Assessing Impact of Outreach through Software Citation for Community Software in Geodynamics

Lorraine J. Hwang, Richard A. Pauloo, and Jane Carlen

Abstract— The Computational Infrastructure for Geodynamics is a community of software users and user-developers who model physical processes in the Earth and planetary interiors. From 2010-2018, the community of researchers published upward of 638 peer reviewed papers in more than 124 venues. We analyzed this corpus of publications to understand the impact of CIG workshops and tutorials, measured through software citation. We automated article analysis using text extraction and tokenization techniques. Patterns in co-mentioned software suggest that usage for some tools cross-cuts many domains, indicating their broad impact. Network analysis of co-authorship and participation in tutorials and hackathons reveal both are effective methods to grow software communities and the importance of developers in bridging research groups and expanding networks. Networks show that workshops broaden the reach of users beyond the developers while hackathons broaden the developer base creating greater collaboration opportunities for hackathon participants.

Index Terms— career development, geodynamics, open source software, software, hackathon, outreach

1 Introduction

THE Computational Infrastructure for Geodynamics (CIG) is a community science driven organization that develops and disseminates software for the study of the Earth's interior and deformation of its crust (geodynamics.org). The CIG community includes users and developers of computational modeling software used to complement field observations, laboratory analysis, and theory of the composition of the Earth and processes such as mantle convection, plate tectonics, and the dynamo acting within it and other planetary bodies. To meet the demand for high-quality, reusable scientific software, CIG maintains an open source software repository to encourage reuse and contributions to the development of software. Actively developed packages follow CIG's best practices [1]. Documentation for all software packages must include a citation statement which is typically to a science publication. In 2017, CIG began archiving and establishing unique identifiers for versioned releases through its Zenodo community (https://zenodo.org/communities/geodynamics). Subsequently CIG has added recommendations to cite software packages using the Zenodo-issued DOIs and also directs users to these citations.

The majority of today's researchers use research software [2], [3], thus, software and its development are critical to researcher success. In the geosciences, more than 60% of graduate students use computer-based methods [4]. In response to this trend, CIG offers software training to its community in two different formats. Traditional training targets users of geodynamics modeling software and is offered as workshops lasting from several hours to several days. Workshops aim to familiarize users with running the software on simple problems so users can gain confidence to apply the computational model to their own research. In contrast, hackathons are longer events that target users who are or are interested in becoming developers. These events center around a single software package and combine formal and informal learning aimed at creating a self-sustaining cohort of user-developers to maintain and expand the software itself. Hackathons aim to improve software, computational, and domain knowledge and create a foundation for future collaborations.

Software development requires significant effort which should be recognized as an intellectual contribution. A goal and measure of effectiveness for CIG is the advancement of careers for the next generation of scientific software developers, many of whom are increasingly engaged in this changing nature of scholarship. Reward systems reflect the values and goals of institutions which often struggle to keep apace. The traditional academic reward system assigns credit through citation of a published work, and measures citations through various indices [5]. One way to recognize those who are launching promising research careers and who are actively engaged in developing research software is by leveraging the established citation system to give credit for code authorship [6].

Software is not used in isolation [7], [8]. Many steps must be taken prior to the execution of computational models used in the CIG community. The physical system to be modeled must be constructed. This may include creating discrete models of the Earth's crust or interior and/or

- *L.J. Hwang is with Earth and Planetary Sciences, University of California, Davis, Davis, CA 95616. E-mail: ljhwang@ucdavis.edu*
- *R.A. Pauloo is with Land, Air and Water Resources; Data Science Initiative, University of California, Davis, Davis, CA 95616. E-mail: rpauloo@ucdavis.edu*
- *J. Carlen is with the Data Science Initiative, University of California, Davis, Davis, CA 95616. E-mail: jacarlen@ucdavis.edu.*

preparing observational data for inverse or adjoint simulations. At all steps, software is used. What software to cite in the course of research is evolving, and arguments have been made at both ends of the spectrum: all software used should be cited vs. only the software regarded as significant to the research outcome. The former is seemingly absolute and clear; the latter nebulous and left to the judgement of the authors, perhaps motivated by what citation can enable credit, transparency, discoverability, and reproducibility.

One limitation of studies of peer reviewed publications is that they measure articles and not author impact. Article co-authorship is a form of a social network [9]. Many studies have focused on the network's global characteristics but less so on individuals within the network [10], [11]. Focusing on individuals and centrality measures can help describe career paths and may provide an alternative way to assess impact [11], [12] and, here, the role and effectiveness of outreach programs.

A previous study [13] on CIG citation practices based on its publication repository from 2010-2015, showed that 83% of the papers mentioned a CIG hosted software package but only 75% of those papers provided a software citation. Here, we add 3 more years of data and automate the text analysis to reduce human error and intervention in discovering and quantifying software mentions for the entire data set. In addition, we look at the co-occurrence of mentions between packages and popular tools, and separately, the co-author relationships for two software packages to provide insight into improving the effectiveness of outreach and community building.

All data used is available in the project archive [14].

2 Data

CIG's publication repository (https://geodynamics.org/cig/news/publications-refbase/) includes publications associated with CIG community activities and software. Software publications are identified either by their authors as using CIG community software or through an automated literature search using a combination of software name, author name, and/or a unique identifier to the software as search terms. We extract only the peer reviewed publications from 2010-2018 to perform content and network analyses. This dataset more than doubles the number of peer reviewed publications (n=638) for analysis from our previous study [13].

The automated literature search is limited to publications indexed by Google Scholar and hence, limited to the English language. We do not search for publications that cite only the references requested by the developers. Research papers are often cited in support of a study and that study may not use the software. These types of papers are excluded from the repository.

Names of some software packages sometime use common words e.g. Relax and Rayleigh. Using software package names that are common words complicates text mining algorithms. Text matches concerning *relax*ation of the crust and the *Rayleigh* number occur frequently as both are common concepts in geodynamics. In these cases, searches are based on both the name of the software and at least one author.

In addition to the use of non-unique words as software package names, frequent misspellings also limited the current study's content analyses. From our earlier study, we developed a whitelist of trusted words that included these variants, as well as variants in lower and uppercase. For some codes, we chose to aggregate code families. For example, CitcomS and CitcomCU are reported as Citcom. We also whitelisted mentions of other commonly cited non-CIG software packages to understand what software packages were frequently mentioned together. Although the sample size of this study has increased, we are confident that these commonly cited packages are still amongst the most common used in our community.

Some publications (<2%) in the resulting list include books, journals, or proceedings that were inaccessible or unavailable through the University's subscription service. This narrowed the documents available for content analysis to 626 publications.

Since 2010, the CIG community has published in 124 unique platforms, which include journals, books, and conference proceedings. The vast majority of journals contribute less than 10 publications each and represent 36% of the collection. The two journals with the most publications are Geophysical Journal International and Journal of Geophysical Research. Both are published by scientific societies, the Royal Astronomical Society and American Geophysical Union, respectively, and promote geophysics, geodynamics, and seismology to further our understanding of Earth processes.

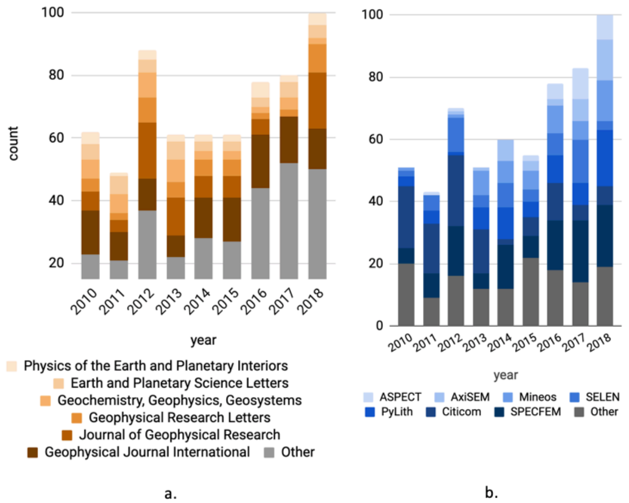

Fig. 1. Publications using CIG software for years 2010-2018. a. The 6 journals with the most publications, and b. the 7 software packages with the most citations are broken out. The remainder are aggregated into Other.

The number of publications from the CIG community has grown steadily in the last decade. Consistent tracking of publications began in 2010, hence prior to this date the record is considered incomplete. This growth is partly explained by software additions to the repository, and the growth of their developer and user bases. For example, PyLith 1.0.0 was first released in 2007, ASPECT 1.0 in 2014, and the SPECFEM family in 2012. The SPECFEM and Citcom family of codes also have a long development history prior to being adopted by CIG and hence, have an established user base. Citcom has been actively developed and used by the community for 25 years - prior to the establishment of CIG in 2005. These long histories are reflected by the relatively larger number of early publications (figure 1).

3 Content Analysis

3.1 Method

We employed methods from natural language processing implemented in R [15] and the pdftools package (version 2.2) [16] to parse the corpus and to build the data pipeline. The pipeline begins with parsing the PDFs, tokenizing sentences, and extracting software mentions.

Next, we iteratively refined a whitelist of software names and known variants [14]. From this list, we located sentences within each document containing matches. Comparing these results to our previous study, we further refined the whitelist by both identifying publications that did not return the expected names and publications that returned unexpected names. Case sensitivity was not an issue for unique names; but for other packages strict adherence to capitalization was needed. The method was verified against the 2010-2015 dataset [13] which was based on manual extraction. Automated extraction identified two additional publications with software mentions. The method was then applied to the full dataset from 2010-2018. 9% of these papers did not identify software, of which <1% is due to punctuation and capitalization issues.

3.2 Co-occurrence of Mentions

Co-occurrence of software mentions in a document, and hence usage, may reveal the dependency of packages on one another and functional and disciplinary similarities [17]. This information is useful in providing recommendations to users for other useful tools, as an indicator to what additional tools should be taught in workshops, and to understand the cross disciplinary nature of the tool itself. It may also convey the importance to the author of proper software attribution and the perceived dependency of the results on the package.

For each publication analyzed, we identified unique software mentions between CIG software and other commonly used packages and proper names to understand their linkages. These co-occurrences are shown in figure 2 and include:

- CIG
- GMT displays map driven data.
- GPlates assembles plate tectonic reconstructions through time.
- MATLAB is a licensed scientific computing and visualization environment.
- Seismology tools:
  - ObsPy enables users to download, manipulate, and process seismological data
  - SAC is a seismic analysis toolkit.
  - Instaseis calculates broadband seismograms from a Green's function databases generated with AxiSEM.
- Zenodo is the archive for CIG software.

GMT (https://github.com/GenericMappingTools/gmt) is the most frequently used package, as it is used across domains to produce map-driven data for seismological, sea level, mantle convection, and crustal deformation studies among others. The tool has been in use by the geoscience community for over 20 years. Releases are accompanied by articles in EOS Transactions AGU - the weekly earth science magazine of the American Geophysical Union - which are used in citation. The authors' request for citation dates back to 1991. The large number of citations of GMT observed in this study can likely be attributed to both its rich tradition of citation in the literature and that the developers provided a citable article.

CIG's high frequency of mention is closely associated with its inclusion in the URL in citations. The seismology family of codes (blue) are often co-mentioned, reflecting the software development and research ecosystem around SPECFEM and AxiSEM. The citation to CIG's repository in Zenodo is sparse but unsurprising as it is typical for several years to pass between a code's release and publication.

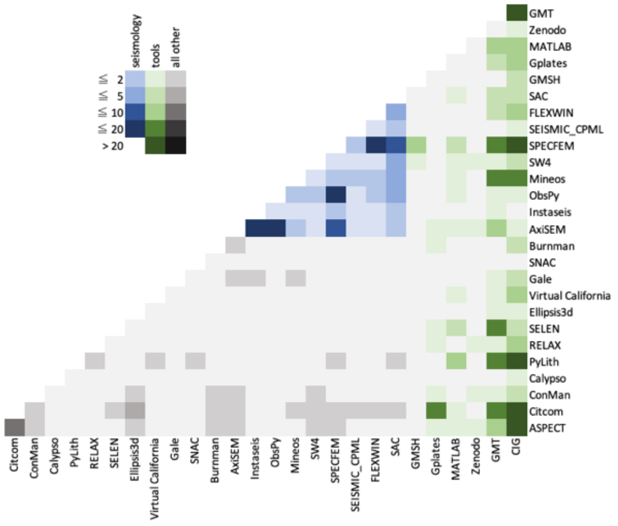

Fig. 2. Co-occurrence of software mentions between CIG software (gray), CIG seismology software (blue), and non-CIG software tools (green).

Co-occurrence should be cautiously interpreted. Text occurrences are not distinguished from References occurrences. Text mentions, for example in the case of ASPECT and Citcom, can be part of the broader discussion of methods or part of a benchmarking exercise.

More information on CIG software packages can be found on our website geodynamics.org/software.

4 Author Networks

4.1 Method

From the entire CIG publication database through the year 2018, we create author networks of peer-reviewed publications. We use the R packages igraph (version 1.2.4) [18] and ggraph (version 1.0.2) [19] to build co-authorship networks for specific software packages and visualize the relationships between authors to examine their importance and role in building software communities.

In the following figures, we map various attributes of the network. Each node in the network represents an author - a larger node size indicates more publications. Each line or “edge” represents a connection to a co-author - the thicker the line, the more co-authored papers. Degree centrality counts how many neighbors a node has. Authors with a higher *degree centrality* have more ties with other nodes. They have the potential to exert greater influence on others and their scientific domain.

Networks can be isolated or well-connected depending on how a research group adopts a code, grows, and establishes new collaborations. For example, we expect research group members to author many papers together, forming a distinct cluster. This cluster may expand as members leave a group and establish their own program and/or form new collaborations with outside research groups. A member may also act as a bridge into their program cluster. In author networks, these nodes have high *betweenness centrality* – a measure of the extent a node lies on the shortest path between other nodes. These authors may engage in interdisciplinary collaborations and are key in knowledge transfer [11].

In this study, network visualizations are grouped by software packages. As discussed above, some software cross-cut domains, and hence, research groups. For software packages whose usage has grown primarily through the growth of academic trainees (e.g. graduate students and early career researchers), we expect to see a higher degree of connectivity between authors

Below we discuss in detail two of these networks whose software packages emerged differently from the community and thus, have employed different approaches to development and outreach.

4.2 PyLith

PyLith traces its roots to a single developer committing to develop a community code to support the study of short-term crustal deformation. Its small development team has been offering workshops since 2007 with the release of PyLith v1.0. The team delivers a week-long workshop for its users annually either in person or virtually. Prior to attending a workshop, participants are expected to work through introductory material available online that includes recordings of past tutorials. Workshops can then concentrate on intermediate skill building and new code features. Extensive “tinker time” is scheduled so participants can work on their own models, get help directly from the developers, and collaborate with attendees.

As early as 2014, citation information was included in the user manual. With release v2.1.4 (2016), this information was available for export via a command line argument. The software is mentioned in 13% of the publication’s dataset. New releases of the software are regularly downloaded worldwide. A large number of downloads originate in Asia, a region known for elevated seismic risk.

In-person PyLith workshops have trained over 350 researchers. From this group of workshop attendees, 12% (n=45) have published papers that mention the usage of PyLith by name. Only 20% of the PyLith papers (n=16) have no identifiable relationship to outreach efforts, an indicator of its broader adoption in the community. The main co-author network extends from two of the three developers (figure 3). This reflects the roles each developer has in code development and research direction.

The development team is comprised of two domain researchers and an expert in numerical methods. Reasonably, the former have more geoscience publications and collaborations and exhibit higher degree and betweenness centrality than the latter, who has very low centrality. We would expect the latter, a computational scientist specializing in numerical methods, to have greater centrality measures in networks that encompass those domains. That developers do not have the largest centrality measures perhaps indicates the tradeoffs between productivity as measured through publications vs. through code development. Researchers outside the developer network with high centrality can be interpreted as a measure of success of outreach to create an expanded and independent user base.

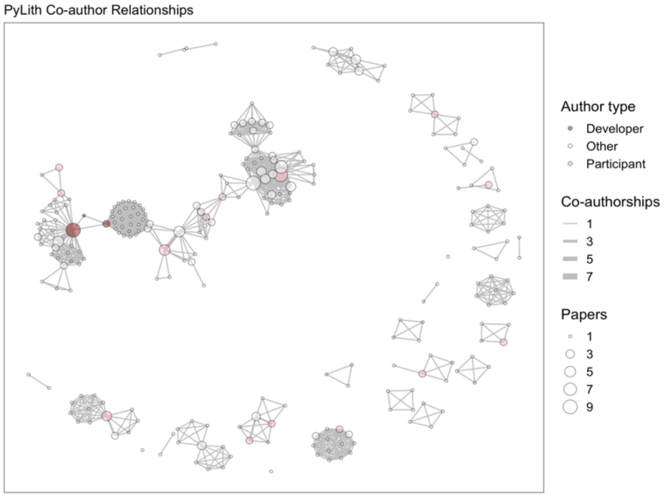

Fig. 3. Author network for publications using the software PyLith.

PyLith virtual workshops have been held in alternate years, with pre-registrants located on 6 continents. Many of these tutorials were recorded and are available online. Pre-registration lists for web-based events are, however, notoriously unreliable and active participation as well as view of tutorial recordings are not tracked. Possibly some authors received training through these means but since attendees are unverifiable, we do not consider this data.

4.3 ASPECT

Development of the thermal convection code ASPECT began in 2011 with CIG support. The package is well documented, and its software and community were conceived to support an open source community model, which actively welcomes and cultivates contributions from users [20]. CIG has been involved with the community from the beginning. The ASPECT community’s close association lends deeper insight into its evolution and relationships to CIG activities such as tutorials, hackathons, and research collaborations.

ASPECT hackathons have been a key driver in growing the community. Hackathons are typically 10-day events of at most 25 user-developers devoted to implementing new features. Selected participants are a complementary mix of domain and computational scientists. The majority of participants are previous attendees who provide mentorship to first time attendees and help maintain its welcoming, collegial, community culture.

As leaders of a community project, the ASPECT founding developers have acknowledged the need for credit, not only to acknowledge contributions, but to advance careers. Credit, software citation, and publication tracking have been part of ASPECT's culture since inception. Requested citations have been provided in BibTeX format in the manual and, more recently, dynamically through community websites at: https://aspect.geodynamics.org/citing.html?src=citation and https://geodynamics.org/cig/abc. Due to these efforts, we consider the publications used below to be a complete data set and have extended the publications dataset used below into early 2019.

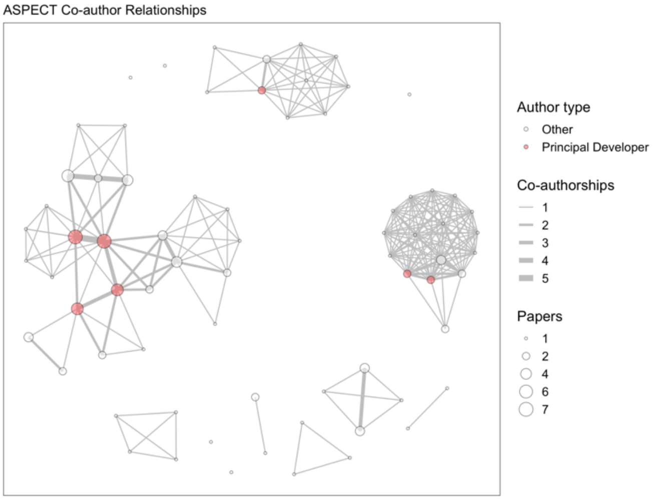

Fig. 4. Author network for publications using the software ASPECT.

The co-author relationships show patterns of connected components of the network clustering around research groups (figure 4). The largest components of the network and the most active authors tend to be bridged by principal developers of the code (red circles), who are among those with the largest degree and/or betweenness centrality [14]. Others with high centrality include research advisors and early adopters who are also in a strong position to form new research collaborations based on their code expertise and/or academic roles.

We also analyzed ASPECT hackathon participation in the context of author networks. The nodes in figure 5 represent hack participants and the edges represent co-participation between attendees. The larger the node size, the larger the number of hacks attended. Principal developers (red circles) have attended nearly all hackathons and are also central and have high degree centrality to the hack networks. Combining co-author relationships with hack attendance (figure 6) shows that authors are highly connected through hacks. Isolated nodes are related through 2 degrees of separation - either an advisor or a research group member was a hack participant Within a domain, close connections are not surprising [9] and less surprising for young communities.

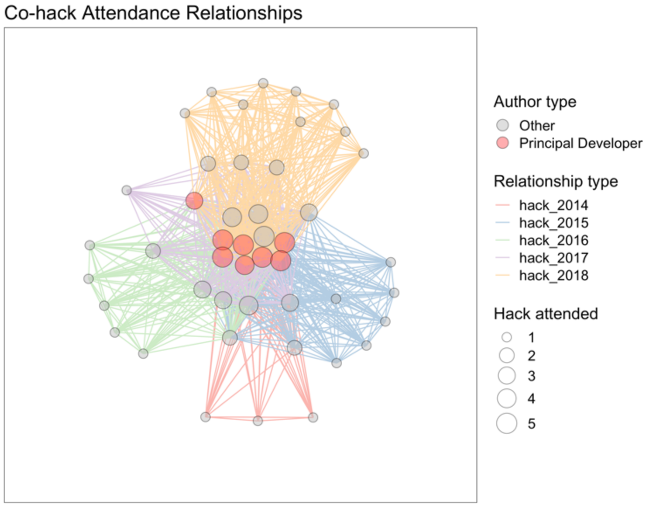

Fig. 5. Hackathon participation network shows Principal Developers (red circles) have attended most hackathons.

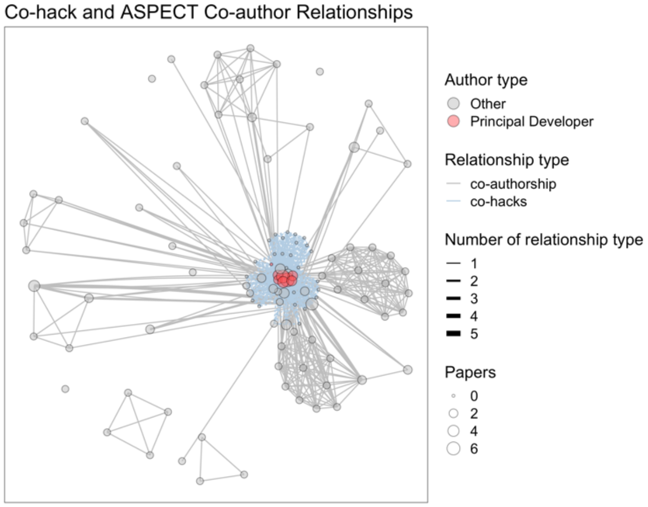

Fig. 6. Combining the hackathon (fig. 5) and author networks (fig. 4) shows that authors are highly related to hackathon participants.

One of the co-author networks not connected to the hackathons has an author who attended a tutorial. However, tutorials alone currently have not been large drivers for code adoption. Tracking of tutorial attendance indicates that <2% of the tutorial participants who were not already established users subsequently attended a hack. Peer networks and ongoing outreach by developers attract new users outside the hack network, and influences code development direction. We expect to see these trends reflected in the growth of the co-author networks with developers as bridges between research networks in the future.

5 Discussion

As a consequence of the data chosen, this study does not include data from researchers who do not mention software or report its usage to CIG. This is perhaps an inherent problem in studies that use citations based on name mentions - the incompleteness of the record [8], [21]. Since the majority of the requested citations for CIG software are to a scientific publication, mining only the requested citation does not identify context of use. On the other hand, requiring the name of the code to be used in the text may miss legitimate references to applications of the code.

Unambiguous citations to the Zenodo DOI are few, unsurprisingly, as their implementation are relatively new in the CIG community. In addition, for many of the codes that can be applied to understanding seismic hazard, an undetermined portion of the user base is international and may publish in non-English-language journals. These journals are not captured in our search. For researchers who have been trained to use or have downloaded CIG software, ambiguity exists over whether they have chosen not to use or not to cite. Carefully constructed surveys of past participants could clarify the range of reasons. However, survey participation is self-selecting and may exclude those who have left the field.

Citation studies based on name mentions are at a disadvantage when software packages use names common in the literature. Confusing application of uppercase in names leads to misspellings in the literature that may be hard to identify. In some cases, this may lead to software packages not identified (false negatives) as well as false positives. Consistent usage and unique software names aid in identification. The increased use of unique identifiers will aid future studies.

CIG outreach events are inconsistent in stressing the importance of citation. Workshops may include formal instruction to participants on how to cite or less formal instruction through contributed presentations. In addition, a reference for the code is required documentation and both PyLith and ASPECT provide additional tools for their users. Since most participants are initiating projects, assessing the impact of outreach events on citation rates requires enough time for research to be completed and published. It should be noted that the emphasis here, however, is not the study of effectiveness of outreach on citation rates but in examining scientific networks through citation.

Scientists tend to use software within their social network [22]. Expanding these networks is fundamental to the health of large open source community projects and a goal of outreach efforts. In the ASPECT community, tutorial participants do not appear to become likely users of the software. However, the data is not yet available to capture the time passed between tutorials, code adoption, and publication. Currently, authors and those on track to publish their first papers are cultivated through collaborative networks through hackathon participants, advisors, or outreach to another user, often a principal developer. Collaborations involve significant investment of time for all parties but are productive in training users to develop their own computational model and, as illustrated above, often result in a co-authored paper. This may change through time as the group grows and is looking to adopt a different outreach strategy to grow its user base.

The PyLith workshop model may be an example of a way to meet a larger demand for researchers interested in jump starting their model development without becoming developers. Hackathons appeal to users who are developers, but successful open source communities have many more users than developers. Hence relying on hackathons is an inefficient way to increase the number of users. The traditional (PyLith) model has led to publications by its attendees, many who have established their own research networks independently of the developers (figure 3). In addition, a large number of authors cannot be positively tracked to a CIG outreach event indicating successful, independent adoption of the code.

The hackathon model is largely viewed in the CIG community as a method to both decrease the bus factor of a code and broaden its functionality from a bottom-up/user-driven approach. Other CIG communities are eager to adopt the model. However, while the mechanics of coordinating a hackathon are replicable, leadership is not. Leadership is a conscious decision not only to dedicate time to the technical but also the social engineering needed to cultivate a community.

6 Conclusions

CIG has been an early supporter of software credit requiring documentation for all codes to include a recommended citation. Adoption of code citation is difficult to measure through name mentions studies as the completeness of the dataset is indeterminate. A long culture of citation for GMT has yielded a large number of citations in this study and is an example of the importance of providing a citable object. The ASPECT community has strengthened citation culture by creating a community through hackathons and by principal developers bridging research groups. For PyLith, most authors who cite the software have attended a tutorial. Yet a significant number have not, perhaps absorbing this practice through the literature, code documentation, or participation in other communities. In lieu of established community norms, journals and their reviewers can play a role in ingraining the practice of software citation into the literature.

For the two groups studied, centrality measures are interpreted within the context of the maturity of a community, its parentage, and its culture. For a young and rapidly maturing community, such as ASPECT, degree centrality conveys the importance of developers in expanding usage and accordingly developers have higher publication rates. However, as the community matures and shifts to other outreach models, such as that practiced with PyLith, we expect connections in author networks to developers to weaken. Hence, proper citation has an increasing role in understanding how code usage expands.

Research software is a creative work and should be acknowledged as such. Newer forms of scholarship should be acknowledged, not as a penalty to those conducting science in the same manner as their academic parents, but to expand and evolve the credit system as new methodologies and technologies evolve. Bestowing credit and acknowledging contributions to software are ingredients to creating communities around software. We hope to understand how software citation contributes to this as we continue to follow the evolution of CIG software communities.

Future studies would benefit by expanded literature searches. Improvements in text mining that could understand context of use and the inclusion of non-English language journals would enable us to expand the data set. Stronger adoption and use by developers and authors of unique identifiers would reduce ambiguity. An in-depth survey of authors would improve our understanding of why certain citation practices prevail, how authors learn to cite, and the role of outreach. Lastly, the ability to identify users who currently anonymously download software would increase any survey's reach and lead to better understanding not only of citation but patterns of adoption.

Acknowledgment

This work and the Computational Infrastructure for Geodynamics is supported by the National Science Foundation award NSF-0949446 and NSF-1550901. We thank Pamela Reynolds, John Naliboff, Daniel Hicks, and anonymous reviewers for their keen comments on earlier versions of this manuscript.

References

[1] L. H. Kellogg, L. J. Hwang, R. Gassmöller, W. Bangerth and T. Heister, "The Role of Scientific Communities in Creating Reusable Software: Lessons from Geophysics," Computing in Science & Engineering, vol. 21, no. 2, pp. 25-35, 1 March-April 2019. doi: 10.1109/MCSE.2018.2883326

[2] S. Hettrick, M. Antonioletti, L. Carr, N. Chue Hong, S. Crouch, D. De Roure, I. Emsley, C. Goble, A. Hay, D. Inupakutika, M. Jackson, A. Nenadic, T. Parkinson, M. I. Parsons, A. Pawlik, G. Peru, A. Proeme, J. Robinson, S. Sufi, "UK research software survey 2014," 2014. https://doi.org/10.5281/zenodo.14809

[3] U. Nangia, and D. S. Katz, "Track 1 Paper: Surveying the U.S. National Postdoctoral Association Regarding Software Use and Training in Research," figshare, 2017. https://doi.org/10.6084/m9.figshare.5328442.v3

[4] Wilson, C. (2014), *Status of Recent Geoscience Graduates*, American Geosciences Institute, Alexandria, VA. 0-922152-99-3

[5] E. Hicks, and J. Melkers, "Bibliometrics as a Tool for Research Evaluation." In Handbook on the Theory and Practice of Program Evaluation, edited by Albert Link and Nicholas Vornatas, 323–49. Cheltenham, UK, and Northampton, MA: Edward Elgar., 2012. https://works.bepress.com/diana_hicks/31/.

[6] D. S. Katz, N. P. Chue Hong, "Software Citation in Theory and Practice," in Mathematical Software – ICMS 2018J, Davenport, M. Kauers, G. Labahn, J. Urban Ed, Cham, Springer, pp. 289-296.

[7] X. Pan, E. Yan, and W. Hua, "Disciplinary differences of software use and impact in scientific literature," Scientometrics, vol. 109, no. 3, pp. 1593–1610, 2016. http://dx.doi.org/10.1007/s11192-016-2138-4

[8] J. Howison, and J. Bullard, "Software in the scientific literature: Problems with seeing, finding, and using software mentioned in the biology literature," Journal of the Association for Information Science and Technology, vol. 67, no. 9, pp. 2137-2155, 2016. doi:10.1002/asi.23538.

[9] M. E. J. Newman, "The structure of scientific collaboration networks," Proceedings of the National Academy of Science of the United States of America, vol. 98, no. 2, pp. 404–409, 2001.

[10] S. Wasserman, and K. Faust, Social network analysis, Cambridge, UK: Cambridge University Press, 1994.

[11] E. Yan, and Y. Ding, "Applying centrality measures to impact analysis: A coauthorship network analysis," J. Am. Soc. Inf. Sci., vol. 60, pp. 2107-2118, 2009. doi:10.1002/asi.21128

[12] B. Cronin and L. I. Meho, "Timelines of creativity: a study of intellectual innovators in information science," Journal of the American Society for Information Science and Technology, vol. 58, no. 13, pp. 1948-1958, 2007. doi: 10.1002/asi.20667

[13] L. J. Hwang, A. Fish, L. Soito, M. Smith, and L. H. Kellogg, L. H., Software and the scientist: Coding and citation practices in geodynamics," Earth and Space Science, vol. 4, pp. 670–680, 2017. https://doi.org/10.1002/2016EA000225

[14] L. J. Hwang, R. A. Pauloo, and J. Carlen, "Project data: assessing impact of outreach through software citation for community software in geodynamics," Zenodo, 2019. doi: 10.5281/zenodo.3311910

[15] R Core Team, "R: A Language and Environment for Statistical Computing," R Foundation for Statistical Computing, Vienna, Austria, URL https://www.R-project.org, 2018.

[16] J. Ooms, "pdftools: Text Extraction, Rendering and Converting of PDF Documents," R package version 1.4.2017. https://CRAN.R-project.org/package=pdftools

[17] K. Li, and E. Yan, "Co-mention network of R packages: Scientific impact and clustering Structure," Journal of Informetrics, vol. 12, no. 2018, pp. 87-100, 2017. doi:10.1016/j.joi.2017.12.0001.

[18] Csardi G, Nepusz T: The igraph software package for complex network research, InterJournal, Complex Systems 1695. 2006.

[19] T.L Pedersen. An Implementation of Grammar of Graphics for Graphs and Networks, v1.0.2, https://cran.r-project.org/package=ggraph, 2018.

[20] W. Bangerth, and T. Heister, "What makes computational open source software libraries successful?" Computational Science & Discovery, vol. 6, no. 1, 015010, 2013. doi: 10.1088/1749-4699/6/1/015010

[21] K. Li, P.-Yi. Chen, and E. Yan, "Challenges of measuring software impact through citations: An examination of the lme4 R package," Journal of Informetrics, vol. 13, no. 1, pp. 449-461, ISSN 1751-1577, 2019. doi: 10.1016/j.joi.2019.02.007.

[22] X. Huang, X. Ding, C.P. Lee, T. Lu, N. Gu, and S. Hall." Meanings and boundaries of scientific software sharing". in Proceedings of conference on computer supported cooperative work, San Antonio, Texas, February 23-27, 2013, New York, ACM, 2013. pp. 423-434

Lorraine J. Hwang is the Associate Director at the Computational Infrastructure for Geodynamics at the University of California, Davis. Her research interests include sustainable software and preservation of historical seismogram data. She received her Ph.D. in Seismology from the California Institute of Technology.

Richard A. Pauloo is a PhD Candidate in Hydrologic Sciences and an Affiliate of the Data Sciences Initiative at the University of California, Davis. His research interests include mathematical modeling of hydrologic systems, machine learning, and natural language processing. Contact him at rpauloo@ucdavis.edu.

Jane Carlen is a postdoctoral scholar at the Data Sciences Initiative at the University of California, Davis. Her research interests include social network modeling and community detection on dynamic networks with applications to urban policy. She received her Ph.D. in Statistics from the University of California, Davis.