database so that users can annotate a code repository with their use type via a data entry form (Fig. 5). A user can link an existing code repository to a database and then classify it by type. In this way, machine learning approaches to solving the classification problem for this typology can implement a human-in-the-loop approach, where database end-users provide both a base set of classifications and can also validate classifications once implemented.

TABLE 2. MEANS OF GITHUB REPOSITORY FEATURES BY DATA ARCHIVE USE TYPE

	User Activity					Documentation by Creator(s)			
Code Repository Data-Use Types	Branches	Forks	Issues	Stars	Total Commits	Badges	Description Character Count	README Character Count	Number of README Headings
Software Development (44)	**6.5**	4.1	3.2	4.0	390	**0.93**	**85**	3700	6.9
Miscellaneous Informational Link (41)	2.9	2.9	**4.5**	2.7	**400**	0.29	51	**10000**	6.9
Original Analysis (22)	1.4	0.8	*0.5*	0.8	64	0.50	61	4500	4.8
Educational (20)	6.0	**5.6**	1.6	**4.4**	200	0.50	57	*1300*	1.8
Miscellaneous - Article (13)	1.4	1.6	2.1	1.3	54	*0.0*	42	1800	2.2
Storage (5)	*1.0*	*0.0*	1.4	*0.2*	130	*0.0*	*25*	2900	**29**
Can't Categorize (2)	*1.0*	1.0	3.5	1.5	*8.5*	*0.0*	50	2300	*0.0*

Note: For each column, the highest value is presented in bold font and the lowest value is presented in italics. Features of individual repositories are divided into features that represent User Activity and Creator Documentation. As described in the text, features under Documentation by Creator(s) represent changes to the repository that clarify what the repository itself does or its contents. User Activity does not explicitly change the content of the repository but represents engagement with the repository. Standardized values for each class are represented graphically in Figure 4.

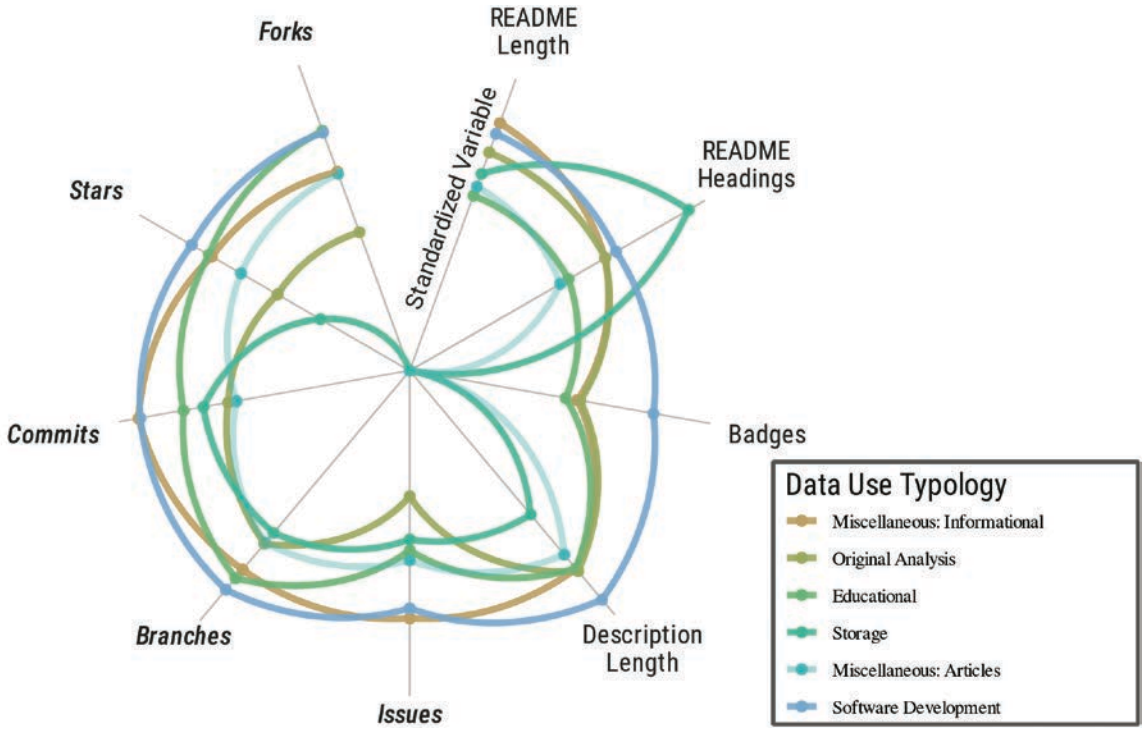

Figure 4. A wind rose diagram compares means of user activity and description metrics across repository types. User Activity metrics are indicated by text labels in bold and italics (e.g., "Stars") that represent dynamic engagement with a repository over time. Description metrics represent static properties of a repository at the time of sampling. Metrics are ordered around the wind rose based on their correlation to one another.

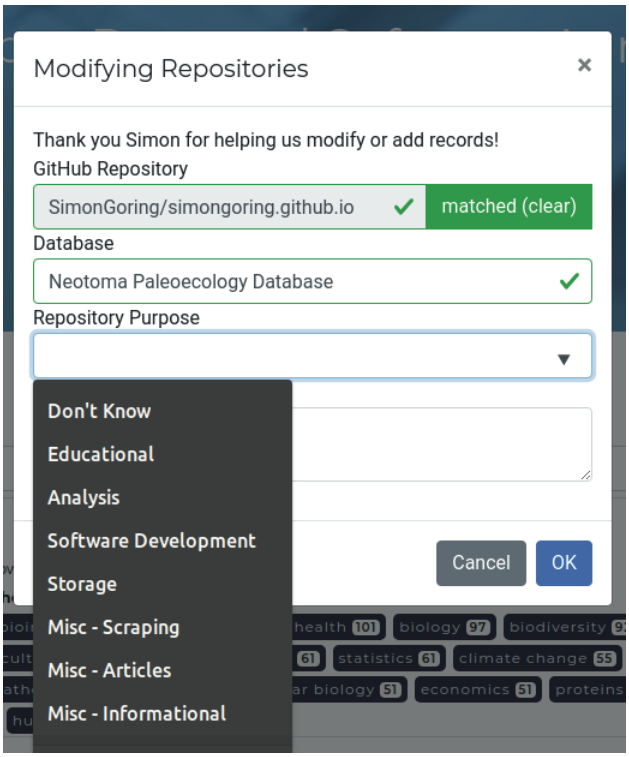

Figure 5. Screenshot from the Throughput Database Code Cookbook interface shows how a logged-in user (the author, using ORCID authorization) can add a new record into the database and classify the repository type using the classes discussed above.

We imagine that a typical end-user for Throughput would be interested in discovering repositories with a specific typology. For instance, a graduate student would be more interested in finding a repository with analysis applied to a data set from the Neotoma Paleoecology Database than he or she would a repository that mentions that database in an XML document.

In our future work, we are interested in exploring how Throughput can be leveraged to not only make code easier to find, but also to help spread best practices in writing code or documenting methods. The Methods section of a paper does not always allow article authors to explain or express all of the decisions made in the data cleaning and alignment stages; however, being able to see examples of code, discovered through a portal like Throughput, makes this process more straightforward. Additionally, it provides the opportunity for community members to discover emerging standards in analysis and define or revise best practices for working with the data (for instance,

the recommendations on working with Tephra data; Abbott et al., 2020). In this way, code repositories can be leveraged as tools for knowledge transmission that transcend institutional or disciplinary boundaries imposed by social barriers. Examples of best practices already can be found in some of the *Educational* repositories we reviewed and sometimes are published by the data archives. For example, the Neotoma Paleoecology Database provides these in the form of vignettes associated with R packages (https://github.com/NeotomaDB/neotoma), or worked code examples are shared as part of online workshops and stored within code repositories (https://github.com/NeotomaDB/Workshops) (Goring et al., 2018).

New Ways of Understanding Data Practices and Data Archive Impact

Our work has underscored the varied ways that the Earth science community makes use of GitHub. Prior studies have estimated that only 63.4% of GitHub repositories are used for software development (Kalliamvakou et al., 2016). Our study similarly showed that GitHub is used by the Earth science community for more than just software development (though we note that our study is not directly comparable to that of Kalliamvakou, as we categorized repositories based on the way they use data archives and not by their primary purpose). We found that many repositories referencing Earth science data archives are used for ad hoc data storage, as sandboxes for analysis, or for the drafting of papers. Future studies of code and data practices must take this broader range of use cases into account.

Additionally, our work provides a greater understanding of the impact and use of Earth science data archives even beyond Earth science communities. Much of the research on data reuse has focused on reuse for research: the "direct reuse" of a data set for a new project or the "integrative reuse" of many data sets for synthesis (Pasquetto et al., 2017). However, we found that data was reused by GitHub developers for a much wider range of purposes, including in tutorials and class assignments, or as a testbed to support software development. The use-based typology we developed for Throughput reveals a much broader universe of data reuse than previously discussed. Future work might consider the ways in which these different forms of use may prove to be important to data archive managers—both in providing new use cases to support and in showing new ways in which Earth science data have impact. For instance, we found that several data science classes had built lessons around data from archives such as the USGS Earthquake Hazards Program, including Google's "Android Basics" course on Udacity (https://www.udacity.com/course/android-basics-nanodegree-by-google–nd803). Though this is not an example of "direct reuse" of data, this would certainly be important for showing the broader utility and impact of the USGS data archive. Similarly, repositories in the *Link to Data Archive Websites* category do not show clear evidence of reuse per se, but they do show an archive's reach and broader impact.

CONCLUSION

In this chapter, we described the Throughput database and our efforts to make reusable code more findable, thereby making it easier to access and process data from large Earth science data archives. We also described our approach to enriching the code repositories with use-based metadata. We took a unique approach to categorizing code repositories: they are classified according to their mode of data use and reuse. Our hope is that by adding this use-based metadata to Throughput, we will make it easier for novice users, in particular, to find relevant resources. We believe that this metadata will help to identify highly useful or reusable code repositories, such as those containing code for *Original Analysis* and *Software Development*. We also believe this work will make it easier for data archives to show their impact in and beyond their disciplines.

The work undertaken by Throughput—to link code repositories to particular data resources—has highlighted the Earth science community's evolving use of GitHub. We see that repositories are used for a broad range of purposes and that different types of repositories may have particular signatures related to user interaction and the descriptive elements of the repositories themselves. The evolution of software and service citation practices, and the increasing use of DOIs within code repositories (ESIP Software and Services Citation Cluster, 2019), will also help introduce a secondary framework for understanding patterns of code and data use. Ultimately, code repositories represent an important resource for education and knowledge dissemination; however, their relative lack of structured metadata and the lack of an existing typology have limited our ability to discover and reuse these resources. This publication represents a first step toward leveraging existing code repositories to reduce "time to science" for the next generation of Earth science researchers.

ACKNOWLEDGMENTS

This work was funded by grants to SJG (NSF-1928366; NSF-1740699) and AKT (NSF-1928317). Simon Goring thanks the members of the EarthCube Science Support Office and members of EarthCube for providing assistance, coordination, and excellent discussions throughout the project. We also thank our Throughput Project partners Kerstin Lehnert, Nick McKay, Doug Fils, Anders Noren, Jack Williams, Shane Loeffler, and Stephen Kuehn.

All code for figures and analysis can be found at https://github.com/throughput-ec/datausetypology.

REFERENCES CITED

Abbott, P., Bonadonna, C., Bursik, M., Cashman, K., Davies, S., Jensen, B., Kuehn, S., Kurbatov, A., Lane, C., Plunkett, G., Smith, V., Thomlinson, E., Thordarsson, T., Walker, J, and Wallace, K., 2020, Community established best practice recommendations for tephra studies—From collection through analysis (version 3.0.0) [Data set]: Zenodo, http://doi.org/10.5281/zenodo.3866266 (June 2021).

Barnes, N., 2010, Publish your computer code: It is good enough: Nature, v. 467, no. 753, https://doi.org/10.1038/467753a.

Belhajjame, K., Corcho, O., Garijo, D., Zhao, J., Newman, D., Klyne, G., Page, K., and Roos, M., 2012a, Workflow-centric research objects: A first class citizen in the scholarly discourse, *in* Van Harmelen, F., et al., eds., Proceedings of the Workshop on Semantic Publishing (SePublica 2012): 9th Extended Semantic Web Conference, Hersonissos, Crete, Greece, May 28, p. 1–12; http://ceur-ws.org/Vol-903/sepublica2012-complete.pdf.

Belhajjame, K., Roos, M., Garcia-Cuesta, E., Klyne, G., Zhao, J., De Roure, D., Goble, C., Gomez-Perez, J.M., Hettne, K., and Garrido, A., 2012b, Why workflows break—Understanding and combating decay in Taverna workflows: Proceedings of the 2012 IEEE 8th International Conference on E-Science, p. 1–9, https://doi.org/10.1109/eScience.2012.6404482.

Berhe, A.A., and Ghezzehei, T.A., 2021, Race and racism in soil science: European Journal of Soil Science, v. 72, no. 3, p. 1292–1297, https://doi.org/10.1111/ejss.13078.

Bonham, K.S., and Stefan, M.I., 2017, Women are underrepresented in computational biology: An analysis of the scholarly literature in biology, computer science and computational biology: PLoS Computational Biology, v. 13, no. 10, https://doi.org/10.1371/journal.pcbi.1005134.

Carrasco, M.A., Barnosky, A.D., Kraatz, B.P., and Davis, E.B., 2007, The Miocene MammaL Mapping Project (Miomap): An online database of Arikareean through Hemphillian fossil mammals: Bulletin of Carnegie Museum of Natural History, v. 39, p. 183–188, https://doi.org/10.2992/0145-9058(2007)39[183:TMMMPM]2.0.CO;2.

Coady, S.A., Mensah, G.A., Wagner, E.L., Goldfarb, M.E., Hitchcock, D.M., and Giffen, C.A., 2017, Use of the National Heart, Lung, and Blood Institute Data Repository: The New England Journal of Medicine, v. 376, no. 19, p. 1849–1858, https://doi.org/10.1056/NEJMsa1603542.

COHMAP MEMBERS, 1988, Climatic changes of the last 18,000 years: Observations and model simulations: Science, v. 241, p. 1043–1052, https://doi.org/10.1126/science.241.4869.1043.

Davison, A., 2012, Automated capture of experiment context for easier reproducibility in computational research: Computing in Science & Engineering, v. 14, no. 4, p. 48–56, https://doi.org/10.1109/MCSE.2012.41.

De Solla Price, D.J., and Beaver, D., 1966, Collaboration in an invisible college: The American Psychologist, v. 21, no. 11, p. 1011–1018, https://doi.org/10.1037/h0024051.

Downs, R.R., Marshall, J.J., and Wolfe, R.E., 2006, The Software Reuse portal: A case study in packaging software to contribute to reuse practices: Eos (Transactions, American Geophysical Union), v. 87, no. 52, http://www.proquest.com/docview/29507017/3A45691584BC4595PQ/7.

Du, C., Cohoon, J., Lopez, P., and Howison, J., 2021, Softcite dataset: A dataset of software mentions in biomedical and economic research publications: Journal of the Association for Information Science and Technology, v. 72, no. 7, p. 870–884, https://doi.org/10.1002/asi.24454.

Duck, G., Nenadic, G., Filannino, M., Brass, A., Robertson, D.L., and Stevens, R., 2016, A survey of bioinformatics database and software usage through mining the literature: PLoS One, v. 11, no. 6, https://doi.org/10.1371/journal.pone.0157989.

ESIP Software and Services Citation Cluster, 2019, Software and services citation guidelines and examples. Ver. 1. ESIP: https://doi.org/10.6084/m9.figshare.7640426.

Federer, L.M., 2019, Who, what, when, where, and why? Quantifying and understanding biomedical data reuse: University of Maryland Digital Repository: https://doi.org/10.13016/60jd-9hux.

Fox, P., Erdmann, C., Stall, S., Griffies, S.M., Beal, L.M., Pinardi, N., Hanson, B., Friedrichs, M.A.M., Feakins, S., Bracco, A., Pirenne, B., and Legg, S., 2021, Data and software sharing guidance for authors submitting to AGU journals: https://doi.org/10.5281/zenodo.5124741.

Gerard, R., Downs, R.R., Marshall, J.J., and Wolfe, R.E., 2007, The Software Reuse Working Group: A case study in fostering reuse: Proceedings of the 2007 IEEE International Conference on Information Reuse and Integration, p. 24–29, https://doi.org/10.1109/IRI.2007.4296592.

Gil, Y., David, C.H., Demir, I., Essawy, B.T., Fulweiler, R.W., Goodall, J.L., Karlstrom, L., Lee, H., Mills, H.J., Oh, J., Pierce, S.A., Pope, A., Tzeng, M.W., Villamizar, S.R., and Yu, X., 2016, Toward the geoscience paper of the future: Best practices for documenting and sharing research from data to software to provenance: Earth and Space Science, v. 3, no. 10, p. 388–415, https://doi.org/10.1002/2015EA000136.

Goble, C., and De Roure, D., 2009, The impact of workflow tools on data-centric research, *in* Data Intensive Computing: The Fourth Paradigm of Scientific Discovery: Redmond, Washington, Microsoft Research, p. 137–146, https://www.microsoft.com/en-us/research/publication/fourth-paradigm-data-intensive-scientific-discovery/.

Goring, S., Thomer, A.K., Loeffler, S., Kuehn, S.C., Noren, A.J., McKay, N.P., Fils, D., Lehnert, K.A., and Lehnert, K.A., 2020, Throughput: A tool to connect research data and code examples to improve learning opportunities and help build better documentation: Geological Society of America Abstracts with Programs, v. 52, no. 6, https://doi.org/10.1130/abs/2020AM-359701.

Goring, S.J., Graham, R., Oeffler, S., Myrbo, A., Oliver, J.S., Ormond, C., and Williams, J.W., 2018, The Neotoma Paleoecology Database: A Research Outreach Nexus (1st ed.): Cambridge, UK, Cambridge University Press, https://doi.org/10.1017/9781108681582.

Graham, R.W., and Lundelius, E.L., 1994, FAUNMAP: A database documenting Late Quaternary distributions of mammal species in the United States: Springfield, Illinois, Illinois State Museum.

Gregory, K., Groth, P., Scharnhorst, A., and Wyatt, S., 2020, Lost or found? Discovering data needed for research: Harvard Data Science Review, v. 2, no. 2, https://doi.org/10.1162/99608f92.e38165eb.

Hey, A.J.G., ed., 2009, The Fourth Paradigm: Data-Intensive Scientific Discovery: Redmond, Washington, Microsoft Research, 241 p.

Howison, J., and Bullard, J., 2016, Software in the scientific literature: Problems with seeing, finding, and using software mentioned in the biology literature: Journal of the Association for Information Science and Technology, v. 67, no. 9, p. 2137–2155, https://doi.org/10.1002/asi.23538.

Ince, D.C., Hatton, L., and Graham-Cumming, J., 2012, The case for open computer programs: Nature, v. 482, no. 7386, p. 485–488, https://doi.org/10.1038/nature10836.

Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D.M., and Damian, D., 2014, The promises and perils of mining GitHub: Proceedings of the 11th Working Conference on Mining Software Repositories: New York, Association for Computing Machinery, p. 92–101, https://doi.org/10.1145/2597073.2597074.

Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D.M., and Damian, D., 2016, An in-depth study of the promises and perils of mining GitHub: Empirical Software Engineering, v. 21, no. 5, p. 2035–2071, https://doi.org/10.1007/s10664-015-9393-5.

Katz, D.S., and Cue Hong, N.P., 2018, Software citation in theory and practice, *in* Davenport, J.H., Kauers, K., Labahn, G., and Urban, J., eds., Mathematical Software—ICMS 2018: New York, Springer International Publishing, p. 289–296, https://doi.org/10.1007/978-3-319-96418-8_34.

Kuehn, S.C., Bursik, M.I., Goring, S.J., Kodama, S., Kuehn, S.C., Kurbatov, A., Lehnert, K., Profeta, L., Ramdeen, S., Quinn, D.P., Wallace, K., and Walker, J.D., 2021, Making tephra data accessible and interoperable through community-driven best practices for digital data collection and documentation: Geological Society of America Abstracts with Programs, v. 53, no. 6, https://doi.org/10.1130/abs/2021AM-370073.

Ludäscher, B., Lin, K., Bowers, S., Jaeger-Frank, E., Brodaric, B., and Baru, C., 2006, Managing scientific data: From data integration to scientific workflows, *in* Sinha, A.K., ed., Geoinformatics: Data to Knowledge: Geological Society of America Special Paper 397, p. 109–129, https://doi.org/10.1130/2006.2397(08).

Lynnes, C., Zhu, M.Q., Blythe, J., Williamson, T.N., Burnett, J., Huffer, E., Armstrong, E.M., Munroe, J.R., Siarto, J., Reese, M., Norton, J., Newman, D.J., and Durbin, C., 2020, Usage-based discovery of Earth observations: Abstract IN012-02 presented at 2020 Fall Meeting, AGU, 1–17 December, https://agu.confex.com/agu/fm20/meetingapp.cgi/Paper/703367.

Marshall, J.J., Downs, R.R., and Samadi, S., 2010, Relevance of software reuse in building advanced scientific data processing systems: Earth Science Informatics, v. 3, no. 1–2, p. 95–100, https://doi.org/10.1007/s12145-010-0054-3.

Mattmann, C.A., Downs, R.R., Marshall, J.J., Most, N.F., and Samadi, S., 2011, Tools to support the reuse of software assets for the NASA Earth Science Decadal Survey missions: IEEE Geoscience and Remote Sensing Society Newsletter, p. 1–6, https://ntrs.nasa.gov/citations/20120010312.

Mayring, P., 2000, Qualitative content analysis: Forum Qualitative Social Research Sozialforschung, v. 1, no. 2, https://doi.org/10.17169/fqs-1.2.1089.

Pasquetto, I.V., Randles, B.M., and Borgman, C.L., 2017, On the reuse of scientific data: Data Science Journal, v. 16, https://doi.org/10.5334/dsj-2017-008.

Pasquetto, I.V., Borgman, C.L., and Wofford, M.F., 2019, Uses and reuses of scientific data: The data creators' advantage: Harvard Data Science Review, v. 1, no. 2, https://doi.org/10.1162/99608f92.fc14bf2d.

Peng, R.D., 2011, Reproducible research in computational Science: Science, v. 334, no. 6060, p. 1226–1227, https://doi.org/10.1126/science.1213847.

Peters, I., Kraker, P., Lex, E., Gumpenberger, C., and Gorraiz, J.I., 2017, Zenodo in the spotlight of traditional and new metrics: Frontiers in Research Metrics and Analytics, v. 2, https://doi.org/10.3389/frma.2017.00013.

Pickering, M.J., 2004, Qualitative content analysis, *in* Lewis-Beck, M., Bryman, A., and Liao, T.F., The SAGE Encyclopedia of Social Science Research Methods: Thousand Oaks, California, Sage Publications, https://doi.org/10.4135/9781412950589.n779.

Smith, A.M., Katz, D.S., and Niemeyer, K.E., 2016, Software citation principles: PeerJ. Computer Science, v. 2, p. e86, https://doi.org/10.7717/peerj-cs.86.

Stall, S., Yarmey, L.R., Boehm, R., Cousijn, H., Cruse, P., Cutcher-Gershenfeld, J., Dasler, R., de Waard, A., Duerr, R., Elger, K., Fenner, M., Glaves, H., Hanson, B., Hausman, J., Heber, J., Hills, D.J., Hoebelheinrich, N., Hou, S., Kinkade, D., Koskela, R., Martin, R., Lehnert, K., Murphy, F., Nosek, B., Parsons, M.A., Petters, J., Plante, R., Robinson, E., Samors, R., Servilla, M., Ulrich, R., Witt, M., and Wyborn, L., 2018, Advancing FAIR data in Earth, space, and environmental science: Eos Science News by AGU, v. 99, https://doi.org/10.1029/2018EO109301.

Stodden, V., Guo, P., and Ma, Z., 2013, Toward reproducible computational research: An empirical analysis of data and code policy adoption by journals: PLoS One, v. 8, no. 6, https://doi.org/10.1371/journal.pone.0067111.

Stodden, V., McNutt, M., Bailey, D.H., Deelman, E., Gil, Y., Hanson, B., Heroux, M.A., Ioannidis, J.P.A., and Taufer, M., 2016, Enhancing reproducibility for computational methods: Science, v. 354, no. 6317, p. 1240–1241, https://doi.org/10.1126/science.aah6168.

Thomer, A.K., and Wickett, K.M., 2020, Relational data paradigms: What do we learn by taking the materiality of databases seriously?: Big Data & Society, v. 7, no. 1, https://doi.org/10.1177/2053951720934838.

Thomer, A.K., Wickett, K.M., Baker, K.S., Fouke, B.W., and Palmer, C.L., 2018, Documenting provenance in noncomputational workflows: Research process models based on geobiology fieldwork in Yellowstone National Park: Journal of the Association for Information Science and Technology, v. 69, no. 10, p. 1234–1245, https://doi.org/10.1002/asi.24039.

Wilkinson, M.D., and 52 others, 2016, The FAIR Guiding Principles for scientific data management and stewardship: Scientific Data, v. 3, 160018, https://doi.org/10.1038/sdata.2016.18.

Williams, J.W., Grimm, E.G., Blois, J., Charles, D.F., Davis, E., Goring, S.J., Graham, R., Smith, A.J., Anderson, M., Arroyo-Cabrales, J., Ashworth, A.C., Betancourt, J.L., Bills, B.W., Booth, R.K., Buckland, P.I., Curry, B.B., Giesecke, T., Jackson, S.T., Latorre, C., Nichols, J., Purdum, T., Roth, R.E., Stryker, M., and Takahara, H., 2018, The Neotoma Paleoecology Database: A multi-proxy, international community-curated data resource: Quaternary Research, v. 89, p. 156–177, https://doi.org/10.1017/qua.2017.105.

Witt, M., Stall, S., Duerr, R., Plante, R., Fenner, M., Dasler, R., Cruse, P., Hou, S., Ulrich, R., and Kinkade, D., 2019, Connecting researchers to data repositories in the Earth, space, and environmental sciences, *in* Manghi, P., Candela, L., and Silvello, G., eds., Digital Libraries: Supporting Open Science (Vol. 988): New York, Springer International Publishing, p. 86–96, https://doi.org/10.1007/978-3-030-11226-4_7.

Wright, H.E., and Bartlein, P.J., 1993, Reflections on COHMAP: The Holocene, v. 3, no. 1, p. 89–92, https://doi.org/10.1177/095968369300300110.

Zimmerman, A.S., 2007, Not by metadata alone: The use of diverse forms of knowledge to locate data for reuse: International Journal on Digital Libraries, v. 7, no. 1–2, p. 5–16, https://doi.org/10.1007/s00799-007-0015-8.

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, v. 33, no. 5, p. 631–652, https://doi.org/10.1177/0162243907306704.

Manuscript Accepted by the Society 17 March 2022
Manuscript Published Online 24 January 2023

Printed in the USA