Exploring code dissemination in open science: traces of GitHub projects in the literature

Alex Friedrichsen*, James Bagrow, Laurent Hébert-Dufresne

University of Vermont, Burlington, VT
*alex.p.friedrichsen@gmail.com

Figure 1: OCEAN Team Mascot

ABSTRACT

Reproducibility is the foundation of published science by which results are validated or refuted and is a key principle of open science. The relative novelty of the current open science paradigm demands inspection of its reproducibility and citing or attribution practices. We extract over 60,000 links to GitHub repository code artifacts within paper texts from the Semantic Scholar Open Research Corpus. We examine these artifacts, extrapolating that a majority of them involve a repository directly created by an author of the paper they were found in. We describe several qualities of this set of links including the degree distribution of linked papers, the frequency of links found over time, and the bidirectionality of the link from repository to paper. We look at the co-occurrence of citations to papers and their associated repositories through the underlying network structure. Finally, we attempt to elucidate the presence of missing or deleted traces to code artifacts.

1 INTRODUCTION

1.1 Open Science and Reproducibility

Open science and the use of open source software development platforms like GitHub correspond to a model of innovation that combines private and collective action. This new paradigm of social coding and free access to information provides society the best of both organizational worlds [17]. Open science has begun to receive a spotlight in research, as researchers question the true impact of free information and the ins and outs of current platforms and practices surrounding it. The tenets of open science demand open access to data, open source publications, and strive for increased rigor, accountability, and reproducibility for research. These goals are met through values that embody inclusion, fairness, equity, and sharing. Open science brings change to the way research is done, who is involved, and the value of research [19].

The reproducibility of research is particularly crucial. According to a survey done through the journal Nature, 52% of 1,576 surveyed researchers believe there is a crisis in reproducibility. Moreover, more than 70% had "tried and failed to reproduce" other scientists' work. The data suggests only 10-40% of publications have reproducible results [8].

While there are many elements that may contribute to the difficulty of reproducing a given scientific result, two particularly important factors include the unavailability of code, and the unavailability of data. Access to researchers' raw data and code may assuage the crisis [26]. Together, these factors contribute towards the irreproducibility of work in 40-50% of surveyed work [26]. They are exacerbated by a lack of standardization in code sharing practices [16]. Several journals and conferences have recently mandated data and/or code availability statements within publications and shared work. These statements are not standardized. Further research seeks to answer what percentage of journals currently subscribe to these code availability policies [13, 24]. It is clear that bettering our understanding of how papers are currently citing code, a growing practice alongside the growth of open science, is a strong first step in addressing the problem and plays towards formulating better practices than are currently in place. In addition, the decay rate of links or "link rot" may make some existing links useless over time. Measuring the quantity and clarity of links between artifacts is where we direct our focus towards understanding and improving the traceability of scientific artifacts.

Past research provides an additional incentive for improving the provision of accurate citations to code and open access to code; publications with open access to code receive higher citation counts on average. Publicly available code is found to increase citation counts on papers in the machine learning field [36]. Publicly available articles in the biostatistics field with open access to code were found to have a 2-fold increase in number of citations [35]. Though, some studies disagree: data sharing policies implemented by journals did not appear to lead to higher citation counts in the following five years after enacting the policies [11].

1.2 Related Literature

The study of traceability of scientific artifacts is an emerging field that spans this question and the other research questions we pose later in this report. Few studies have been published within this field. We cover in some detail the findings of previous work here. Firstly it is important to familiarize oneself with the common research tools and data sources for papers dealing with the traceability of scientific artifacts.

A difficult piece in creating new research on the traceability of scientific artifacts is choosing or building a dataset to work from. Oftentimes, a dataset can be the key to unlocking novel findings within a field. Most scientific projects involving code choose to employ a piece of software for keeping track of their code and facilitate coding collectively. Several domains exist to fulfill this internet hosting and version control niche, including Bitbucket, Sourceforge, GitLab, and most prevalently GitHub among alternatives. In two studies, to obtain a sample of 20000 GitHub projects, the authors used regular expressions to match paper-link-appearing text objects [16, 38]. Then, a representative sample was taken from within this 20000 and links were tested for validity. This data set is available for future use [37].

A GitHub repository recommender named *paper2repo* uses machine learning to recommend users with repositories that are like a chosen paper on an academic search system such as Microsoft Academic [30]. However, this does not necessarily give the repositories used by the authors of the paper itself, which is problematic for the type of analysis we want to carry out. Similarly, a browser extension available for Chrome and Firefox, *CatalyzeX*, attempts to link to code for machine learning papers when browsing in Google Scholar, Twitter, and other popular sources [10]. Another method to obtain a sample was to take all links from papers to repositories from Microsoft Academic Graph classified as primary and belonging to computer science, resulting in collection of around 5000 at time of publication links, of which about three-fifths were downloadable [15]. An effort aimed at collecting machine learning projects and their related papers has been cultivated on the site *paperswithcode.com* [5]. This site is another potential source for accurate links between repositories and papers.

Our research question falls squarely in the emerging zone of traceability of scientific artifacts. Much of the prior research here also looks at links inside of GitHub. Prana *et al.* [28] categorize the contents of README files systematically by first manually labeling and then automatically classifying different sections of README files. One of the sections labeled "References" includes external links out of the repository which is of interest when trying to identify links to papers from repositories. The aptly titled paper "science-software linkage" by Hata *et al.* [16] is one of the first to explore links between papers and repositories. The software domain was categorized, evolution of links analyzed (link maintenance), and potential bidirectional links categorized. This bidirectionality was categorized by including links to papers that do and do not link back to their own repository, broken links (404 errors), and papers that do not link back to any repository. Links from academic papers link to GitHub most of the time, among all types of links [38].

Numerous other relevant questions were investigated in Wattanakrienkrai *et al.* The number of repositories that link to papers has been estimated to be 91% in a sample of 20,000 README files [38]. The authors of linked papers and repositories have been examined by affiliation and found to come mostly from academia [38], [25]. In addition, the authors have been cross compared with repository contributors to see if the people publishing the paper are the same making edits to the code and to see if the lead authors are the lead contributors to the repository, finding that 40% of repositories are being edited by the authors of the linked paper, and greater than 50% are other contributors implementing the research published in the paper. The most referenced arXiv papers from GitHub repositories are identified. It is discovered the most cited papers are also the most referenced in GitHub. Finally, the evolution of links on GitHub is also analyzed finding that changes in links are rare. Additionally, when a paper is updated a corresponding update to the link in a linked repository is rare. Several questions about the distribution of basic properties of linked repositories were examined by Färber [15]. The distribution of repositories by number of stars, forks, contributors, lengths of repository manuals, programming languages used, machine learning frameworks used, and fields of study were each visualized. Moreover, affiliation by institution, conference series, and journal was also displayed.

A study by McGuinness and Sheppard [24] look at data availability statements in a set of medRxiv preprints, finding that while data availability statements do increase open access of code, they are not the end all be all. They find that lack of enforcement and stringency in data availability policies often resulted in data not being accessible, either before or at the time of publishing. Moreover, Federer *et al.* [13] inspected a data sharing policy in PLOS ONE requiring researchers to share the data underlying their results. Their analysis included just under 50,000 papers between March 2014 and May 2016. They found that compliance with the policy had increased, as well as a decline in papers that did not include a data availability statement, however only about 20% had made the data available in a repository, the preferred method as stated by the policy. The research suggests that while data availability statements and policies are helping improve access, other knobs need tuning to increase data availability further.

2 METHODS

2.1 Motivation

In this section we introduce our research questions, data collection and data engineering, and state steps to reproducing this results of this paper. We set out to fill a gap in the small but growing traceability of scientific artifact literature to improve understanding

of the growth of open science. We laid out a handful of research questions to explore that corresponded to this line of inquiry.

Research question one: is there some kind of structure in the network of traces between papers and code? The type of structure we expected to find would be most comprehensible through a bipartite network visualization. At the full scale of the data, the graph may be too dense to make informed observations. Do smaller components of the full network exhibit patterns? Particularly, what patterns may be associated with the degree of a node in the network? The full network could also be filtered by any number of qualifiers prior to visualization. This leads us to ask which attributes of the network most clearly lend structure to the graph visualization?

Research question two: What features help us understand the distribution of links observed between papers and repositories in our dataset? While broad, the analysis of our dataset would not be complete without a thorough inspection of the attributes of papers and repositories in our sample. We questioned: is repository authorship and distribution of contributors correlated with paper authorship? Moreover, how do the distributions of fields, journal, stargazers, and citation counts relate to each other in our set and compared to a null sample of papers, if at all? Our final question seeks to mine the temporal dimension of our data. Has the frequency of links changed over time relative to the growth of published projects. We were particularly interested in the possibility of changes in links, missing links, or deleted links. Do links stay connected from papers to their code artifacts?

2.2 Data Engineering

GitHub is the predominant collaborative coding site with over 100 million repositories as of November 2018 [7]. Working with GitHub warrants us several clear benefits. Firstly, many GitHub repositories have README markdown files where most commonly the "what" and "how" of a repository is outlined, along with being the most common place in a repository for outbound links and citations [38]. Secondly, GitHub has a free rate-limited RESTful API that facilitates repository level data collection for public repositories. These API requests are stored in json format and offer many useful repository attributes such as contributors, issue information, commit information, and repository creation data. GitHub is also the primary data source for researchers seeking to understand how users collaborate on software projects [20]. Lastly, several features on GitHub serve to promote teamwork and discussion within a project and allow users to track and clone (fork) others' work. GitHub is an obvious and compelling choice for a data source in any study pertaining to open source science and software development.

Though README files and a token-gated API facilitate data collection, they each have their own limitations. The structure of README files is entirely unstandardized thus increasing the variability and difficulty in verification of a repository's bidirectional status, where bidirectionality assumes the repository has a link back to the academic paper that cited it. The API does not provide full page html or text for any of the repository's pages, nor does it allow the querier to go into the code contained within files in the repository. This limits our ability to search for links anywhere other than within the README.md files due to the manual checking requirements imposed. Other shortcomings may be avoided through the wisdom of prior work done with GitHub data. Kalliamvakou *et al.* [**?**] help researchers identify some of these pitfalls, such as knowing not all repositories are full projects or exist on GitHub in combination with some other platform.

To pair with our GitHub repository data collection we choose to look at a set of academic papers from the Semantic Scholar Open Research Corpus, S2ORC. We filtered by papers with links specifically to GitHub. The Semantic Scholar Open Research Corpus is an open, general-purpose corpus intended for non-commercial use. Paper metadata, abstracts, and citation edges associated with conventional citation graphs are grafted with full text pdf parses of papers that maintain consistency in paper structure. Papers loaded into S2ORC are all open-access. The corpus covers more than 136 million paper nodes and 467 million citation edges [22].

The Semantic Scholar corpus was downloaded and stored on a database due to its large (1000 gigabyte) size. We began the task of filtering through sections of the database to pull out ids of papers that contained a link to GitHub. Each paper has a unique id in the corpus that is referenced in citations between papers and is the key to an accessible web page through *semanticscholar.org* [22] where an open-source link to a PDF copy of the paper can be found. A regular expression was chosen to match any *github.com* appearing link which allowed us to capture links in the most common format "https://github.com/user/repo" as well as a number of other cases in our test set, which can be found here (*regexr.com/6i842* [6]). Notably, due to the difficulty of parsing PDFs, our matching ran into inconsistencies while running on the corpus due to additional spaces inserted into links where unexpected. This issue was mitigated through matching links that allowed for spaces in the "https://github.com" portion of the link, as well as any portion preceding a forward slash in the next word. Finally, the word following the word containing the final forward slash was matched in order to attempt to mitigate spaces in the final section of the link. This resulted in duplicate matches for every link found.

Following the initial matching, links were transformed into the format of GitHub API calls: "https://API.github.com" and requests were performed for each potential link match. The results of the requests were saved in json format and the status codes of the requests were compared for each of the two potential links per match. The link returning a 200 status code was labeled as the "true" link. If both links returned 200 codes, the longer of the two links was assumed to be the correct match given the probability that a space could've been inserted between the user identifier and repository identifier inside the URL. If neither link returned a positive status code, a note was made that there was no positive match - in this case the link could have more spaces in the last word (impossible to match without exponential computation costs and inordinate rate-limited timeliness) or the link could be a dead or deleted link.

This matching left us with a set of 80,720 matched links of which 60,687 were unique links and 67,358 were unique papers. While we were able to obtain repository level information about many of these links through the API, confirming that a paper and a link had a direct relationship for a particular paper (the repository came directly out of research done by the authors for the paper or vice versa) proved a difficult issue. Our solution was to manually check a sample of our link to see if they were a true match in order to

extrapolate what the true proportion of link matches would be. In addition during our manual checking we would ensure the link was not a duplicate, the link was parsed correctly compared to within the paper, and whether the repository's README.md file contained a link or citation back to the original paper (bidirectionality).

This manual checking was done on a sample of 171 links and associated papers using the following process guidelines. The rater was first instructed to check if the link was the same as one of the paper's other links in the cells above. Then, they were to open the API link and mark the box if the API was accessed correctly, and whether they could access the repository from the API (the repository was not private). Then, the rater would open the semantic scholar link through the paper's id, open the paper's pdf, and control-find for all github links in the paper. Once they reached the link matching the link repository link, they would use context clues in the paper to determine if the two were related. They would expand upon this by double checking within the repository for matching contributor-author names, title, subject, and finally scanning the README.md for a link back to the paper.

3 RESULTS

3.1 Sample Results

Our manual checking found that 56% (95/171) of links were matches with their papers (the repository came directly out of research done by the authors for the paper or vice versa). Of these matches, 47% (45/95) were bidirectional matches where the repository also included a link back to the paper, corresponding to 26% (45/171) of the entire manual sample.

Sample Percentage of Total Matches and Bidirectional

	Total	*Match*	*Bidirectional*
Total	100	56	26
Match	56	100	47
Bidirectional	26	47	100

To answer research question two we created several figures from the data in a slightly larger sample than our manual sample, that included 511 links within a set of 377 unique papers. In figure 2 we see the distribution of the number of authors per paper. The mean number of authors is 6.48 authors. The distribution is right skewed with most papers having less than 30 authors. A few outliers have in excess of 50 authors.

The number of contributors in sample repositories are also distributed excluding outliers below about 30 contributors with a strong right skew as can be seen in figure 3. Repositories with greater than 30 contributors are binned together in the top bin. Nevertheless, there is strong right skew with several repositories having a very high number of contributors. Most repositories have one contributor while about half of that have 2 contributors and half of that still 3, decreasing less than exponentially from there and on. The mean number of contributors is 4.33 contributors. The correlation between numbers of contributors and numbers of authors is low, only 0.045 even when filtering out the papers with abnormally high author counts.

The fields of papers were determined based on the Microsoft Academic primary field of study. If a paper was listed under multiple fields, it was counted towards both of those fields. The top

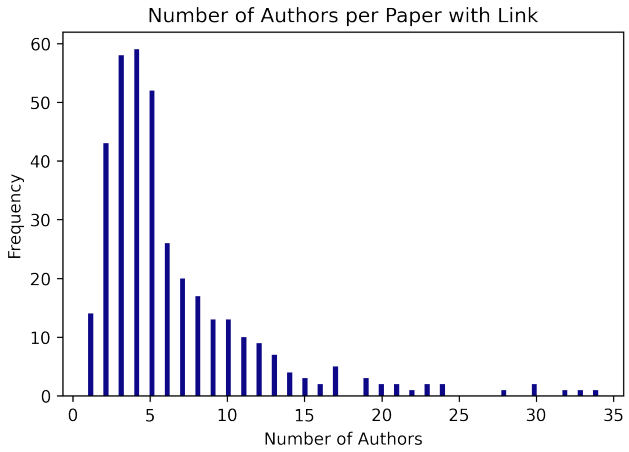

Figure 2: Distribution of Number of Authors per Paper with a Link

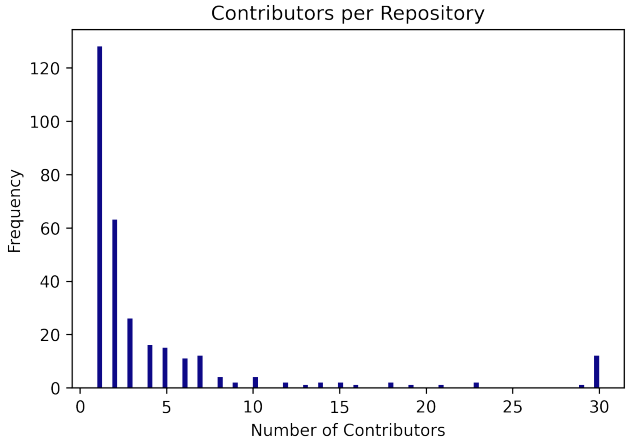

Figure 3: Distribution of Number of Contributors per Repository

fields of papers with links to repositories are overwhelming biology, medicine, and computer science. In a random sample of papers taken from S2orc, the most popular fields were respectively medicine, computer science, chemistry, engineering, and biology per figure 4. These fields were significantly different.

It can be seen in figure 5 the top journals of papers with links were arXiv and bioRxiv with 15.6% and 14.4% of the papers in our dataset, respectively. Compared to a random sample of papers from S2ORC, both were relatively less popular compared with other journals in papers with links 6.

The number of inbound citations and number of outbound citations were also compared for our sample against a random sample of S2ORC papers. Inbound citations refer to papers citing the original paper. Outbound citations refer to sources that the original

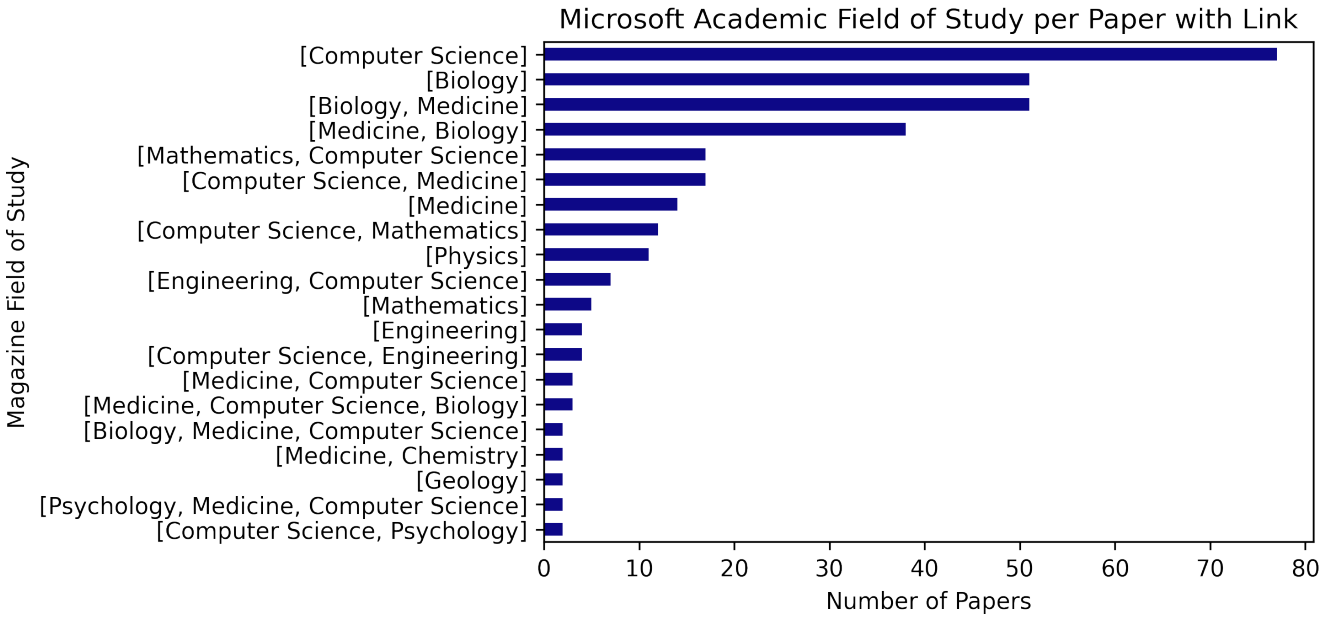

Figure 4: Most Frequent Magazine Primary Field(s) of Study for Papers with Links

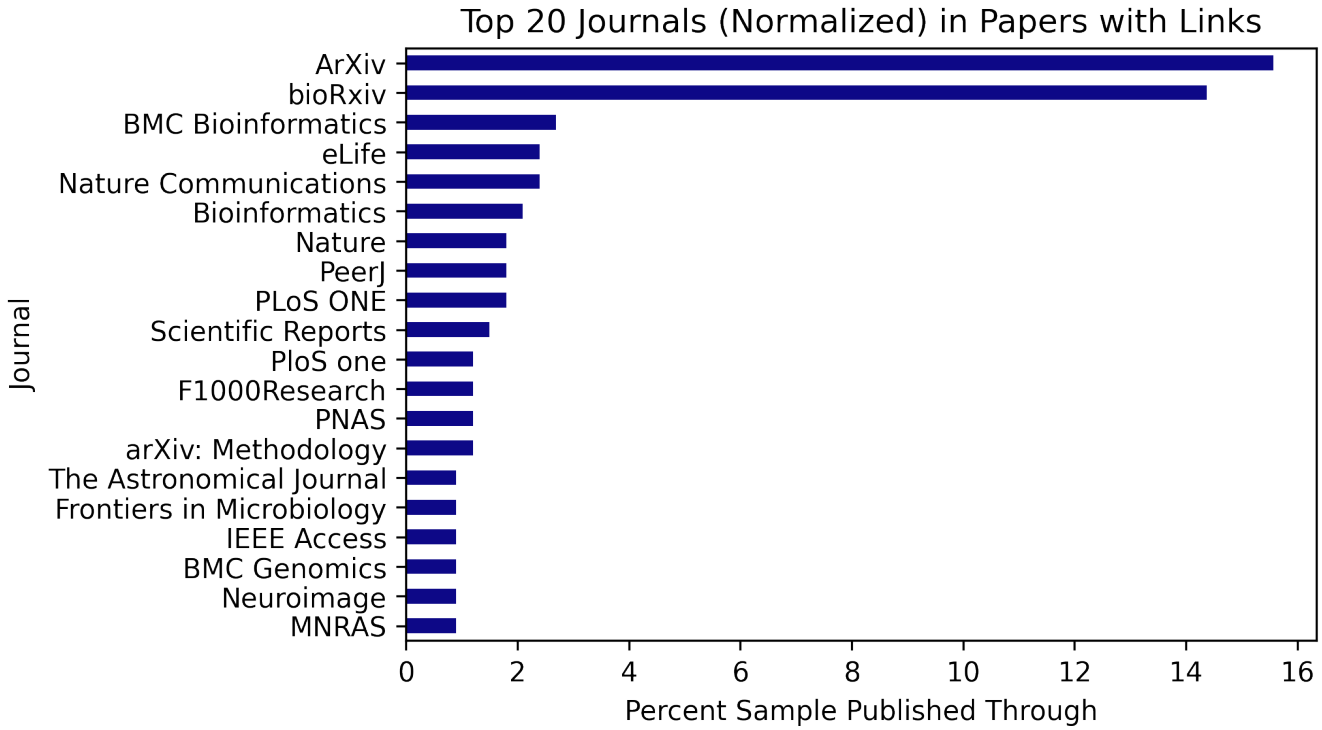

Figure 5: Top Journals in Papers with Links

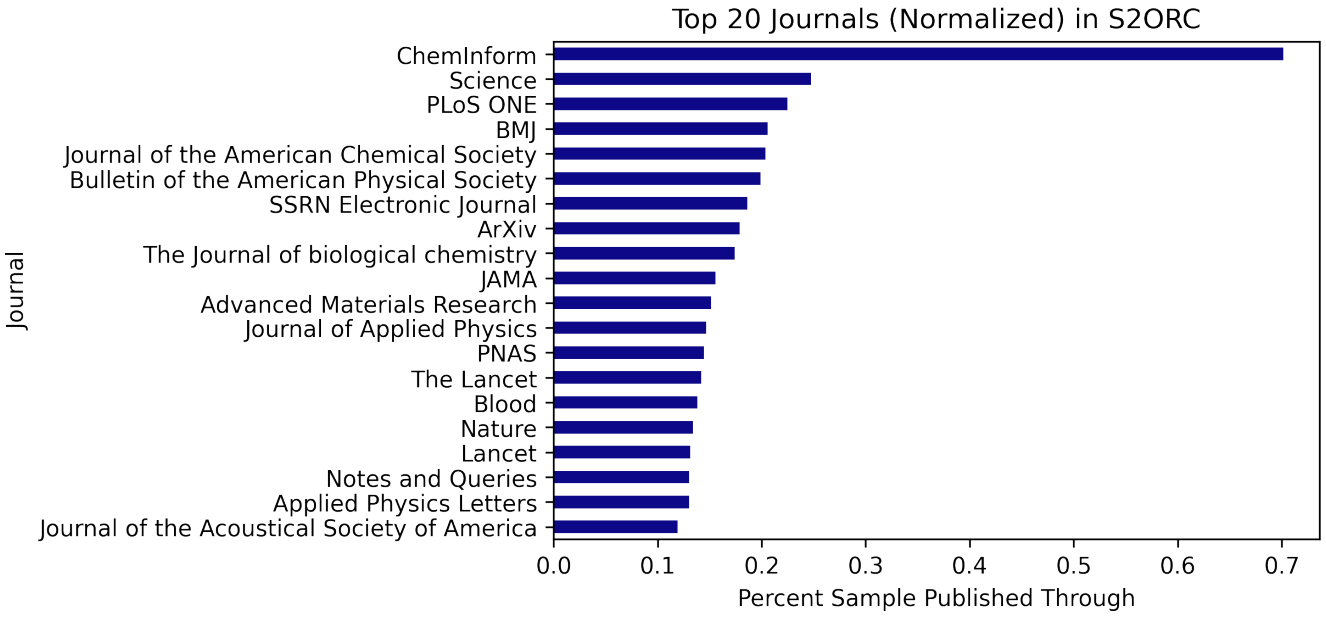

Figure 6: Top Journals in S2ORC

paper is citing. A log scale is used to transform the results of inbound citation graphs. The inbound distribution's median is zero citations and there is a sharp drop in the distribution for papers with greater than 0 citations 7. This is similar to the random sample albeit the random sample's center on 0 is more pronounced 8.

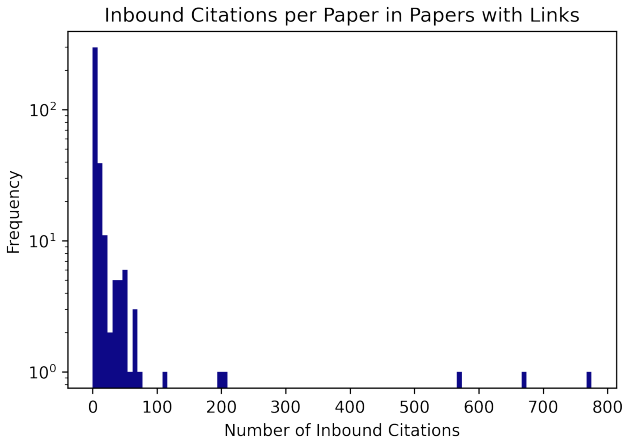

Figure 7: Number of Inbound Citations in Papers with Links

The distribution of outbound citations is centered around 20 with a gradual right skew 9. The distribution of outbound citations for the random sample of papers is not shown, but is heavily concentrated in the first bin at zero citations.

Some temporal data was collected about the sample. The date created for each repository is graphed in 10 up to 2019. About 18%

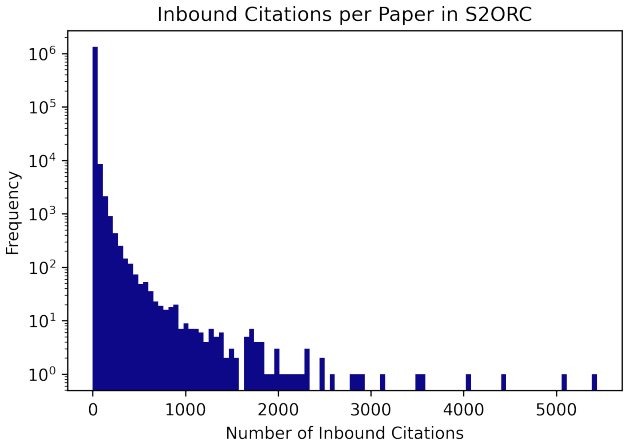

Figure 8: Number of Inbound Citations in S2ORC

of repositories were created in 2016, then again in 2017, and in 2018, with a slight decrease to about 12% in 2019 12. The year published for each paper in the sample is also graphed and a similar graph is made for comparison to a random sample of S2ORC papers. Papers from 2020 and forward are excluded due to incomplete yearly data. There is a steady increase of papers published each year with a super-linear growth rate between years 10. In the control sample, about an equal number of papers are published yearly **??**. Another graph is attached in the appendix **??** from [4] detailing the annual growth in the number of GitHub repositories.

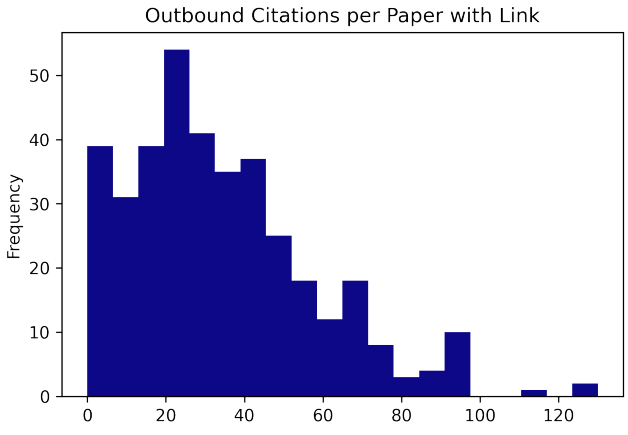

Figure 9: Number of Outbound Citations in Papers with Links

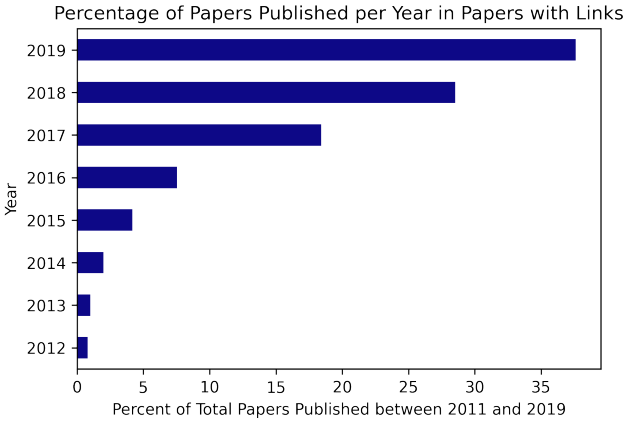

Figure 10: Distribution of Years Published for Papers with Links

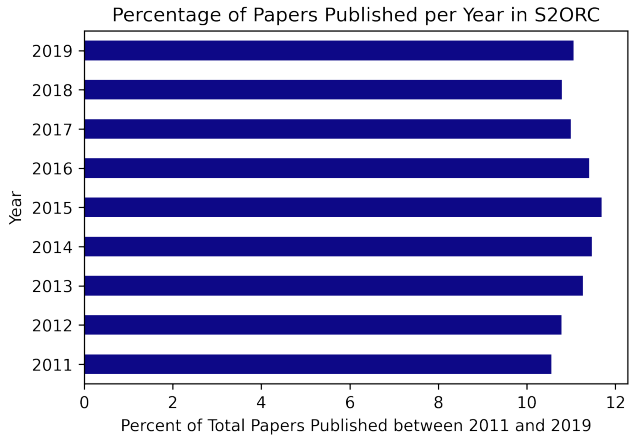

Figure 11: Distribution of Years Published for Papers in S2ORC

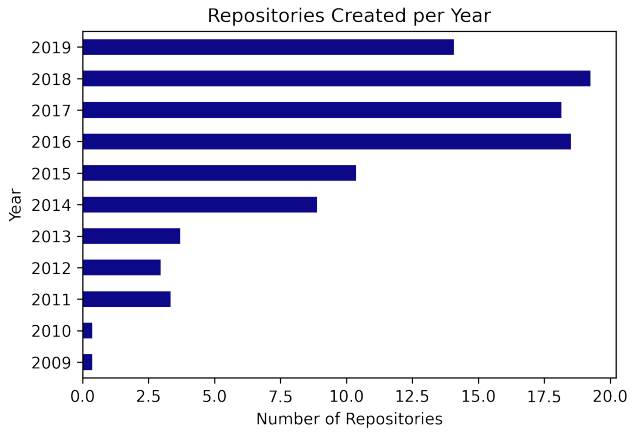

Figure 12: Distribution of Years Linked Repositories were Created

The Pearson correlation coefficients were calculated for each of the above measures and were summarized in a correlation matrix in figure 13.

3.2 Full Sample and Bipartite Network Results

The network of papers and links has an interesting structure. The full network with 60,687 link nodes and 67,358 paper nodes is drawn with a bipartite coloration in 14, excluding degree one nodes. Edges are links to repositories embedded in papers. The Yifan-Hu force directed layout algorithm is applied to minimize overlapping of network components and reveal existing symmetries in the network. The average degree of the full network is 1.261. A glut of the full network consists of degree 1 pairs of nodes with no other connections; one paper citing one repository without other interaction. A second view of the network filtering for nodes with degree greater than one is provided to mitigate this graphical density in 14. Table 1 displays the degree and repository links of the top 6 highest degree nodes within the full network.

The largest connected component of a network is the piece of the network with the largest proportion of nodes that are reachable from every other node in the component. We visualize the largest connected component (LCC) and calculate the degree distribution 15. The average degree of a node in the GCC is 2.173 16. The average path length is 9.13. The betweenness centrality distribution is not shown but is heavily right skewed. Several smaller “chains”, one of many smaller connected components, and a zoomed in view of a highly cited node "bloom" are also highlighted in figures 17, 18, and 19 as further examples of structure within the network.

We also examine how often papers with links are cited by papers that also cite the associated code. This co-occurrence of citation forms what can be understood as triangles in the citation graph. The edges in this triangle represent an inbound (conversely an

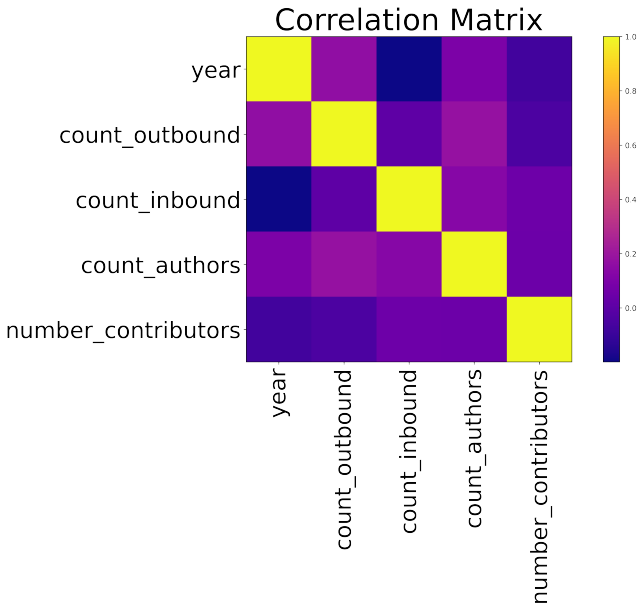

Figure 13: Correlations between Attributes of Papers with Links

outbound) citation between papers, and two edges from both papers to the cited repository. In our sample of 377 unique papers we extrapolated the papers citing these papers through their inbound citations. We then checked whether any of these papers had links to the original set of 511 cited repositories in our sample. Forty three triangles were located in this set. These triangles are composed of ninety-one unique papers and repositories. The total possible number of triangles is calculated to be the number of excecss paper nodes beyond repository-paper pairings in the set, excluding triangles comprised of two papers in which one paper cites two repositories. This is estimated to be 5507 triangles possible. Thus we see at maximum a ratio of 43 to 5507 triangles or .7% the total possible triangles. The number of connected components, which is 331. The average degree of this co-occurrence network is 1.966.

Finally, we discern the rate of decay of links in our set over time. Twenty nine percent of links contain an invalid or broken link. There was no way to differentiate links that were broken from an inexact parse, or from a 404 error, deleted repository, or changed link. In the future, looking into a more detailed synopsis of why links were missing or broken might prove fruitful. We performed a logistic regression by year paper published and availability of link. The results did not suggest that there was a correlation between a link's validity and the year published.

4 DISCUSSION

In general, matches were relatively common at about half of all GitHub links found within papers. Bidirectional links were about half this rate still, at about 1 in four repositories. This result is consistent with findings from Wattanakriengkrai *et al.* [38].

The number of paper authors is consistently higher than the number of repository contributors and there is no correlation between these two numbers. This indicates that not all authors are contributing to the codebase for a project, and in fact oftentimes only one author will contribute to the repository. One limitation of this finding is the possibility that scientists share one GitHub account between them. Comparatively to the control sample of S2ORC papers, the papers containing links less frequently come from engineering and chemistry fields. In addition, biology is an overrepresented field in papers with links. The popularity of bioRxiv likely reflects the frequency of biology and medicine papers, however there is some possibility that more popular venues like bioRxiv and arXiv have stricter code and data availability policies. Further analysis should investigate whether these venues and other noted top journals are more frequently requiring code and data availability statements. Our findings indicate that several large connected components exist that center on bioinformatics papers with all 6 highest degree nodes being bioinformatics repositories. These 6 repositories formed the core nodes of each of their respective components from which stemmed several offshoot "branches". What each of the six have in common is their intended purpose: they are all source code for tools commonly used in bioinformatics research, as opposed to links to data sources, visualizations, analysis, data sets, or more broadly categorizable GitHub repositories. They are all also heavily starred repositories, indicating their popularity in the broader GitHub community, potentially in an adjacent or separate domain from the scientific community.

The number of inbound citations for papers that link to repositories is notably higher than the random sample of S2ORC papers. This result is consistent with previous findings that tied code availability to higher citation counts.

The years papers are published seems to echo the trend of the year repositories are created. This suggests that most scientists are either beginning their own repositories about a year in advance of publishing their findings, or are basing their new publications off of new open source tools and then cited in the publication about a year later, on average. The prevalence of papers with links to code is increasing relative to the number of papers published, which seems to be remaining about the same. This result matches our intuition given the growth of the open science paradigm. Notably, the prevalence of papers linking to GitHub is increasing relative to the growth of GitHub on the whole, which gives more weight to this result.

The lack of correlations between some of our metrics is surprising. One might expect that the year of publication and number of inbound citations are correlated, as the paper becomes more cited over time. Nevertheless, our results suggest that there is no such correlation, leading us to believe that perhaps in papers linking to code, other factors outweigh the effect of time on inbound citations. We postulate these effects could be the expedited spread of code through GitHub and therefore the linked paper itself. Additionally, the correlation between the number of authors and the number of inbound citations is remarkable as a result specific to open science papers. Similarly, the lack of correlation between outbound citations and inbound citations may be contradictory to evidence in other fields of papers, such as ecology [21].

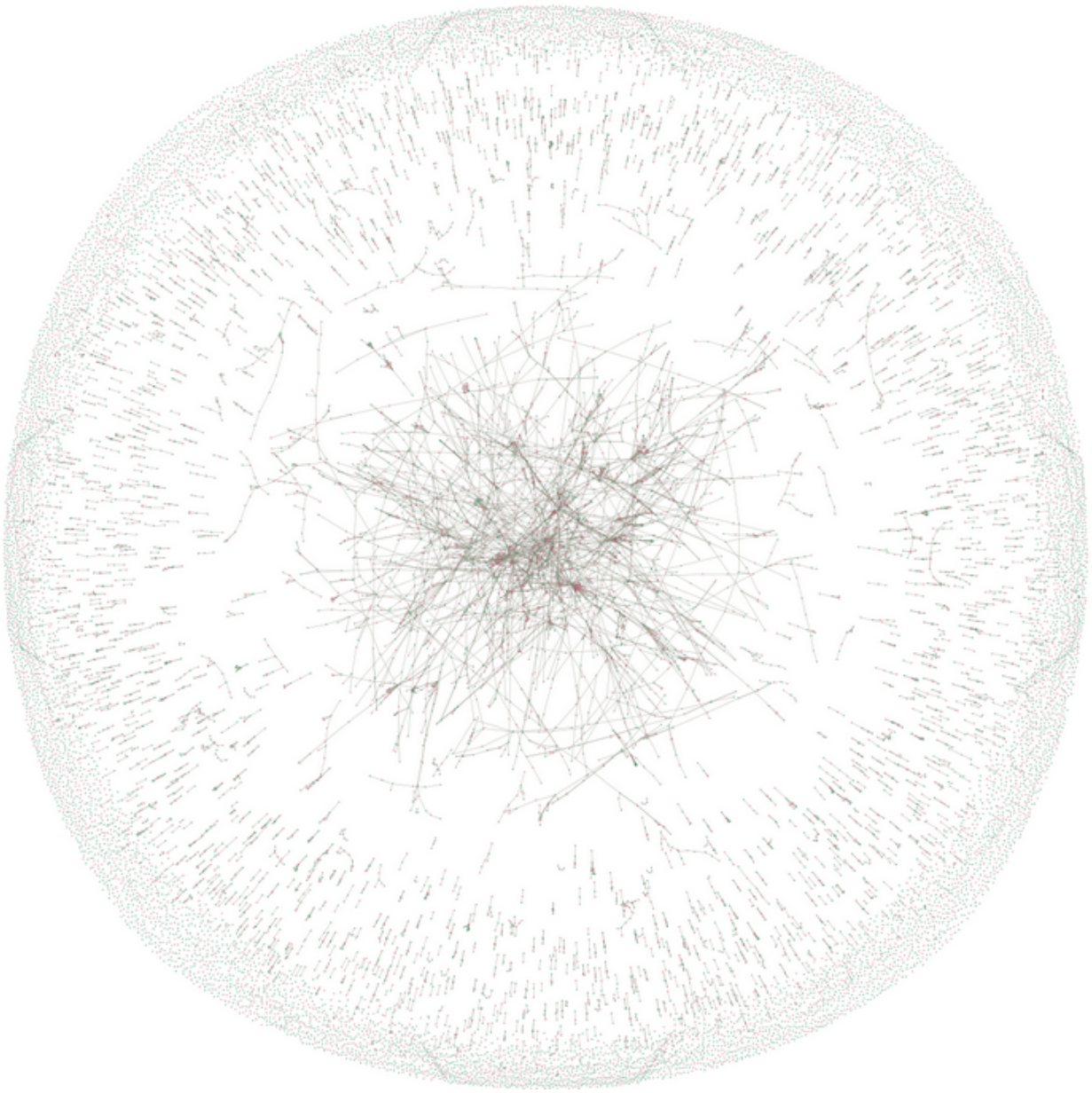

Figure 14: Bipartite Network between Papers and Repositories for Nodes Degree > 2. Green nodes are repositories, red nodes are papers.

The structure of the full bipartite network reveals several interesting insights into paper-link tracing. Firstly, it is very common for a paper-link pair to be disconnected from the rest of the network. However, it may be worthwhile to keep an eye on whether this is becoming less frequent as the frequency of links within papers increases along with the available code and associated literature on GitHub and other similar platforms. The full network naturally breaks down into larger connected components by field of study, and these connected components also exhibit unique structure. Most of the larger connected components center around one link. Such high-centrality link nodes have a tendency to be coding toolkits used by many scientists to perform their research as one step in their process. Nevertheless, connected components also often exhibit long chain-like structure, where a repository is cited by a paper that also cites another repository, that is then cited by a different repository, and so on. Investigating the nature of these

Figure 15: Largest Connected Component of the Bipartite Network. Green nodes are repositories, red nodes are papers.

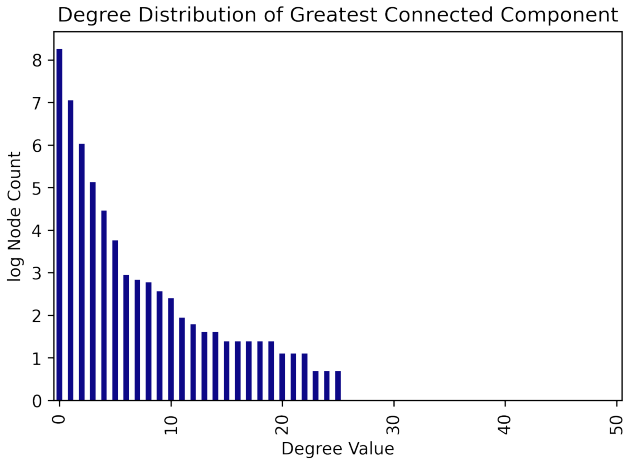

Figure 16: Largest Connected Component Degree Distribution

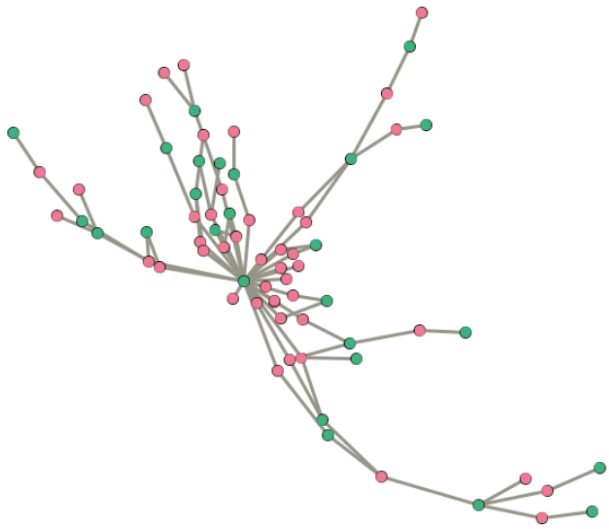

Figure 17: A smaller connected component from the full bipartite network. Green nodes are repositories, red nodes are papers.

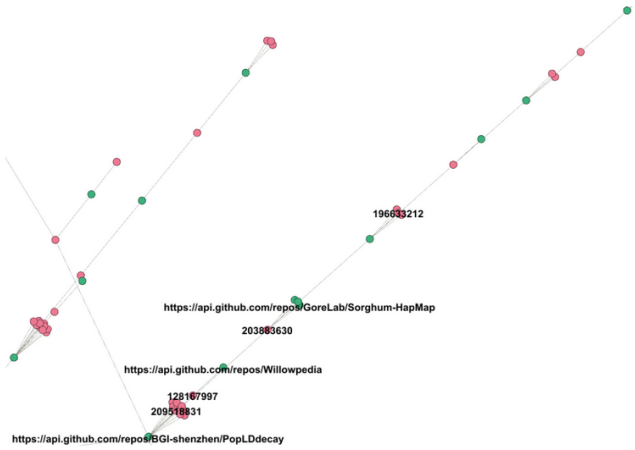

Figure 18: A zoomed in view of one of the "chains" of the GCC. Labeled are the nodes in the beginning of the chain by repository link and S2ORC paper ID. Green nodes are repositories, red nodes are papers.

chains, we find that they are highly specialized subsections of the literature. For instance, in one such chain an author built upon their own work using a tool that they built to aid in genome sequencing by using the tool in another paper, which also cited the repository of a tool for another step in the paper's methodology. Then, the same author referenced the second tool in yet another paper. Finally, the co-occurrence of papers citing another paper and that paper's repository seems few and far in between, in our sample. The low prevalence of co-occcurrence at less than a percent speaks to the infrequency with which papers are citing another paper's GitHub repository. It occurs much more frequently that a paper will either cite the other paper or the repository only. Paper citations tend to outrank in degree link citations, with highly cited papers being more popular than highly cited repositories in our sample.

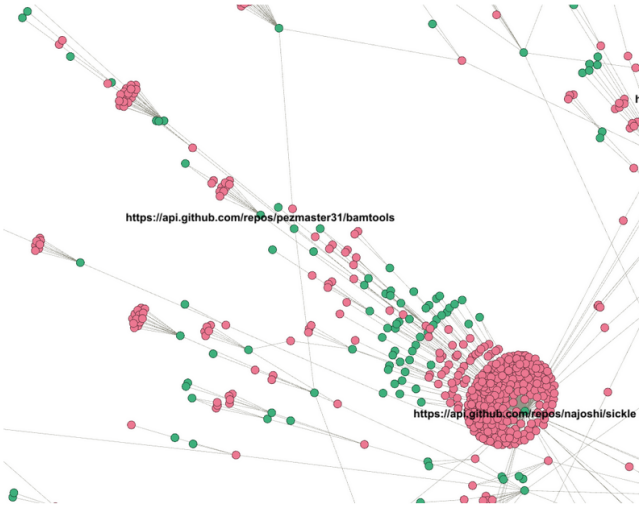

Figure 19: A zoomed in view of one of the "blooms" of the GCC. Labeled are the highest degree nodes. Green nodes are repositories, red nodes are papers.

5 LIMITATIONS AND FUTURE WORK

There were several limitations to our analysis and important caveats we wold like to cover in this section. Regarding our data source, S2ORC may influence the rate of a paper having a link due to all papers being open source, which is concerning to the generalizability of our results to all online papers. In the future, also increasing the generalizability, we would like to match links to alternative version control or software hosting sites. More specific to our data, we would like to match links that contain more than one space in the last word after a word with a forward slash.

The decay rate of paper links as mentioned is a general trend instead of exact data, as our set of non-working links is comprised of broken links and additionally links that were parsed incorrectly. The frequency of an incorrect parse is from our sample estimated to be about one in five (80%) with 31 out of 159 papers being parsed correctly. We would like to expand the manual checking of papers and enlarge the sample. In future analyses we hope to employ an inter-rater reliability score to add more statistical certainty to the results of our manual checking. The sample could also in future work be expanded by increasing the speed of metadata retrieval.

We also hope to check the collected repositories for links to other papers in addition to bidirectional links. A similar type of automation could be employed such as the regular expressions used in this study. We are left with numerous other questions we hope to answer in future studies surrounding the traceability of scientific artifacts. Our analysis on number of authors leads us to the question of whether lead author(s) contribute to the repository more often or more, proportional to the number of issues or commits in the repository? Or is this work delegated to another member on the team more often than not? GitHub repositories are frequently cited in papers. GitHub and other software services are being used to answer the questions of myriad scientific researchers. However, is the manner in which they are being used different from the manner in which independent software developers, enterprises, or other entities use them?

Finally, we question whether the evolution of code or links to code in papers with code artifacts that receive awards or increased recognition. We hope that some of these question can be answered in future studies or analyses in the coming years.

ACKNOWLEDGMENTS

The authors would like to thank Edward Gilbert, Dylan Gooley, and Nick Knudsen for assistance in manually checking links. We would also like to acknowledge the support of the University of Vermont Honors College.

REFERENCES

[1] [n.d.]. *Add It Up: Takeaways from GitHub's Octoverse Report.* https://thenewstack.io/add-it-up-takeaways-from-githubs-octoverse-report/
[2] [n.d.]. *GitHub Repositories with Links to Academic Papers: Open Access, Traceability, and Evolution.* https://github.com/NAIST-SE/GH2Papers original-date: 2020-08-24T04:37:34Z.
[3] [n.d.]. *Google Scholar.* https://scholar.google.com/?inst=11539667347833751441
[4] [n.d.]. *mirror/rise-of-github.ipynb at master · bugout-dev/mirror.* https://github.com/bugout-dev/mirror/blob/master/notebooks/rise-of-github.ipynb
[5] [n.d.]. *Papers with Code - The latest in Machine Learning.* https://paperswithcode.com/
[6] [n.d.]. *RegExr: Learn, Build, & Test RegEx.* https://regexr.com/
[7] [n.d.]. *Thank you for 100 million repositories.* https://github.blog/2018-11-08-100m-repos/
[8] Monya Baker. [n.d.]. 1,500 scientists lift the lid on reproducibility. 533, 7604 ([n. d.]), 452–454. https://doi.org/10.1038/533452a Number: 7604 Publisher: Nature Publishing Group.
[9] Xuyang Cai, Jiangang Zhu, Beijun Shen, and Yuting Chen. [n.d.]. GRETA: Graph-Based Tag Assignment for GitHub Repositories. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)* (2016-06), Vol. 1. 63–72. https://doi.org/10.1109/COMPSAC.2016.124 ISSN: 0730-3157.
[10] CatalyzeX. [n.d.]. *CatalyzeX: machine intelligence to catalyze your projects.* https://www.catalyzex.com/
[11] Garret Christensen, Allan Dafoe, Edward Miguel, Don A. Moore, and Andrew K. Rose. [n.d.]. A study of the impact of data sharing on article citations using journal policies as a natural experiment. 14, 12 ([n. d.]), e0225883. https://doi.org/10.1371/journal.pone.0225883 Publisher: Public Library of Science.
[12] Giovanni Colavizza, Iain Hrynaszkiewicz, Isla Staden, Kirstie Whitaker, and Barbara McGillivray. [n.d.]. The citation advantage of linking publications to research data. 15, 4 ([n. d.]), e0230416. https://doi.org/10.1371/journal.pone.0230416 Publisher: Public Library of Science.
[13] Lisa M. Federer, Christopher W. Belter, Douglas J. Joubert, Alicia Livinski, Ya-Ling Lu, Lissa N. Snyders, and Holly Thompson. [n.d.]. Data sharing in PLOS ONE: An analysis of Data Availability Statements. 13, 5 ([n. d.]), e0194768. https://doi.org/10.1371/journal.pone.0194768 Publisher: Public Library of Science.
[14] Eitan Frachtenberg. [n.d.]. Research artifacts and citations in computer systems papers. 8 ([n. d.]), e887. https://doi.org/10.7717/peerj-cs.887 Publisher: PeerJ Inc.
[15] Michael Färber. [n.d.]. Analyzing the GitHub Repositories of Research Papers. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020* (New York, NY, USA, 2020-08-01) *(JCDL '20)*. Association for Computing Machinery, 491–492. https://doi.org/10.1145/3383583.3398578
[16] Hideaki Hata, Jin L. C. Guo, Raula Gaikovina Kula, and Christoph Treude. [n.d.]. Science-Software Linkage: The Challenges of Traceability between Scientific Knowledge and Software Artifacts. ([n. d.]). arXiv:2104.05891 http://arxiv.org/abs/2104.05891
[17] Eric von Hippel and Georg von Krogh. [n.d.]. Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science. 14, 2 ([n. d.]), 209–223. https://doi.org/10.1287/orsc.14.2.209.14992 Publisher: INFORMS.
[18] Sven E. Hug and Martin P. Brändle. [n.d.]. The coverage of Microsoft Academic: analyzing the publication output of a university. 113, 3 ([n. d.]), 1551–1571. https://doi.org/10.1007/s11192-017-2535-3
[19] Julien Jomier. [n.d.]. Open science – towards reproducible research. 37, 3 ([n. d.]), 361–367. https://doi.org/10.3233/ISU-170846 Publisher: IOS Press.

[20] Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M. German, and Daniela Damian. [n.d.]. An in-depth study of the promises and perils of mining GitHub. 21, 5 ([n. d.]), 2035–2071. https://doi.org/10.1007/s10664-015-9393-5

[21] Roosa Leimu and Julia Koricheva. [n.d.]. What determines the citation frequency of ecological papers? 20, 1 ([n. d.]), 28–32. https://doi.org/10.1016/j.tree.2004.10.010

[22] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. [n.d.]. S2ORC: The Semantic Scholar Open Research Corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, 2020). Association for Computational Linguistics, 4969–4983. https://doi.org/10.18653/v1/2020.acl-main.447

[23] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Dan S. Weld. [n.d.]. S2ORC: The Semantic Scholar Open Research Corpus. ([n. d.]). arXiv:1911.02782 http://arxiv.org/abs/1911.02782

[24] Luke A. McGuinness and Athena L. Sheppard. [n.d.]. A descriptive analysis of the data availability statements accompanying medRxiv preprints and a comparison with their published counterparts. 16, 5 ([n. d.]), e0250887. https://doi.org/10.1371/journal.pone.0250887 Publisher: Public Library of Science.

[25] Reed Milewicz, Gustavo Pinto, and Paige Rodeghero. [n.d.]. Characterizing the Roles of Contributors in Open-Source Scientific Software Projects. In *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)* (2019-05). 421–432. https://doi.org/10.1109/MSR.2019.00069 ISSN: 2574-3864.

[26] Tsuyoshi Miyakawa. [n.d.]. No raw data, no science: another possible source of the reproducibility crisis. 13, 1 ([n. d.]), 24. https://doi.org/10.1186/s13041-020-0552-2

[27] Audris Mockus, Diomidis Spinellis, Zoe Kotti, and Gabriel John Dusing. [n.d.]. A Complete Set of Related Git Repositories Identified via Community Detection Approaches Based on Shared Commits. In *Proceedings of the 17th International Conference on Mining Software Repositories* (New York, NY, USA, 2020-06-29) *(MSR '20)*. Association for Computing Machinery, 513–517. https://doi.org/10.1145/3379597.3387499

[28] Gede Artha Azriadi Prana, Christoph Treude, Ferdian Thung, Thushari Atapattu, and David Lo. [n.d.]. Categorizing the Content of GitHub README Files. 24, 3 ([n. d.]), 1296–1327. https://doi.org/10.1007/s10664-018-9660-3

[29] Karthik Ram. [n.d.]. Git can facilitate greater reproducibility and increased transparency in science. 8, 1 ([n. d.]), 7. https://doi.org/10.1186/1751-0473-8-7

[30] Huajie Shao, Dachun Sun, Jiahao Wu, Zecheng Zhang, Aston Zhang, Shuochao Yao, Shengzhong Liu, Tianshi Wang, Chao Zhang, and Tarek Abdelzaher. [n.d.]. paper2repo: GitHub Repository Recommendation for Academic Papers. In *Proceedings of The Web Conference 2020* (New York, NY, USA, 2020-04-20) *(WWW '20)*. Association for Computing Machinery, 629–639. https://doi.org/10.1145/3366423.3380145

[31] Jyoti Sheoran, Kelly Blincoe, Eirini Kalliamvakou, Daniela Damian, and Jordan Ell. [n.d.]. Understanding "watchers" on GitHub. In *Proceedings of the 11th Working Conference on Mining Software Repositories* (New York, NY, USA, 2014-05-31) *(MSR 2014)*. Association for Computing Machinery, 336–339. https://doi.org/10.1145/2597073.2597114

[32] Giuseppe Silvestri, Jie Yang, Alessandro Bozzon, and Andrea Tagarelli. [n.d.]. Linking Accounts across Social Networks: the Case of StackOverflow, Github and Twitter. ([n. d.]), 12.

[33] Marcus Soll and Malte Vosgerau. [n.d.]. ClassifyHub: An Algorithm to Classify GitHub Repositories. In *KI 2017: Advances in Artificial Intelligence* (Cham, 2017) *(Lecture Notes in Computer Science)*, Gabriele Kern-Isberner, Johannes Fürnkranz, and Matthias Thimm (Eds.). Springer International Publishing, 373–379. https://doi.org/10.1007/978-3-319-67190-1_34

[34] Patrick Wagstrom, Corey Jergensen, and Anita Sarma. [n.d.]. Roles in a Networked Software Development Ecosystem: A Case Study in GitHub. ([n. d.]), 12.

[35] Amy E. Wahlquist, Lutfiyya N. Muhammad, Teri Lynn Herbert, Viswanathan Ramakrishnan, and Paul J. Nietert. [n.d.]. Dissemination of novel biostatistics methods: Impact of programming code availability and other characteristics on article citations. 13, 8 ([n. d.]), e0201590. https://doi.org/10.1371/journal.pone.0201590 Publisher: Public Library of Science.

[36] XU Yuanjie WANG Shuwen and XU Yuanjie WANG Shuwen. [n.d.]. Influence Mechanism of Code-Sharing on Paper CitationsAn Empirical Analysis on Computer Science Field. 3, 2 ([n. d.]), 93–102. https://doi.org/10.11871/jfdc.issn.2096-742X.2021.02.011

[37] Supatsara Wattanakriengkrai, Bodin Chinthanet, Hideaki Hata, Raula Gaikovina Kula, Christoph Treude, Jin Guo, and Kenichi Matsumoto. [n.d.]. GitHub Repositories with Links to Academic Papers: Open Access, Traceability, and Evolution. ([n. d.]). arXiv:2004.00199 http://arxiv.org/abs/2004.00199

[38] Supatsara Wattanakriengkrai, Bodin Chinthanet, Hideaki Hata, Raula Gaikovina Kula, Christoph Treude, Jin Guo, and Kenichi Matsumoto. [n.d.]. GitHub Repositories with Links to Academic Papers: Public Access, Traceability, and Evolution. ([n. d.]). arXiv:2004.00199 http://arxiv.org/abs/2004.00199

[39] Liguo Yu and Srini Ramaswamy. [n.d.]. Mining CVS Repositories to Understand Open-Source Project Developer Roles. 8–8. https://doi.org/10.1109/MSR.2007.19

[40] Yun Zhang, David Lo, Pavneet Singh Kochhar, Xin Xia, Quanlai Li, and Jianling Sun. [n.d.]. Detecting similar repositories on GitHub. In *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)* (2017-02). 13–23. https://doi.org/10.1109/SANER.2017.7884605

[41] Yu Zhang, Frank F. Xu, Sha Li, Yu Meng, Xuan Wang, Qi Li, and Jiawei Han. [n.d.]. HiGitClass: Keyword-Driven Hierarchical Classification of GitHub Repositories. In *2019 IEEE International Conference on Data Mining (ICDM)* (2019-11). 876–885. https://doi.org/10.1109/ICDM.2019.00098 ISSN: 2374-8486.

6 APPENDICES

APPENDIX I: CODES

- Alex Friedrichsen's GitHub for code (may be inaccessible prior to publication)