Please cite this as: Batist, Z. and Roe, J. 2024 Open Archaeology, Open Source? Collaborative practices in an emerging community of archaeological software engineers, Internet Archaeology 67. https://doi.org/10.11141/ia.67.13

Open Archaeology, Open Source? Collaborative practices in an emerging community of archaeological software engineersOpen Data

Zachary Batist and Joe Roe

Surveying the first quarter-century of computer applications in archaeology, Scollar (1999) lamented that the field relied almost exclusively on 'hand-me-down' tools repurposed from other disciplines. Twenty-five years later, this is no longer the case: computational archaeologists often find themselves practising the dual roles of data analyst and research software engineer (Baxter *et al*. 2012; Schmidt and Marwick 2020), developing and applying new tools that are tailored specifically to archaeological problems and archaeological methods. Though this trend can be traced to the very earliest days of the field (Cowgill 1967), its most recent manifestation is distinguished by its apparent embrace of practices from free and open-source software. Most prominently, since around 2015, there has been a rapid uptake of workflow tools designed for open-source development communities, such as the version control system git and associated online source code management platforms (e.g. GitHub, GitLab). These tools facilitate collaboration among developers and users of open source software using patterns that can diverge quite radically from conventional scholarly norms (Tennant *et al*. 2020).

In this article, we investigate modes of collaboration in this emerging community of practice using 'open-archaeo ', a curated list of archaeological software, and data on the activity of associated GitHub repositories and users. We conduct an exploratory quantitative analysis to characterise the nature and intensity of these collaborations and map the collaborative networks that emerge from them. We document uneven adoption of open source collaborative practices beyond the basic use of git as a version control system and GitHub to host source code. Most projects do make use of collaborative features and, through shared contributions, we can trace a collaborative network that includes the majority of archaeologists active on GitHub. However, a majority of repositories have 1-3 contributors, with only a few projects distinguished by an active and diverse developer base. Direct collaboration on code or other repository content - as opposed to the more passive, social media-style interaction that GitHub supports – remains very limited. In other words, there is little evidence that archaeologists ' adoption of open-source tools (git and GitHub) has been accompanied by the decentralised, participatory forms of collaboration that characterise other open-source communities. On the contrary, our results indicate that research software engineering in archaeology remains largely embedded in conventional professional norms and organisational structures of academia.

1. Introduction

A 2012 special issue of *World Archaeology* marked the coming of age of 'open archaeology', a new, digitally orientated archaeology 'predicated on promoting open redistribution and access to the data, processes and syntheses generated within the archaeological domain 'with the aim of 'maximizing transparency, reuse and engagement while maintaining professional probity' (Beck and Neylon 2012, 480-81), situated within the wider open science movement. In the same issue, Ducke (2012) specifically drew attention to software -- the programs and other operating information used by a computer to analyse archaeological information -- calling on archaeologists to more actively engage in open-source practices - making software with source code available for anyone to freely access, modify or reuse. Open source, though originating outside academia, emerged as an important component of and complement to open science, sharing its vision for a communal, self-correcting and transparent mode of knowledge production. Where before archaeologists had relied almost exclusively on 'hand-me-down' software repurposed from other fields (Scollar 1999), open archaeology envisaged a community of archaeological research software engineers orientated around open-source development practices and tools.

More than decade on, we are in a position to ask whether this hopeful early rhetoric of open archaeology has been borne out in actually existing open-source research software engineering in the field. Does academic open source actually make research processes more transparent and improve research outcomes? Is it actually boosting efficiency by establishing a common store of knowledge and productive code? Is it actually helping to foster new globe-spanning connections and lead to novel research trajectories that would not otherwise come to pass? In other words, is there more to 'open archaeology' than just uploading text files to the internet?

The aspirations of open archaeology's early proponents (e.g. Kansa *et al*. 2014; Kintigh *et al*. 2015) were tempered by notes of caution that 'the reward structures in academic and professional archaeology do little to incentivise participation in Open Archaeology' (Lake 2012, 475, echoing Beck and Neylon 2012; Kansa and Whitcher Kansa 2012; Huggett 2012; Limp *et al.* 2011). Following recent work by Nguyễn and Rampin (2022), Pownall *et al*. (2023) and Leonelli (2023), we believe that the outcomes listed above indeed only arise in contexts where there are organisation structures, governance strategies, and professional norms that encourage them. Thus practical circumstances and systemic value regimes that frame what it means to work as an archaeologist presently inhibit the potential for radical transformation, even among open science's most ardent supporters.

There is no question that archaeologists are prolific software developers (Batist and Roe 2023). But beyond simply making their code available on the web, do archaeologists also implement social strategies to advance open-source ideals? Does archaeological open source actually help achieve greater transparency, sustainability, and community participation? And if not, what does it actually achieve?

This article presents a survey of open-source archaeological software development with two goals in mind:

1. we identify *what* kinds of software archaeologists are making; and
2. we evaluate *how* archaeologists create these tools, with particular emphasis on practices of collaboration.

We use quantitative analysis to consider how archaeological software development may be benefiting from, or missing out on, the affordances that open-source development models

provide, specifically the value added through working as part of a broader community of invested stakeholders, processes of iterative improvement, and increased code transparency. As such, our work examines whether archaeologists are harnessing the collaborative potential that the open science movement ascribes to the use of open source software and resources.

2. Open Science and Open Source

Academic open source has a complicated relationship with open source as practised by professional software developers, which has its own distinct history and is framed by different objectives, challenges, and value regimes. Despite this, the open science movement, within which open archaeology emerged, draws direct inspiration from open source. For instance, the Open Knowledge Foundation (2015) publishes a widely accepted definition of 'open' in the context of scholarly communication that explicitly refers to the definition of 'open source' published by the Open Source Initiative (2007), an authoritative open-source advocacy group. The open science movement further mimics open source by operationalising scholarly communication through technical infrastructures and protocols that closely resemble systems and processes designed to develop open-source software (e.g., the use of plain text, line-resolution version control, emphasis on formal licencing, the general hacker aesthetic). However, academic work, including the development of academic software, differs significantly from the work involved in massive open-source projects that literally run the internet, such as the Linux kernel, openSSL and the Firefox web browser. While they may use similar tools and technical protocols to manage coding operations, the open science and open source movements are governed by different social and professional warrants and interests. In other words, publishing code openly on the web has different meanings, impacts and implications for archaeologists and professional software developers (Ratto 2007; Kelty 2008, chap. 9).

2.1 Open source

Open source is a software development model that prioritises transparent work processes. Initially driven by the idea that computer users should be free to understand and manipulate the software that they install on their computers (e.g., 'free software ', as initially conceived by the Free Software Foundation), open source has become a means of collaborative software development (Kelty 2008, chap. 3, especially page 99 onwards). By putting one's code on the web without restriction on how it may be used or manipulated, this encourages creativity to flourish as people contribute to help improve the code base. Software thus emerges from the coordinated labour of worldwide volunteers, who shape the product according to the collective vision. An open code base may also be used to support alternative projects whose missions diverge from the original plan, and an entire project may be 'forked ', or taken in a new direction if contributors are dissatisfied with how core developers run things.

Open source has traditionally been referred to as being based on meritocratic principles (Raymond 1999, 39). A good test of whether a contribution should be included in a published software release is whether it is functional (Kelty 2008, 220). Moreover, with more eyes looking over a code base it is easier to identify flaws with a contribution, and flag potential bugs or security issues (Raymond 1999, 27-30). This is all done in the spirit of producing functional code, and in ideal circumstances faulty contributions will be corrected before inclusion. Personal ego is minimised in favour of co-creating stable and functional outcomes (Raymond 1999, 39-41).

However, this is not the same as saying that open source is completely anarchic or based on the 'wisdom of crowds '. In fact, successful open-source projects incorporate complex organisational structures, governance strategies, and forms of social mediation to help delegate and vet contributions made by distributed participants (O 'Neil 2009). They rely on, rather than eschew, institutional support structures, in order to motivate work, keep volunteer maintainers involved, and generally ensure that the project can be sustained over the long term. Open source is more than just putting your code online; to be successful, it requires participation in a social experience (Ratto 2003; Kelty 2008).

In other words, as with many so-called 'soft skills' that are crucial for academic professional development, additional competencies relating to the maintenance, management, and distribution of software, such as the ability to receive and implement feedback, set and stick with long-term goals, coordinate labour, document work practices, and collaborate with others, are grossly under-appreciated factors that contribute to an open-source project's success.

We therefore consider open source to be a means of collaboration more than a means of transmitting information. It involves developing software as part of a group, developing consensus, and working with common purpose. Crucially, it also involves having a welcoming attitude, a sense of humility, and an understanding that one's work may be appropriated and used in unanticipated ways.

2.2 Open science

The open science movement comprises a series of practices and principles intended to make research more accessible, transparent, and efficient. Although the concept of 'open' is somewhat nebulous in terms of its abstract definition and with regard to what real-world applications count as being open, one commonly cited definition describes content that 'can be freely used, modified, and shared by anyone for any purpose' (Open Knowledge Foundation 2015). This definition does not state what open is for, how to be open, or any sort of social or discursive framing behind the open movement. However, most open science advocates (including archaeologists, as exemplified by Beck and Neylon (2012) and Marwick *et al*. (2017)) claim that they are motivated by a desire to facilitate novel research opportunities, make participation in scientific research more equitable, reclaim science as a public good, and enhance how findings are validated and legitimised.

The idea that scientists should generally contribute to a public domain of knowledge without profit motive has led to open science being heralded as revolutionary, community orientated, and anti-capitalist means of production. However, while open science does have the *potential* to effect radical change, this is not a given. The social and institutional contexts in which we do science is firmly embedded within capitalist and neoliberal power structures that reward individualistic competition and do little to actually encourage equitable and accessible research practices, and as such, make it difficult to fully embrace open science ideals (Mirowski 2018). Moreover, the open science movement, which is dominated by STEM disciplines, prioritises a grossly simplified and asocial notion of what science is and entails. Namely, it considers science as the accumulation and assembly of a species-level understanding of the world, which is not held by any one individual but is stored in seemingly value-neutral and disembodied media, facts and observations. This is manifested by information and communication technologies that host files, document processes, facilitate co-working opportunities, and perform automated processes (e.g. digital repositories, reproducible notebooks, automated evaluation of research findings, systems for ensuring stable references to consistent identifiers, etc.).

This culminates in an obsessive concern with digital workflows pertaining to legal and logistical issues; for many proponents of open science, publishing is largely considered the business of typesetting and copyright law, which could be rendered moot by using automated publishing workflows and by encouraging use of open licensing agreements (cf. Foster and Deardorff 2017; Harnad 1998). Viewed as merely technical systems, these could be resolved through technical means. However academic publishing and ownership involve social arrangements that serve to stabilise knowledge, grant authority to validated claims, and enable science to move forward (Kelty *et al*. 2008: 274-275). In other words, technocentric visions of publication workflows tend to ignore the fact that publication is a cultural phenomenon, whereby projects are made complete and knowledge claims are articulated, credited, and rendered accountable to the people who proposed them. However, technological systems have become so emblematic of open science that the use of these tools and resources designed to *support* open science is often mistaken for *actually doing* open science (Leonelli 2023, 23-24).

Open science is typically compared with the open-source movement in that they both involve a distributed, digitally mediated and worldwide labour force, who somehow derive rough consensus directed towards assets held in the public domain (Tennant *et al*. 2020). But they differ in terms of the contexts in which they operate, the stakeholders involved, and the kinds of outcomes they produce. Whereas open source emerged from concerns over consumer rights and then developed as a means of maintaining resilient and collectively motivated projects, open science is driven by a warrant to make research practices more transparent and accessible. Open source is performed by professional and hobbyist software developers alike, and participants contribute in a wide variety of ways (including: programming, writing documentation, translating software and documentation, bug reporting, and financial support), but, in open science, scientists are usually the only participants actively involved in creating and maintaining contributions.[1] Moreover, whereas open source projects often attract participants with varied stakes in the software and use cases in mind, open science projects are typically bounded by small communities of specialists with very particular needs (Kling *et al*. 2003). Additionally, open science is bounded by the professional contexts in which science operates and, as such, produces outputs that can be easily credited to specific sets of individuals for reasons of resumé-building, tenure and promotion (Mirowski 2018; Dorta-González *et al*. 2021). Open science projects whose contributions are supported by research funding also face sustainability concerns, as participants lose motivation to contribute once funding runs out (Carver *et al*. 2022; Adema and Moore 2021). Once a project is completed, papers have been published, and credit has been allocated, it is common for scientists to mark their projects as finished and move on to new endeavours (Kelty 2008, 271-75; Howison and Herbsleb 2013). Scientists may archive their work in an institutional repository, at which point the work enters a stasis state that invites reference to the completed work, but which precludes any potential for the work to be updated, modified, or directly built upon. Open-source projects, on the other hand, are motivated by a more practical need for the software to function properly in perpetuity, and contributors may remain actively or sporadically involved to satisfy users' needs, or to direct users to derivative and functional forks of abandoned software (Kelty 2008, 278-81; Coleman 2012, 116-22; Hippel and Krogh 2003).

The adoption of open-source development models among archaeologists is generally informed by the broader open science movement. However, the predominant concern with implementing best tools to use, adopting optimal data processing pipelines, and tying into global, web-based infrastructures, protocols and standards (cf. Kansa *et al*. 2014; Kintigh *et al*. 2015; Roosevelt *et al*. 2015) distract from fundamental tensions and contradictions regarding the actual value of working in the open. For instance, Faniel *et al*. (2013, 299-301), Atici *et al*. (2013, 676-77), Huggett (2018; 2022), Sobotkova (2018), Opitz *et al*. (2021), Hacıgüzeller *et al*. (2021), and Batist (2023) demonstrate that to make the reuse of

archaeological data feasible and useful in a practical sense, it is necessary to re-introduce social friction that these infrastructures are designed to eliminate. In other words, the pressures and circumstances of being an archaeologist and doing archaeological research - such as the inherent subjectivities involved in characterising finds, or the reliance on analysts' reputation and academic pedigree to establish trust in the data they produce - assert themselves when attempting to make practical use of these infrastructures, and therefore must be accounted for in their design and implementation. In this article we draw attention to similar sources of dissonance with regard to the promise, potential, and actual implementation of open-source software development models among archaeologists by analysing the collaborative milieu in which archaeological software are actually being built and relating our findings to broader observations about how archaeologists tend to collaborate in practice.

2.3 Git and GitHub

Open source is an inherently internet-based development model and is supported by technical infrastructures that facilitate global distribution of labour and code. Here we provide a brief overview of key technologies that archaeologists have come to rely on as they develop open-source software. See Table 1 for a glossary of the git-, GitHub- and software engineering-related terminology which we use here and throughout this article.

Table 1: Glossary of git and GitHub terminology

Term	Definition
CodeBerg	Open source alternative to GitHub
Comment	On GitHub, text post attached to an issue, including the first one that describes the issue
Commit	Set of changes (addition, alteration, or deletion) to files in a repository that has been recorded by git as one entry in its log
Commit access	Ability to make changes to a repository directly, without making a pull request
Contributor	User that has made at least one commit to a specified repository
Follow	Add activity by another user to a user's timeline
Forge	Web-based platform for hosting, distributing and facilitating collaboration on version-controlled computer code, e.g. GitHub, GitLab, Codeberg
Fork	Copy of a repository owned by another user; forking is a prerequisite to making a pull request
git	Open source version control software

GitHub	Commercial platform that freely hosts git repositories and provides extended collaboration and social networking features, such as pull requests, issues and stars
GitLab	Open source alternative to GitHub
Issue	Feature of GitHub that records and tracks a bug report, feature request or other suggestion in a repository
Maintainer	Individual that has overall control of a repository, generally assumed to be its primary contributor. Repositories can have multiple users with commit access in addition to the maintainer
Merge	Accept a pull request and incorporate its changes into a repository
Organisation	Entity representing a group of users, which can also own repositories
Pull request	Mechanism by which users that don 't have commit access to a repository can contribute to it. The repository's maintainer or another user with commit access must decide whether to merge (accept) the changes, or decline them
Repository	Individual project that uses git for version control. Can include a mix of different types of files
Star	GitHub's version of a 'like ', applied by users to a repository
Timeline	Chronological feed of GitHub activity from repositories a user has starred and other users they follow. Also includes repositories that a user is not following if they are 'trending' or determined relevant by GitHub's algorithm
User	On GitHub, an individual with an account that can own repositories
Version control	System for tracking changes (additions, alterations, or deletions) in a set of files, typically but not exclusively computer code

Chief among these is git, a protocol designed to facilitate open and distributed contributions to a common code base. It operates by providing mechanisms for synchronising communal, web-based public repositories with local iterations stored on contributors' private workstations. Contributors who volunteer or are assigned to develop, inspect, or revise a specific aspect of a code base download a copy of the public repository into their own work environment, create a fork in which they apply their modifications, and then request that their fork be merged into the central code base. After a public repository's maintainers decide to merge the proposed changes into the communal code base, other developers may use git to download these changes while keeping their own independent forks intact.

Git is also designed to facilitate code review and version control. All modifications are tracked as 'diffs ', which highlight additions or deletions to the code base, including changes within individual files. Typically, a contributor will group a series of changes into a more comprehensive 'commit' based on a specific task or part of a workflow. Commits are always accompanied by a message, in which the contributor (ideally) describes the reason and context for the changes included in the commit. Moreover, git assigns each commit a unique identifier and identifies the contributor by name and email address to ensure some degree of public accountability.

Software forges - collaborative web platforms like GitHub, GitLab and Codeberg - are designed to facilitate open-source software development by hosting public git repositories. However, they also support common software developer and project management practices, such as issue and bug tracking, code-commenting, task management, identity and permissions management, web publishing, vulnerability detection, creation and maintenance of metadata, and financial sponsorship.[2] These platforms also implement standard social media functions, like the ability to follow projects and individual users to receive updates on their activities, 'star' certain repositories as a combined bookmarking and 'like' feature, and maintain a public-facing profile that includes personally identifying information (e.g. profile picture, username, real name, employer or affiliation), references to all public activity on the platform, and links to the user's other social media profiles. Code-sharing platforms thus serve as comprehensive developer portfolios and community networking resources. While these additional features are meant to complement and enhance the experience of contributing to open-source projects, they are not actually part of the git protocol.

As a concrete example of how git and GitHub is used by archaeologists, we can take https://open-archaeo.info itself. The website at that address was at the time of writing generated from a *git* source repository containing the data on the individual entries, a set of HTML templates, and some R scripts that translate between them. A copy of this repository can be downloaded and worked on locally by anyone, who will have access not just to the current state of the source code but its full history through the git *version control* software. The authoritative version is hosted on *GitHub* at https://github.com/zackbatist/open-archaeo. The GitHub URL indicates the primary *maintainer* of the repository, 'zackbatist' - that is, one of us (ZB) - who created and has ultimate control over it. However, various other *contributors* (such as JR) have added entries, corrected entries, or added functionality to the website using the *fork* and *pull request* features of GitHub. The basic unit of this type of contribution is the *commit*, which is a discrete set of changes (e.g. adding an entry) associated with one contributor at a single point of time. Others have contributed by raising *issues* describing problems with or suggestions for the project, leaving *comments* on these issues, or more loosely by using GitHub's social media features (*stars* and *following*). The full history of all these types of contributions to *open-archaeo* can be accessed through GitHub - or, as we use here, its programmatic API. This basic workflow is the same whether the project in question is a document like open-archaeo, or a piece of software, or a research paper; though actual patterns of collaboration vary markedly, as we will see.

3. Data and Methodology

We present an exploratory quantitative analysis of *open-archaeo* (Batist and Roe 2023), a directory of 493 pieces of open-source archaeological software and other digital resources maintained primarily by one of us (ZB) since 2018.

We compiled the dataset by browsing collaborative software development platforms, relying heavily on their social networking features. More specifically, we update *open-archaeo* by manually crawling through archaeologists' profiles on these platforms, as well as on other personal, professional, and institutional websites that describe and host additional archaeological software. We supplement this quasi-systematic collection strategy with word-of-mouth contributions made by interested individuals, who reached out via email, social media or at conferences to identify relevant work that we initially overlooked, including work that they created themselves.

Open-archaeo is a relatively comprehensive list. While our initial intention was to only list open-source software, its scope has expanded to include all software created by and for archaeologists. Apart from regular updates by its primary maintainer (ZB), it has been

expanded by a wider network of contributors and has benefited from the wider range of domain specialisms this has brought. However, *open-archaeo* generally lacks software written before archaeologists started using collaborative software development platforms such as GitHub, and software that is not shared on the web at all. It also includes numerous non-software resources, as well as software developed and distributed without the use of software forges. Open-archaeo is also limited by the experiences of its primary maintainers, which affects the dataset's overall scope and how comprehensively it covers various domains of archaeological research. [3]

Table 2: Software forges used by open archaeology projects

Host	n	%
GitHub	410	83.0%
Codeberg	16	3.2%
GitLab	6	1.2%
Bitbucket	1	0.2%
Launchpad	1	0.2%
None	60	12.1%

Where applicable, we obtained more detailed information about each repository's contents and contribution histories from the GitHub API (application programming interface). Our analysis incorporates data on 407 repositories,[4] 145548 commits, 1920 issues/pull requests, and 22303 comments from 561 distinct users, as well as repository metadata on programming languages used, licensing, stars and forks, and so on.

We opted to collect repository data only from GitHub because it is the most popular forge platform used by *open-archaeo* projects (Table 2). This means that we excluded projects that do not use version control (12% of the total) or that develop and host code on platforms other than GitHub (5% of the total) from the parts of the analysis that examine *how* archaeologists develop software. However, we were still able to draw from all records to ascertain the general composition of open-archaeo, and by extension, to address *what* kinds of software and resources archaeologists make.

That being said, we cannot account for practices that occur through offline or private channels, or forms of collaboration we do not know about. We did not directly observe or interview archaeological software developers, though our conclusions do draw heavily from our experience as members of that community ourselves. [5] Our earliest data are from 2005

and our study can say little about collaborative software development in archaeology before this point, though we know there was a significant amount of it (Ducke 2013; Whallon 1972).

These caveats notwithstanding, the *open-archaeo* directory and the supplemental data from the GitHub API provide a rich resource to explore the nature of collaborative software engineering in archaeology. Here we employ exploratory data analysis (*sensu* Tukey 1977) to identify and describe overall patterns visible in this rich dataset. In Section 4, our focus is on examining the general state of open-source archaeological software and resource development. In Section 5, we refine our analysis to examine development processes, with specific focus on collaborative experiences. Finally, in Section 6, we apply network analysis methods to investigate the formation of broader collaborative communities. Our analyses combine to support our objectives of understanding what kinds of software and resources archaeologists are making, how they create these tools in response to specific needs and use-cases, and how this work is situated within the context of an emerging community of practice.

The quantitative analyses and figures presented here were generated with R version 4.3.1 (2023-06-16) (R Core Team 2023). The full data and code are available in the compendium that accompanies this article (Roe and Batist 2024).

4. Open Archaeology

As of writing, *open-archaeo* catalogues 493 resources created by and for archaeologists. It includes both software and documents, but not research compendiums (collections of digital resources, including data, code, and documentation, which accompany or enhance a scientific publication; see https://research-compendium.science.).[6] Each record in *open-archaeo* is assigned to a category based on how the tool or resource is meant to be accessed or used, and is annotated with tags that describe what aspect of archaeological research each item was meant to address. Tags are ascribed based on how developers identified their projects' purpose and scope, and each record can have multiple tags. See Batist and Roe (2023) for a more comprehensive overview of the tags and categories applied to open-archaeo.

Table 3: Categories of open archaeology projects

Category	Scope	n	%
Software			
Packages and libraries	Sets of functions assembled with clear purpose, and made accessible using standards established by an underlying platform.	223	45%
Standalone software	Software that may be operated without needing to first access an underlying platform.	71	14%

Scripts	Sets of pragmatically assembled mutable functions, often lacking complete documentation or adherence to protocols that would otherwise facilitate secondary use outside their original contexts of creation.	65	13%
Documents			
Lists and datasets	A series of consistently organised observations assembled with purpose.	76	15%
Guides	An educational resource or documented protocol meant to instruct readers how to apply relevant tools or techniques.	29	6%
Products	Stable outcomes of creative work.	15	3%
Specifications, protocols and schemas	A formal data structure or framework intended to be used as a model.	14	3%

Table 4: Platforms and programming languages used by open archaeology projects

Platform	n	p
R	200	68.5%
Python	43	14.7%
QGIS	15	5.1%
Mobile app	7	2.4%
MATLAB	6	2.1%
ArcGIS	3	1.0%

LibreOffice Calc	3	1.0%
Microsoft Excel	3	1.0%
Blender	2	0.7%
Open Data Kit	2	0.7%
Other	8	2.7%

From our breakdown of *open-archaeo* by category (see Table 3), we can infer the prevalence of various development models, and the requisite technical capabilities that developers assume users hold. Most resources (45%) included in *open-archaeo* are designed to be used with an existing 'platform' - for example a package that extends a programming language (e.g. radiocarbon calibration is implemented in R in the package 'rcarbon' or Python in the package 'iosacal ') or a plugin for an application (e.g. 'ArchaeoAstroInsight' adds tools for measuring astronomical alignments to QGIS). Essentially such projects create additional functions within the base platform that are useful for archaeological purposes. Others create standalone software (14%) that can be run independently of such platforms, for example desktop or web apps. A significant number of projects also comprise datasets (15%) and non-packaged code snippets (13%) that have been made available for general use.

Some 41% of all projects are extensions to the statistical programming language R, making it the most widely used platform by a large margin (Table 4). Python, another programming language, is also relatively popular (9%), as are plugins for the open source geographic information system QGIS (3%).[7] Beyond that, there is a rather fragmented landscape of plugins for other desktop software (e.g. AutoCAD, ArcGIS), a number of lesser used programming languages, and a genre consisting of custom forms and spreadsheet templates. Many of these are targeted by only one or two developers; the larger platforms tend to be more diverse.

At first glance, the relative popularity of R versus Python is perhaps surprising; Python is regularly ranked as the most popular programming language in the world, with R a distant runner-up. However, it accords with the popularity of R as a tool for data analysis in archaeology (Schmidt and Marwick 2020) and other scientific disciplines (Lai *et al*. 2019).

Our analysis of thematic tags highlights aspects of archaeological work that software developers are inclined to contribute to (Table 5). The most common themes are work that naturally benefits from advanced information processing afforded by computers, such as statistical analysis, sample calibration, geographical analysis, data management, and chronological modelling. Educational resources and practical guides are also well represented owing to the web's usefulness as a medium for sharing and communication.

When we compare categories with thematic tags, we see the general domains that each kind of resource is designed to serve. We see that packages are fairly common across the board. Tags that are notable for having a higher proportion of standalone software include archaeogenetics, data management, 3D modelling, photogrammetry, drivers and IO, and

simulations or agent based modelling. These tools may require greater access to system resources, or may require more complex user interfaces than what R or Python IDEs (integrated development environments) tend to provide.

Table 5: Themes of open archaeology projects. See Batist and Roe (2023, table 1) for a description of each tag's scope. [ONLINE ONLY]

To enact their mandate of ensuring that anyone can access and modify software and other creative works, the open source and open science movements encourage developers and scientists to adopt open licenses. Licenses are legally binding statements that stipulate how a creative work can be accessed and used. Proprietary licenses usually require explicit permission to be granted so that the work can be accessed or modified, usually in exchange for financial compensation. Open licenses, on the other hand, are more permissive, and allow anyone to use creative works without such harsh restrictions. While it is certainly possible to write your own license, it is very common to simply use one of several standardised open licenses (see choosealicense.com). Some licenses, like GNU, MIT and Apache, are explicitly suited for distributing software, and specify certain use cases that are afforded by digital media. Other licenses, like the Creative Commons variants, are more suited to other kinds of creative works such as books, articles, movies, music, photographs, and websites. The Creative Commons licenses also include clauses that cater to academic or creative sensibilities, such as requirements to attribute credit to the original authors, to restrict commercial use, and to propagate similar restrictions in derivative works.

Table 6: Licenses used by open archaeology projects on GitHub

License	n	%
None detected	245	49.7%
GPL	123	24.9%
MIT	77	15.6%
CC0	12	2.4%
CC-BY	8	1.6%
Apache	7	1.4%
AGPL	5	1.0%
Unlicensed	4	0.8%

CC-BY-NC-SA	3	0.6%
CC-BY-SA	3	0.6%
CECILL	2	0.4%
BSD-3-Clause	1	0.2%
GFDL	1	0.2%
MPL	1	0.2%
ODbL	1	0.2%

Roughly half of *open-archaeo* repositories are accompanied by an explicit license (Table 6). Two common free software licenses account for the majority of repositories that do contain licenses: the GNU General Public License (GPL, 52%) and the MIT License (31%). These differ primarily in the restrictions they place on reuse: the MIT License aims to be maximally permissive, while the GPL is a 'copyleft' license specifying that all derivative works must be distributed under similar terms (in other words, it prohibits the use of open-source software within non-open software (Dusollier 2007). Interestingly, archaeologists' preference for the more restrictive of these two licenses is the reverse of the general trend seen in open-source projects on GitHub (Balter 2015). Creative Commons licenses are a distant third place (10% of repositories), in contrast to their widespread use for other forms of scholarly output (Kim 2007). Many repositories do not specify a license; given a documented misconception among academics that GitHub can serve as a sustainable and long-term code and data hosting platform (Milliken *et al*. 2021; Escamilla *et al*. 2022; 2023), it is possible that many maintainers whose work is included in *open-archaeo* similarly assumed that making their work available, without explicitly stating permissible use, is enough to allow unrestricted access to the repository's contents. However, we cannot verify this potential explanation given the methods we currently employ, and more discursive qualitative research is needed to explore the rationales behind such decisions.

Archaeological software development activity has increased significantly over the years. Figure 1 shows the cumulative growth of code contributions committed and pushed to GitHub repositories, and the number of GitHub repositories that host archaeological software and resources.

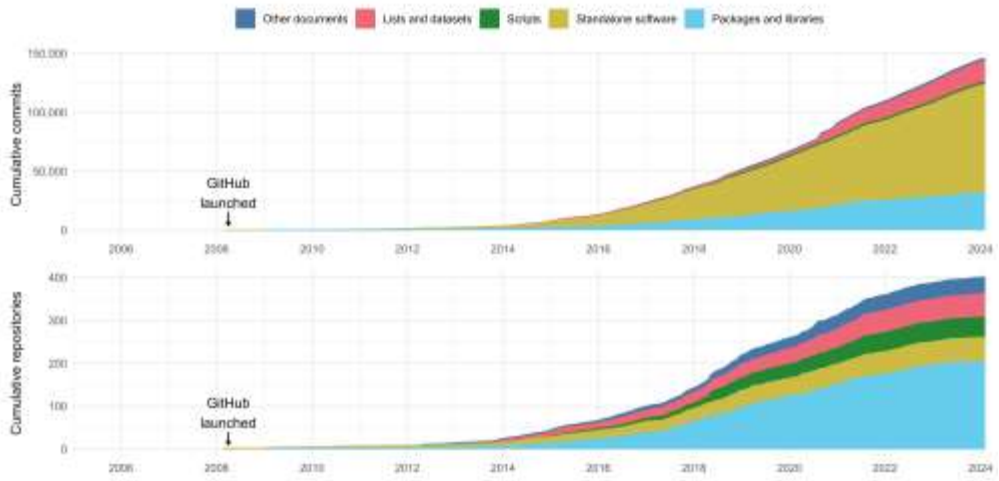

Figure 1: Growth of open archaeology projects on GitHub

Archaeologists have been using git from at least the late 2000s, shortly after GitHub was launched,[8] but it saw a marked increase in popularity *c*. 2014-2015. From that point and until recently there was an exponential uptake of GitHub by archaeologists, but while we were preparing this article (*c*. 2022) the first signs of a slowdown in growth have appeared, with the number of cumulative commits continuing to rise but the number of repositories hitting a sharp plateau. This could be explained by market exhaustion, a shift in emphasis to maintaining existing code and working on established projects, and/or growing doubts about the appropriateness and sustainability of GitHub following its acquisition by Microsoft (Kansa 2022). [9]

GitHub's entry into the digital archaeology mainstream in 2014-2015 also marks the point at which we see it being used for things other than packaged source code (e.g. documents and scripts). This suggests that the 'early adopters' of GitHub, pre-2014, were more directly embedded in existing (open source) software development communities, while those that came later also saw version control systems as a potential medium for dissemination and archiving. It may also reflect a general move towards git- and GitHub-based workflows by archaeologists attracted to open, participatory, and/or generally 'nerdy' academic practices.

5. Collaborative Practices

As well as hosting source code, GitHub and other software forges include systems for facilitating collaboration on code and other projects. The basic collaborative workflow is inherited from git, which allows multiple users to commit code to the repository (see Table 1 for definitions of this and other git terminology used in this section). A user with commit access to a repository can change any of its contents at will, so this is usually reserved for the project maintainer and known, trusted, collaborators. GitHub extends this model with its pull request feature, by which any user can fork a repository to which they don 't have commit access, make changes, then offer to contribute those changes back to the original repository. The maintainer can choose to merge (accept) or decline the pull request, facilitating contributions from a wider network of collaborators without the need for permission to be sought in advance.

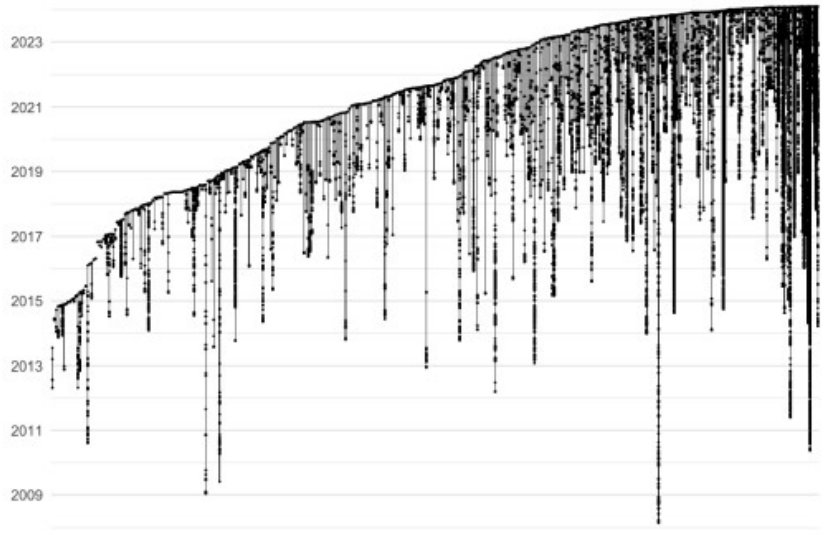

Figure 2: Lifespan of open archaeology repositories. Each point indicates a commit; excludes repositories with only one commit

We measured the lifespan of a repository as the time between the first and latest commit, and its activity as the number of commits per day. Here, therefore, we refer to the *development* lifespan of a project, which is not necessarily related to its use-life. By these metrics, the lifespan and activity of repositories in *open-archaeo* vary greatly (Figure 2). The average project lasts 920 days with 0.76 commits per day. Many projects are active for only a short period of time: about 17% less than 30 days, 26% less than 90 days, and 38% less than a year. However, the vast majority (all but three) do have more than one commit, suggesting that use of GitHub as a pure host for already finished projects is not common; some degree of iteration, if not collaboration, is almost always present. The longest-lived projects have been active for between 10 and 17 years. The most active projects see up to 13 commits per day, but the majority of repositories (84%) receive less than one commit per day.

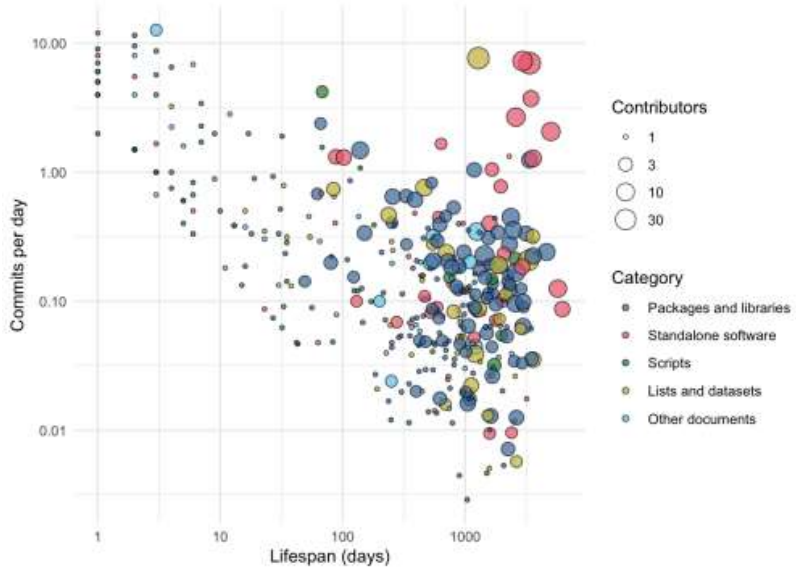

Figure 3: Lifespan and commit rate of open archaeology repositories

The interaction between project longevity, activity, and number of contributors is multifaceted (Figure 3). Highly active projects (one commit per day or more) tend to be either very long-lived or very short-lived; few fall in the centre of the distribution. Short-lived projects tend to be characterised by a 'spree' of activity (a high commit rate), while long-lived projects have a broader range of activity profiles. The most 'successful' projects according to open-source norms (i.e. long-lived and active) are, with few exceptions, those projects with the largest contributor base in our dataset. However, the modal project in the centre of the distribution is more modest, lasting around three years, maintained by an individual or a small group, with around three commits per month.

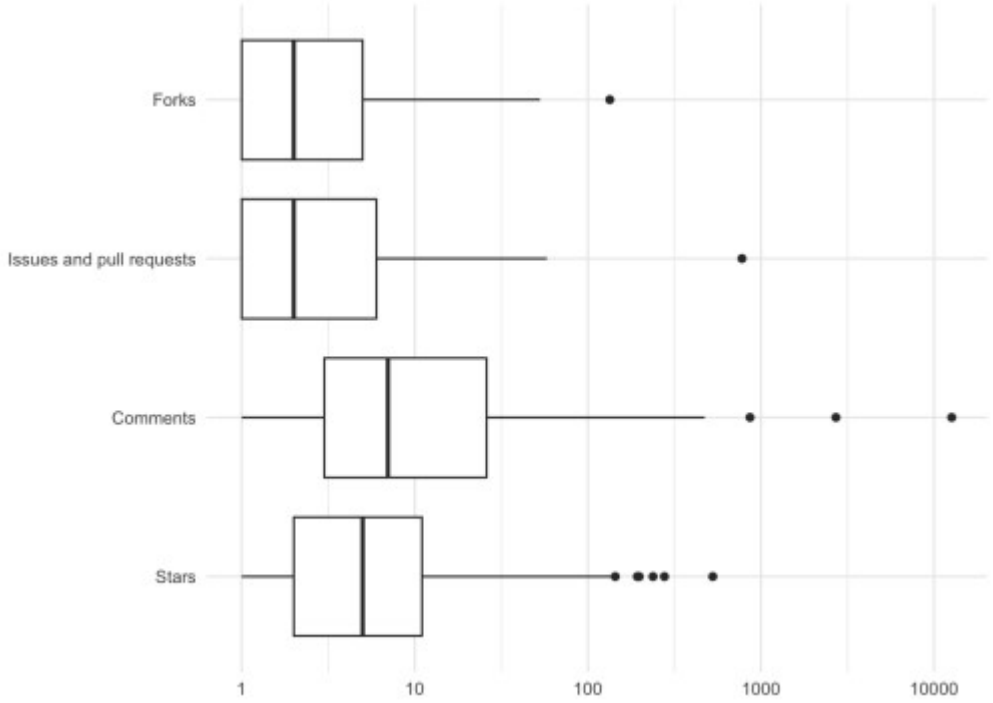

Figure 4: Box plot showing use of GitHub collaboration features in open archaeology repositories

GitHub also facilitates collaboration on broader project management tasks, primarily through its issues feature.[10] Unless a repository's maintainer specifically configures it otherwise, any user can create an issue attached to another user's repository, or comment on an existing issue. Issues are typically used to log and track bug reports, feature requests, and other comments and suggestions from the project's user base. GitHub's pull request feature is also implemented via this system - a pull request is a special type of issue. According to the data we collected from the GitHub API, these features are not widely used by *open-archaeo* projects (Figure 4). Only 46% of repositories have been forked at least once and only 38% of repositories make use of issues/pull requests. Those repositories that do use issues do not use them very extensively; 33% have only one issue and 85% have ten or less.

Another way GitHub users can engage with repositories and other users is with social media-like features such as starring a repository, commenting on an existing issue, or following a user. These actions populate a timeline of through which users can see recent activity and discover new projects related to those they have interacted with in the past.[11] While not as direct a contribution as pull requests or issues, these features can facilitate the formation and maintenance of collaborative networks, in the same way that other social media platforms serve other professional networks. These features are used more widely than forks, issues and pull requests (Figure 4): 83% of repositories have at least

one star and, in those repositories that use issues, 33% of them received at least one additional comment.

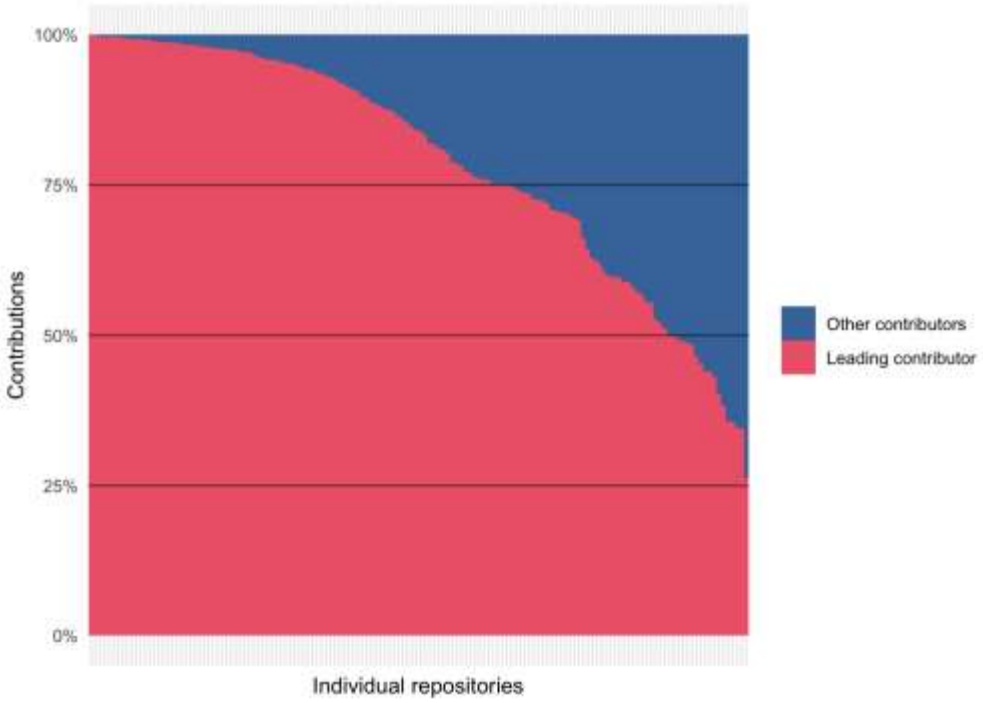

Figure 5: Distribution of contributions in multi-contributor open archaeology repositories

Perhaps unsurprisingly, given the low uptake of GitHub's collaborative features, 62% of *open-archaeo* repositories only contain commits from a single user. Even in the minority of projects that have more than one contributor, work (as measured by number of recorded commits) is distributed highly unevenly (Figure 5). The lead maintainer almost always does the lion's share of the work: they are responsible for more than half of commits in 88% of projects and more than three-quarters in 61%. This may be attributed to the steep learning curve commonly attributed to working with git. While git can be a great way to track changes and manage distributed contributions to a common code base, it can also be unwieldy in situations when multiple users (especially those with less experience using git for collaborative purposes) are expected to contribute within short spans of time. This corresponds with our prior observations that projects exhibiting higher commit rates have fewer contributors. Additionally, our analyses neglect to account for contributions that are not tracked via git or GitHub. Those who do not code may provide creative guidance or feedback during in-person meetings, via email, or using alternative online messaging or social media platforms. A more focused qualitative assessment of these non-coding and supportive work practices would shed more light on the totality of effort that goes into producing and maintaining open-source projects.

The prototypical open-source project comprises a core group of developers (often a single maintainer) that regularly commit new code, a wider network of collaborators that contribute through forks and pull requests, plus an active user base that create and comment on issues, who have indicated their support for the project by starring its repository. It is unclear whether archaeological software developers actually aim to operate following this model, or whether it is even suitable for supporting what open science aims to achieve. However, it is clear that only a small number of *open-archaeo* projects operate according to this model. The majority of projects are in fact short-lived, with few contributors and a small number of commits. Use of GitHub's collaboration features is also generally low (Figure 4), although the data also show a divergence between the uptake of features that facilitate direct code

contributions (forks, issues, pull requests), which have markedly zero-skewed distributions, versus more indirect, social media-like features (comments, stars), which are moderately well-used.

These findings show a preference for passive/reactive rather than active/proactive engagement with others' work, which is not conducive to achieving the desired outcomes of opening source code, namely enabling greater engagement, reuse and critique. While limited time or technical capability may be contributing factors (which should be targeted in more focused investigations), we believe that social norms, such as expectations and taboos surrounding the permeability of a grant-funded project's boundaries, or whether it is proper to actively engage or interfere with work directed under the aegis of another project, play a very significant role. Our network analysis of collaborative ties, presented in the following section, corroborates this claim.

6. An Emerging Community of Practice?

By contributing to shared repositories - whether with code (commits), issues, or comments - archaeologists using GitHub form a collaborative network that we can map using data from the GitHub API. Here we consider two facets of this network: repositories connected by common contributors (the repository-repository graph, Figure 7), and users connected by contributions to common repositories (the user-user graph, Figure 8). In both cases, number of contributions constitutes a natural measure of the strength or weight of the connection, which can be further broken down by type of contribution (commit, issue/pull request, or comment). We identified clusters using the edge-betweenness community detection method, which operates by locating the edges that are situated along the shortest paths between all pairs of nodes, and which therefore exhibit high betweenness centrality; these edges form indirect links between otherwise completely unconnected sections of the network, and as such are the loci that distinguish groups of nodes that are more highly connected to each other than they are to others (Girvan and Newman 2002).

Figure 6: Graph of open archaeology repositories and users connected by contributions. Darker edges indicate a great number of contributions. Node colour indicates membership of the largest clusters according to the edge-betweenness method (Girvan and Newman 2002). Excludes isolate nodes. [ONLINE ONLY]

Figure 7: Graph of open archaeology repositories connected by common contributors. Darker edges indicate a great number of common contributors. Node colour indicates membership of the largest clusters according to the edge-betweenness method (Girvan and Newman 2002). Excludes isolate nodes. [ONLINE ONLY]

Figure 8: Graph of open archaeology users connected by contributions to common repositories. Darker edges indicate a great number of common repositories. Node colour indicates membership of the largest clusters according to the edge-betweenness method (Girvan and Newman 2002). Excludes isolate nodes. [ONLINE ONLY]

Our data show that there is a significant network of archaeologists collaborating on GitHub. In total, 67% of repositories and 88% of users in our dataset are connected to at least one other repository or user. Of these, 94% of repositories and 80% of users belong to a single connected subgraph (Figure 6 and Figure 7). This indicates that most repositories that have had more than one contributor, or whose contributors have worked independently on more

than one repository, are not isolated, and that at least one of their contributors have, at least one time, worked with members of a broader interconnected population.

We delimited 63 distinct clusters that outline the topography of the repository-repository network (Figure 7). While many of these clusters are interconnected, some discrete components containing between 2-20 repositories appear as distinct from a primary core. The core cluster is characterised by repositories whose contributors commit to projects other than their own, and it includes a smorgasbord of projects whose contributors share varied interests.

Clustering also reveals distinct collaborative networks within the user-user graph (Figure 8). We again see a complementary primary interconnected subgraph and several unconnected subgraphs. The primary subgraph comprises several connected clusters, including one central cluster and several more peripheral clusters, which are internally-cohesive and exhibit few connections with other peripheral clusters. The central cluster bridges all the peripheral clusters. Moreover, the central cluster is not uniform, and comprises several relatively discrete components representing collaborative sub-communities. While these components are internally cohesive, they exhibit enough connections to other members of the central cluster to preclude them being considered as separate or peripheral clusters.

In both the repository-repository and user-user networks, the peripheral clusters correspond with either the connections surrounding specific projects or the series of repositories created by single individuals and sometimes also their close colleagues. On the other hand, the central cores exhibit greater internal variety that may correspond with social connections and the formation of a complex software development community. This is evident through the fact that many of the connections represented in the cores emerge from more conventional professional networks, e.g. the ISAAKiel group based at the University of Kiel or CAA-SSLA, a special interest group of the international scholarly society 'Computational and Quantitative Applications in Archaeology' (CAA) focused on scientific programming, and which clusters around users 'nevrome' (Clemens Schmid) and 'martinHinz' (Martin Hinz), who are core members of both organisations. Peripheral clusters that are connected to the central core by only a few relationships represent the sole (or perhaps initial) integration of lone developers into a broader community.

Table 7: Repositories ranked by centrality to the repository-repository network. Centrality is measured by node betweenness weighted by number of contributions

Rank	Repository	Category	Tags	Commits
1	benmarwick/ctv-archaeology	Lists and datasets	Lists	688
2	zackbatist/open-archaeo	Lists and datasets	Lists	360
3	ahb108/rcarbon	Packages and libraries	Radiocarbon dating, calibration and sequencing	881

4	ropensci/c14bazAAR	Packages and libraries	API interfaces and web scrapers; Radiocarbon dating, calibration and sequencing	1057
5	ropensci/neotoma	Packages and libraries	API interfaces and web scrapers; Palaeoclimate modelling	809
6	lakillo/archaeology-machine-learning	Lists and datasets	Lists; Machine learning	62
7	ekansa/open-context-py	Standalone software	Platforms and publications	4575
8	demjanp/Res14C	Packages and libraries	Radiocarbon dating, calibration and sequencing	8
9	paleolimbot/tidypaleo	Packages and libraries	Data management; Palaeoclimate modelling	168
10	dainst/idai-field	Standalone software	Data management	21407

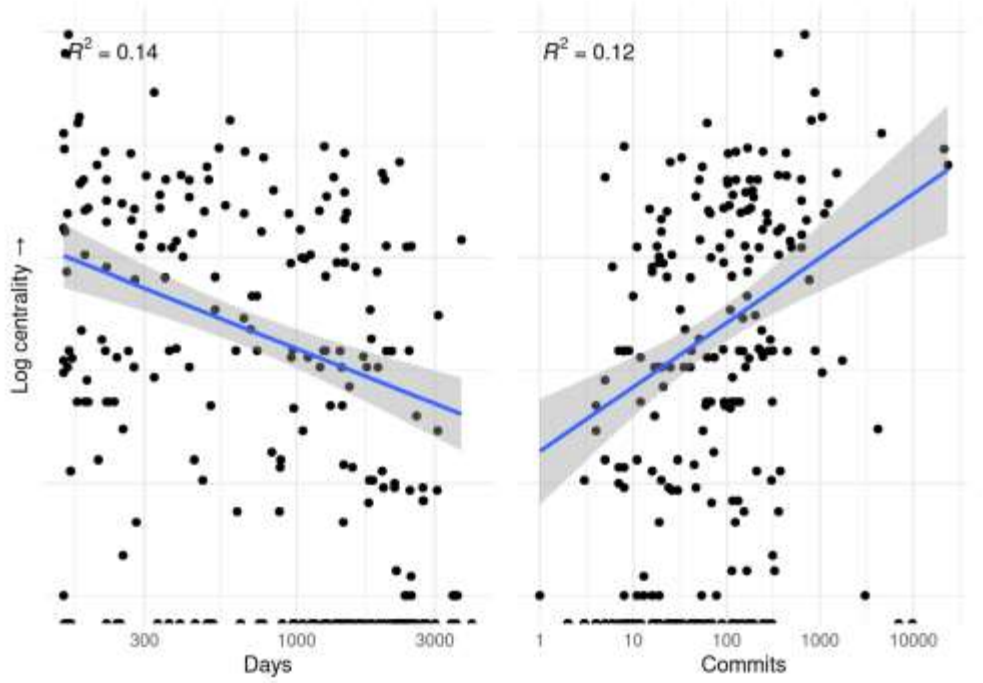

Figure 9: Repository centrality by age (left) and length (right). Centrality is measured by node betweenness weighted by number of contributions. Blue line indicates a generalised linear regression, with confidence envelope shaded in grey

The repositories most central to the network as a whole (Table 7) include three lists and directories, including *open-archaeo* itself. Three relate to making large data repositories accessible for analysis, and one is a very well-supported field recording application. Community input is therefore centred on infrastructural projects, including those that index and publicise available tools and resources. Moreover, three relate to radiocarbon data modelling and two relate to palaeoenvironment reconstruction, which reflects the fact that these have long been prominent foci of statistical software development in archaeology.

Repository centrality is predicted by the total number of commits it has received but, somewhat surprisingly, younger repositories rather than older ones tend to be more central (Figure 9). Tentatively, we interpret this as an indication that the network has become more connected over time, but we leave a fuller analysis of temporal trends in collaborative activity to future work.

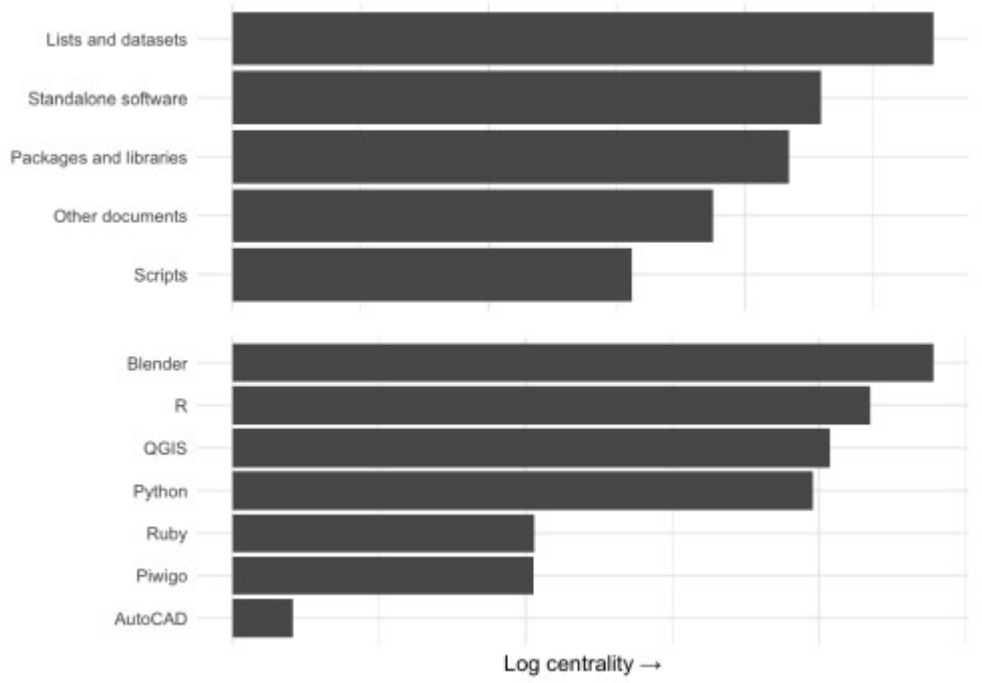

Figure 10: Mean repository centrality by category (top) and platform (bottom). Centrality is measured by node betweenness weighted by number of contributions

When comparing across categories and platforms, the highest mean centrality is seen in repositories that contain lists and datasets, standalone software, or packages and libraries, and in repositories based on Python, R or QGIS (Figure 10). Interestingly, these trends depart from the observed popularity of different categories and platforms in the *open-archaeo* dataset as a whole (see Section 4): standalone software is more central than packages/libraries, even though there are more of the latter by a significant margin. This may be due to the fact that many packages are developed to support specific practices or use-cases (often inspired by personal need), or are designed to run relatively stable statistical functions that need not change over time. These are therefore relatively stable and require little additional input after release. On the other hand, standalone software tend to integrate multiple system components and may evolve over time to add new features or support new workflows. Moreover, standalone software are generally rooted in longer-term and community-held objectives, and their development may therefore be backed by institutions with funding and resources to support developers.

Despite being a minority language, Blender packages are more central than all other package platforms on average, but this is a statistical anomaly caused by uneven sampling (only two blender packages, maintained by a single individual, are accounted for). R is naturally the platform with the next highest average centrality since it serves as a *lingua franca* that draws developers from across the discipline. Many of the QGIS plugins add various specialised features to the extensible GIS platform, and are therefore developed by interdisciplinary teams, which explains its high rank. Python projects, which tend to relate to the development of information infrastructures or are of interest to members of other fields (e.g., palaeo-ecology, other fieldwork-based disciplines), are also highly ranked in terms of average centrality. Further analysis is warranted to qualify these observations regarding the significance of development patterns when working across different languages and platforms.

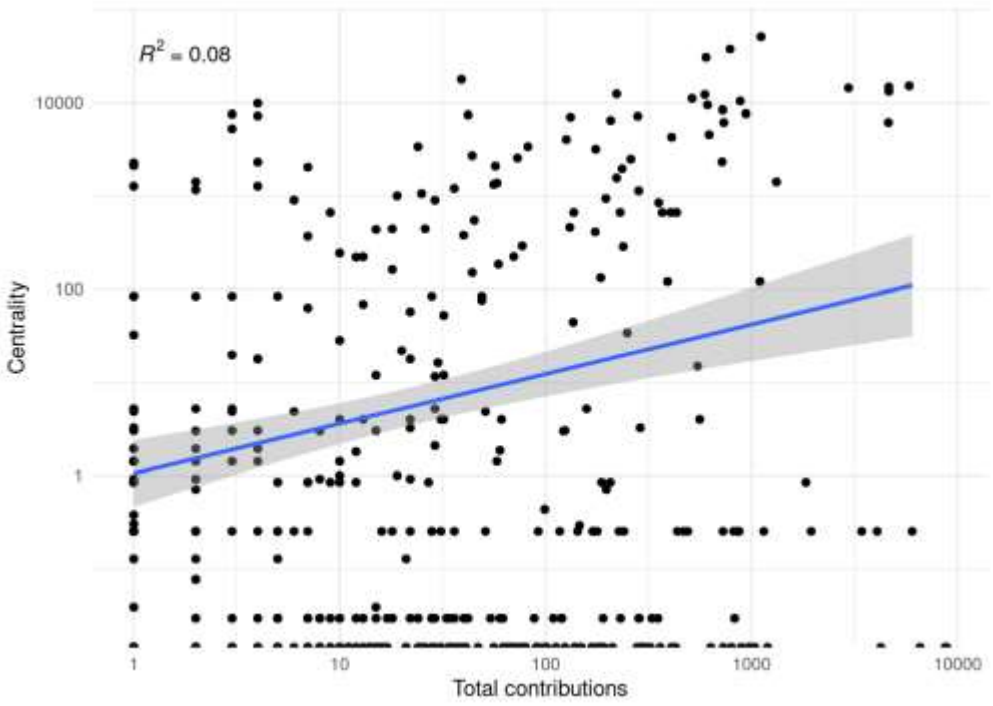

Figure 11: User centrality by total number of contributions. Centrality is measured by node betweenness weighted by number of contributions. Blue line indicates a generalised linear regression, with confidence envelope shaded in grey

Centrality to the user-user graph is weakly predicted by a user's overall rate of activity, as measured by their total number of contributions (Figure 11). We did not collect demographic data on users that appear in our dataset, but based on our own knowledge of the community we can observe that those highly central to the network tend to be employed in (junior) academic positions, or in a few cases in cultural heritage authorities, rather than specifically as research software engineers. Such positions tend not to actively reward or encourage software development, at least not on a par with more traditional academic outputs (Baxter *et al*. 2012), and are increasingly precarious (Cornelius-Bell and Bell 2021). This obviously poses a serious risk to the sustainability and growth of open-source software in archaeology: if the people who occupy central positions in the network cease to be active, then it is likely that the overall network would fragment. Assessing and mitigating this risk should be a high priority for future research in this area.

7. Conclusion

Our goal in this study was to investigate the under-explored research practices involved in research software engineering in archaeology. We sought to identify not only *what* kinds of software archaeologists are making, but *how* archaeologists create these tools as part of a broader community of practice. Our emphasis on the collaborative experiences involved in open-source software development emerged from our experience maintaining open-archaeo. We observed that making one's code openly available on the web does not necessarily garner the benefits often touted by open science advocates, namely that source code can be audited, forked, and appropriated for alternative use cases, which are effectively social and collaborative experiences.

To investigate these concerns, we operationalised open-source collaborative experiences as the use of certain features of git and GitHub visible to us in data from the GitHub API. With these data, we documented that open-source software development in archaeology has seen a rapid and sustained rise beginning around 2014 (Figure 1). This is marked by a variety of applications and use cases, including the use of git and GitHub to track and host content other than code. Moreover, archaeologists are very involved in broader scripting ecosystems, as is evident through the predominant creation of R packages and Python libraries designed to process the rich variety of archaeological information. At the same time, archaeologists also create standalone software for more intensive tasks that require greater access to system resources or that warrant more complex user interfaces than R and Python IDEs are capable of providing. These tools tends to be focused on various means of identifying distribution patterns (spatial, temporal, statistical), calibrating data obtained from various instrumental methods (XRF, luminescence dating), supporting specialised finds analysis (zooarchaeology, palaeobotany, archaeogenetics), and supporting the collection and processing of archaeological materials. These foci signify gaps in the archaeological toolbox that archaeologists recognised, and have attempted to fill, on their own terms.

There is an emerging community of practice around open-source research software in archaeology. All but a handful of the GitHub repositories we analysed have more than one commit, showing that archaeologists use it for ongoing work rather than merely to upload finished products. They relatively frequently make use of the 'star' and 'comment' features to engage with others' repositories (Figure 4) and, via these and other shared contributions, we can trace a collaborative network that includes the majority of archaeologists active on GitHub (see Section 6).

On the other hand, we found that the forms and intensity of collaboration remains limited. Most work is performed individually (Figure 5) and is short-lived (Figure 2 and Figure 3). The vast majority of repositories have 1-3 contributors, with only a few distinguished by an active and diverse developer base. Our analysis also shows an uneven use of git and GitHub's extended features, beyond their basic usage as a version control system and repository host. While GitHub's more passive collaborative features (stars, comments) are commonly used, those that involve direct engagement with repository content (issues, forks, pull requests) are not (Figure 4); perhaps because people do not want to 'step on toes' or be seen to be intruding on others' projects. This may relate to the fact that most developers on this list are academics who hold different values relative to the designers of open-source development environments, regarding how collaboration should occur, for example, when dealing with how projects and ideas are 'owned' by individuals or communities, and how work should be iteratively improved upon.

Our network analysis (Section 6) similarly draws attention to the real-world collaborative ties that underpin archaeological open-source software development. We identify a core cluster

representing a series of collaborative ties among members of an archaeological software engineering community of practice. This core exhibits complexity that corresponds with social patterns, such as the presence of various clusters representing interconnected interest or affinity groups. Indeed, we have inferred that 'real-world' social connections and institutional support structures are strong predictors of centrality, since these clusters are representative of established professional partnerships. This suggests that archaeological open source is firmly embedded within existing power structures that permeate academic life, both online and offline. Similarly, we found that the individuals who play critical roles in supporting the archaeological open source community are precariously employed workers. Far from open source being inherently distributed, resilient, and open-ended, this indicates that research software engineering is actually quite centralised, fragile, and based heavily on existing professional connections and endeavours.

These findings call into question the notion that archaeologists benefit from the positive outcomes that are commonly argued to be the natural results of open-source development models - namely, greater degrees of extensibility and participatory action. While opening the source code may facilitate these positive outcomes as necessary preconditional factors, we argue that this only amounts to establishing the *potential* for people to put these values into practice. We argue that the objectives and circumstances that frame archaeological practice significantly influence how far archaeologists (and academics in general) are willing to push for these values, and limit the ability for archaeologists to do open source in ways that resemble more mainstream open-source projects. For instance, successful open-source projects like the Linux kernel, openSSL, or the Firefox web browser are driven by collective and popular interest in ensuring that code remains functional, and the code base is therefore constantly in flux and bears an accumulating list of contributing members. This differs from the organisational principles that govern much archaeological work, namely where a director or directors (of a field project, research group, etc.) sets the goals and orientation of the group and commissions and manages other actors accordingly. Moreover, archaeological projects ultimately seek to produce stable textual outcomes bearing clear delineation of authorship and that require no upkeep whatsoever. Sustaining an open-source project is simply not compatible with the factors that currently drive the momentum behind archaeological work. This is compounded by the fact that many archaeologists consider software development as a form of support work and of lesser value than traditional academic research activities, and that research software engineers are often precariously employed (Baxter *et al*. 2012; Cornelius-Bell and Bell 2021).

As such, we advocate for more focused attention on specific disciplinary norms and institutional support structures that inform how knowledge is created and validated, and how varied contributions to the scholarly enterprise are mediated, credited, and valued (cf. Leonelli 2023; Bennett 2021; Khan *et al.* 2024). In other words, if we want to make open source effective in relation to the aforementioned goals of encouraging greater inclusivity, transparency, and productivity, we also need to foster a culture that supports active, pragmatic and humble critique, and which instils a de-territorialised attitude concerning what it means to contribute to collective knowledge (Morgan 2015; Ducke 2015). This means fighting against the pathological power-relations that scaffold all aspects of academic life, and not fooling ourselves into believing that technical solutions (i.e. using git) will, on their own, resolve the wicked social problems that lie at the heart of scientific research practice.

Data Availability

The results presented here are based on the directory of open archaeology projects maintained at https://open-archaeo.info (Batist and Roe 2023). The specific version we used is available from Zenodo at https://doi.org/10.5281/zenodo.10625236 as well as in the

compendium associated with this article (Roe and Batist 2024; https://doi.org/10.5281/zenodo.8393043), in CSV format under analysis/raw_data.

Further data on activity in the git repositories associated with these projects was obtained via the GitHub API. This data are available from Zenodo in the compendium associated with this article (Roe and Batist 2024; https://doi.org/10.5281/zenodo.8393043), in RData and CSV format under analysis/derived_data.

Acknowledgements

The impetus for this work was a session at the Computer Applications and Quantitative Methods in Archaeology (CAA) virtual annual meeting in Cyprus in June 2021. Our thanks to Martin Hinz, Clemens Schmid, Sophie Schmidt and other members of the Scientific Scripting Languages in Archaeology Special Interest Group (CAA-SSLA) for organising the session and encouraging us to develop the work further.

Our original manuscript benefited greatly from the insights of the anonymous reviewer for *Internet Archaeology*. We are also grateful to *IA* editor Judith Winters for managing the process and facilitating the interactive figures.

JR was supported by the Swiss National Science Foundation (SNSF Project #198152) for the duration of the production of this work.

Footnotes

1. Participatory or community-based research does exist, though, and is an exception to this generalisation (see Morgan and Eve (2012)). It requires active effort to do well and should not be considered a passive by-product of posting one's research on the web.←

2. Much of this relies on proprietary software owned and managed by commercial organisations, and there has been some controversy surrounding the take-over of open-source platforms by for-profit entities (see Saunders (2022) and Brembs *et al*. (2023)).←

3. We welcome anyone, especially domain specialists who are familiar with the kinds of tools commonly used in their specific fields, to help fill in these gaps. Instructions for contributing to *open-archaeo* can be found at: https://github.com/zackbatist/open-archaeo.←

4. We excluded three GitHub repositories from the analysis for technical reasons. For example, https://github.com/carpentries-incubator/R-archaeology-lesson is a repository within the scope of open-archaeo, but it was forked from a pre-existing repository that is not (https://github.com/datacarpentry/R-ecology-lesson) and thus includes in its commit history irrelevant data from the parent repository.←

5. JR has been active in the development of open source research software for archaeology for ten years. Through his maintenance of *open-archaeo* over the past 5+ years, ZB has developed an extensive understanding of the people who create archaeological software and the institutional ecology that supports their work.←

6. See https://github.com/benmarwick/ctv-archaeology for a similar list of archaeology publications that include R code←

7. While Blender and QGIS plugins are written using the Python language, our intent while categorising platforms was to get a sense of the developer ecosystems in which archaeological software engineers participate, rather than to simply gauge the popularity of different languages (Batist and Roe 2023, 2).←

8. We excluded commits that have obviously erroneous dates, e.g. 2001-01-01 in projects started in the mid-late 2010s.←

9. Concerns we very much share.←

10. Apart from issues, GitHub has a very wide range of project management and social media-like features, including wikis, discussion forums and 'kanban' boards. We have not analysed the use of these features here.←

11. This feature of GitHub's timeline was one of the primary ways we compiled open-archaeo.←

Bibliography

Adema, J. and Moore, S. 2021 'Scaling small; or how to envision new relationalities for knowledge production',*Westminster Papers in Communication and Culture* **16**(1). https://doi.org/10.16997/wpcc.918

Atici, L., Kansa, S.W., Lev-Tov, J. and Kansa, E.C. 2013 'Other people's data: a demonstration of the imperative of publishing primary data, *Journal of Archaeological Method and Theory* **20**(4), 663. https://doi.org/10.1007/s10816-012-9132-9

Balter, B. 2015 'Open Source License Usage on GitHub.com', *The GitHub Blog*, March 10, 2015. https://github.blog/2015-03-09-open-source-license-usage-on-github-com/

Batist, Z. 2023 *Archaeological Data Work as Continuous and Collaborative Practice*, PhD thesis, University of Toronto. https://hdl.handle.net/1807/130306

Batist, Z. and Roe, J. 2023 'Open-Archaeo: a resource for documenting archaeological software development practices', *Journal of Open Archaeology Data* **11**. https://doi.org/10.5334/joad.111

Baxter, R., Chue Hong, N., Gorissen, D., Hetherington, J. and Todorov I. 2012 'The research software engineer' in *Digital Research Conference*, Oxford 2012, Oxford. 1-3. https://www.research.ed.ac.uk/en/publications/the-research-software-engineer

Beck, A. and Neylon, C. 2012 'A vision for open archaeology', *World Archaeology* **44**(4), 479-97. https://doi.org/10.1080/00438243.2012.737581

Bennett, E.A. 2021 ;Open Science From a Qualitative, Feminist Perspective: Epistemological Dogmas and a Call for Critical Examination;, *Psychology of Women Quarterly* **45**(4), 448-456. https://doi.org/10.1177/03616843211036460

Brembs, B., Lenardic, A., Murray-Rust, P., Chan, L. and Irawan, D.E. 2023 'Mastodon over Mammon: towards publicly owned scholarly knowledge', *Royal Society Open Science* **10**(7), 230207. https://doi.org/10.1098/rsos.230207

Carver, J.C., Weber, N., Ram, K., Gesing, S. and Katz, D.S. 2022 'A survey of the state of the practice for research software in the United States', *PeerJ Computer Science*, 8:e963. https://doi.org/10.7717/peerj-cs.963

Coleman, E.G. 2012 *Coding Freedom: The Ethics and Aesthetics of Hacking*, Princeton University Press. https://doi.org/10.1515/9781400845293

Cornelius-Bell, A. and Bell, P. 2021 'The academic precariat post-COVID-19', *Fast Capitalism* **18**(1). https://doi.org/10.32855/fcapital.202101.001

Cowgill, G.L. 1967 'Computer applications in archaeology' in *AFIPS '67 (Fall): Proceedings of the November 14-16, 1967, fall joint computer conference*, New York: Association for Computing Machinery. 331-37. https://doi.org/10.1145/1465611.1465654

Dorta-González, P., González-Betancor, S.M. and Dorta-González, M.I. 2021 'To what extent is researchers' data-sharing motivated by formal mechanisms of recognition and Credit?', *Scientometrics* **126**(3), 2209-25. https://doi.org/10.1007/s11192-021-03869-3

Ducke, B. 2012 'Natives of a connected world: free and open source software in archaeology', *World Archaeology* **44**(4), 571-79. https://doi.org/10.1080/00438243.2012.743259

Ducke, B. 2013 'Reproducible data analysis and the open source paradigm in archaeology' in A. Bevan and M. Lake (eds) *Computational Approaches to Archaeological Spaces*, Walnut Creek, CA: Left Coast Press. 315-26.

Ducke, B. 2015 'Free and open source software in commercial and academic archaeology' in A.T. Wilson and B. Edwards (eds) *Open Source Archaeology: Ethics and Practice*, Warsaw, Poland: De Gruyter Open. https://doi.org/10.1515/9783110440171-008

Dusollier, S. 2007 'Open source and copyleft: authorship reconsidered?' in W.T. Gallagher (ed) *Intellectual Property*, London, UK: Routledge, 563-78. https://www.taylorfrancis.com/chapters/edit/10.4324/9781315252148-24/open-source-copyleft-authorship-reconsidered-severine-dusollier

Escamilla, E., Klein, M., Cooper, T., Rampin, V., Weigle, M.C. and Nelson. M.L. 2022 'The rise of GitHub in scholarly publications' in G. Silvello, O. Corcho, P. Manghi, G. Maria Di Nunzio, K. Golub, N. Ferro and A. Poggi (eds) *Linking Theory and Practice of Digital Libraries*, Lecture Notes in Computer Science, Cham: Springer International Publishing. 187-200. https://doi.org/10.1007/978-3-031-16802-4_15

Escamilla, E., Salsabil, L., Klein, M., Wu, J., Weigle, M.C. and Nelson, M.L. 2023 'It's not just GitHub: identifying data and software sources included in publications' in O. Alonso, H. Cousijn, G. Silvello, M. Marrero, C. Teixeira Lopes and S. Marchesin (eds) *Linking Theory and Practice of Digital Libraries*, Lecture Notes in Computer Science, Cham: Springer Nature Switzerland. 195-206. https://doi.org/10.1007/978-3-031-43849-3_17

Faniel, I., Kansa, E.C., Whitcher Kansa, S., Barrera-Gomez, J. and Yakel, E. 2013 'The challenges of digging data: a study of context in archaeological data reuse' in *Proceedings of*

the 13th ACM/IEEE-CS Joint Conference on Digital Libraries, New York: ACM. 295-304. https://doi.org/10.1145/2467696.2467712

Girvan, M. and Newman, M.E.J. 2002 'Community structure in social and biological networks', *Proceedings of the National Academy of Sciences* **99**(12), 7821-26. https://doi.org/10.1073/pnas.122653799

Hacıgüzeller, P., Taylor, J.S. and Perry, S. 2021 'On the emerging supremacy of structured digital data in archaeology: a preliminary assessment of information, knowledge and wisdom left behind', *Open Archaeology* **7**(1), 1709-30. https://doi.org/10.1515/opar-2020-0220

Hippel, E. von and Krogh, G. von 2003 'Open source software and the "private-collective" innovation model: issues for organization science', *Organization Science* **14**(2), 209-23. https://doi.org/10.1287/orsc.14.2.209.14992

Howison, J. and Herbsleb, J.D. 2013 'Incentives and integration in scientific software production' in *Proceedings of the 2013 Conference on Computer Supported Cooperative Work. CSCW 13*, New York, NY, USA: Association for Computing Machinery, 459-70. https://doi.org/10.1145/2441776.2441828

Huggett, J. 2012 'Lost in information? Ways of knowing and modes of representation in e-archaeology', *World Archaeology* **44**(4), 538-52. https://doi.org/10.1080/00438243.2012.736274

Huggett, J. 2018 'Reuse remix recycle: repurposing archaeological digital data', *Advances in Archaeological Practice* **6**(2), 93-104. https://doi.org/10.1017/aap.2018.1

Huggett, J. 2022 'Data legacies, epistemic anxieties, and digital imaginaries in archaeology', *Digital* **2**(2), 267-95. https://doi.org/10.3390/digital2020016

Kansa, E.C. 2012 'Openness and archaeology's information ecosystem', *World Archaeology* **44**(4). https://doi.org/10.1080/00438243.2012.737575

Kansa, E.C. 2022 'On infrastructure, accountability, and governance in digital archaeology' in K. Garstki (ed) *Critical Archaeology in the Digital Age: Proceedings of the 12th IEMA Visiting Scholar's Conference*, Los Angeles: Cotsen Institute of Archaeology Press. 141-52. https://escholarship.org/uc/item/0vh9t9jq#page=156

Kansa, E.C., Whitcher Kansa, S. and Arbuckle, B. 2014 'Publishing and pushing: mixing models for communicating research data in archaeology', *International Journal of Digital Curation* **9**(1), 57-70. https://doi.org/10.2218/ijdc.v9i1.301

Kelty, C.M. 2008 *Two Bits: The Cultural Significance of Free Software*, Duke University Press.

Khan, S., Hirsch, J.S. and Zeltzer-Zubida, O. 2024 'A dataset without a code book: ethnography and open science', *Frontiers in Sociology* **9**, 1308029. https://doi.org/10.3389/fsoc.2024.1308029

Kim, M. 2007 'The Creative Commons and copyright protection in the digital era: uses of Creative Commons Licenses', *Journal of Computer-Mediated Communication* **13**(1), 187-209. https://doi.org/10.1111/j.1083-6101.2007.00392.x

Kintigh, K.W., Altschul, J.H., Kinzig, A.P., Limp, W.F., Michener, W.K., Sabloff, J.A., Hackett, E.J., Kohler, T.A., Ludäscher, B. and Lynch, C.A. 2015 'Cultural dynamics, deep time, and data: planning cyberinfrastructure investments for archaeology', *Advances in Archaeological Practice* **3**(1), 1-15. https://doi.org/10.7183/2326-3768.3.1.1

Kling, R., McKim, G. and King, A. 2003 'A bit more to it: scholarly communication forums as socio-technical interaction network', *Journal of the American Society for Information Science and Technology* **54**(1), 47-67. https://doi.org/10.1002/asi.10154

Lai, J., Lortie, C.J., Muenchen, R.A., Yang, J. and Ma, K. 2019 'Evaluating the popularity of R in ecology', *Ecosphere* **10**(1), e02567. https://doi.org/10.1002/ecs2.2567

Lake, M. 2012 'Open archaeology', *World Archaeology* **44**(4), 471-78. https://doi.org/10.1080/00438243.2012.748521

Leonelli, S. 2023 *Philosophy of Open Science*, 1st edition, Elements in the Philosophy of Science, Cambridge University Press. https://doi.org/10.1017/9781009416368

Limp, W., Kansa, F.E. and Kansa, S. 2011 'Web 2.0 and beyond, or on the Web nobody knows you're an archaeologist', *Archaeology* **2**, 265-80. https://escholarship.org/uc/item/1r6137tb#page=281

Marwick, B., d Alpoim Guedes, J., Barton, C.M., Bates, L.A., Baxter, M., Bevan, A., Bollwerk, E.A., Bocinsky, R.K., Brughmans, T., Carter, A.K. *et al*. 2017 'Open science in archaeology', *SAA Archaeological Record* **17**(4), 8-14. https://eprints.gla.ac.uk/148887/

Milliken, G., Nguyễn, S. and Steeves, V. 2021 'A behavioral approach to understanding the git experience' in *Proceedings of the 54th Hawaii International Conference on System Sciences*, Kauai, HI. 7239-7248. https://hdl.handle.net/10125/71493

Mirowski, P. 2018 'The future(s) of open science', *Social Studies of Science* **48**(2), 171-203. https://doi.org/10.1177/0306312718772086

Morgan, C. 2015 'Punk, DIY, and anarchy in archaeological thought and practice', *AP: Online Journal in Public Archaeology* **5**, 123-46. https://doi.org/10.23914/ap.v5i0.67

Morgan, C. and Eve, S. 2012 'DIY and digital archaeology: what are you doing to participate?', *World Archaeology* **44**(4), 521-37. https://doi.org/10.1080/00438243.2012.741810

Nguyễn, S. and Rampin, V. 2022 'Who writes scholarly code?', *International Journal of Digital Curation* **17**(1). https://doi.org/10.2218/ijdc.v17i1.839

O'Neil, M. 2009 *Cyberchiefs: Autonomy and Authority in Online Tribes*, London, UK: Pluto Press.

Open Knowledge Foundation 2015 *Open Definition 2.1*. https://opendefinition.org/od/2.1/en/

Open Source Initiative 2007 *The Open Source definition*. https://opensource.org/osd/

Opitz, R., Strawhacker, C., Buckland, P., Cothren, J., Dawson, T., Dugmore, A., Hambrecht, G. *et al*. 2021 'A lockpick's guide to dataARC: designing infrastructures and building

communities to enable transdisciplinary research', *Internet Archaeology* **56**. https://doi.org/10.11141/ia.56.15

Pownall, M., Azevedo, F., König, L.M., Slack, H.R., Evans, T.R., Flack, Z., Grinschgl, S. *et al*. 2023 'Teaching open and reproducible scholarship: a critical review of the evidence base for current pedagogical methods and their outcomes', *Royal Society Open Science* **10**(5), 221255. https://doi.org/10.1098/rsos.221255

R Core Team 2023 *R: A language and environment for statistical computing*, Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/

Ratto, M. 2003 'Re-working by the Linux Kernel developers', FLOSShub, Department of Communication, University of California, San Diego. https://flosshub.org/sites/flosshub.org/files/ratto.pdf

Ratto, M. 2007 'A practice-based model of access for science. Linux Kernel development and shared digital resources', *Science & Technology Studies* **20**(1), 73-105. https://doi.org/10.23987/sts.55220

Raymond, E. 1999 'The cathedral and the bazaar', *Knowledge, Technology & Policy* **12**(3), 23-49. https://doi.org/10.1007/s12130-999-1026-0

Roe, J. and Batist, Z. 2024 'Zackbatist/Openarchaeo-collaboration: v1.0', *Zenodo*. https://doi.org/10.5281/zenodo.8393043

Roosevelt, C.H., Cobb, P., Moss, E., Olson, B.R. and Ünlüsoy, S. 2015 'Excavation is ~~destruction~~ digitization: advances in archaeological practice', *Journal of Field Archaeology* **40**(3), 325-46. https://doi.org/10.1179/2042458215Y.0000000004

Saunders, J.L. 2022 'Decentralized infrastructure for (neuro)science' *arXiv*, 2209.07493 (cs) https://doi.org/10.48550/ARXIV.2209.07493

Schmidt, S.C. and Marwick, B. 2020 'Tool-driven revolutions in archaeological science', *Journal of Computer Applications in Archaeology* **3**(1), 18-32. https://doi.org/10.5334/jcaa.29

Scollar, I. 1999 '25 Years of computer applications in archaeology' in L. Dingwall, S. Exon, V. Gaffney, S. Laflin and M. van Leusen (eds) *Archaeology in the Age of the Internet*, Oxford: Archaeopress. 5-10. https://proceedings.caaconference.org/paper/02_scollar_caa_1997/

Sobotkova, A. 2018 'Sociotechnical obstacles to archaeological data reuse', *Advances in Archaeological Practice* **6**(2), 117-24. https://doi.org/10.1017/aap.2017.37

Tennant, J., Agarwal, R., Baždarić, K., Brassard, D., Crick, T., Dunleavy, D.J., Evans, T.R. *et al*. 2020 'A tale of two Opens: intersections between free and open source software and open scholarship', *SocArXiv*, 6 March 2020. https://doi.org/10.31235/osf.io/2kxq8

Tukey, J.W. 1977 *Exploratory Data Analysis*, Reading, MA: Addison-Wesley Publishing Company. http://theta.edu.pl/wp-content/uploads/2012/10/exploratorydataanalysis_tukey.pdf

Whallon, R. 1972 'The computer in archaeology: a critical survey', *Computers and the Humanities* **7**(1), 29-45. https://doi.org/10.1007/BF02403759