Disciplinary differences of software use and impact in scientific literature

Xuelian Pan[1, 2], Erjia Yan[3†], Weina Hua[1]

[1]School of Information Management, Nanjing University, Nanjing, China

[2]Jiangsu Key Laboratory of Data Engineering and Knowledge Service, Nanjing, China

[3]College of Computing and Informatics, Drexel University, Philadelphia, U.S.A.

Abstract

Software plays an important role in the advancement of science. Software developers, users, and funding agencies have deep interest in the use and impact of software on science. This study investigates the use and impact of software by examining how software mentioned and cited among 9,548 articles published in *PLOS ONE* in 12 defined disciplines. Our results demonstrate that software is widely used in scientific research and a substantial uncitedness of software exists across different disciplines. Findings also show that the practice of software citations varies noticeably at the discipline level and software that is free for academic use is more likely to receive citations than commercial software.

Introduction

Software is of vital importance to scientific research—it is employed in a number of practices such as control processes, data analytics, and knowledge dissemination. Scientists believe that software plays a critical role in their research (Howison & Bullard, 2016; Hannay et al., 2009) and consumes as much as 40% of their time in developing and using software (Prabhu et al., 2011; Hannay et al., 2009). They also hold the belief that sharing software benefits the scientific community and, accordingly, have made an effort to reduce the barriers of software use, evidenced by the popularity of free and open software (Pan et al., 2015). It is argued sometimes that academic reputation is a major incentive to many scientists who develop and share software (Trainer et al., 2013; Howison & Herbsleb, 2013).

Although there is a consensus that software is useful to the scientific community,

† Corresponding author: ey86@drexel.edu, College of Computing and Informatics, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104-2875, U.S.A. Tel: +1 (215) 895-1459

software has long been considered as a supporting service instead of a formal research product (Howison & Herbsleb, 2014, p. 2). A number of studies have found that acknowledging the provenance of software is inconsistently practiced (Trainer, Chaihirunkarn, & Herbsleb, 2013; Howison & Bullard, 2016; Pan et al., 2015). Therefore, a clear tension exists: on the one hand, scientists put a lot of effort into developing software and their software benefits the scientific community; on the other hand, software is typically not credited in the same way as publications in the current scientific reward system—as Poisot (2015) noted, "while there exists an incentive to write good papers, there is no clear incentive to write good software" (p. 159).

Lately, scientists have gained awareness of this issue, recognizing that an impact assessment should take into considerations both publications and non-traditional research outputs such as software and data (Piwowar, 2013). Among these non-traditional research outputs, research data have garnered attention from academic and industry communities as researchers and practitioners probed into the workforce of data—including data reuse (Chao, 2011; Rolland & Lee, 2013), data publishing (Candela, Castelli, Manghi, & Tani, 2015), data sharing (Tenopir et al., 2011), data citation (Robinson-García, Jiménez-Contreras, & Torres-Salinas, 2015), and data evaluation (Piwowar & Chapman, 2010).

In contrast, the value of software has yet to be explored and recognized. Scientists have just started to understand the lifecycle of software and its potential implications to scientific research, having conversations in venues such as special issues on software attribution (Poisot, 2015; Segal & Morris, 2008) and workshops sponsored by the U.S. National Science Foundation (Katz et al., 2014; Katz et al., 2015; Stewart, Almes, & Wheeler, 2010). Regardless, many questions remain unanswered, particularly in reference to patterns of software use and citation across different disciplinary communication channels.

Our previous study on software entity extraction in full texts (Pan et al., 2015) enables us to study the disciplinary characteristics of software use and citation. In our previous work, more than two thousand software entities were identified from articles published in *PLOS ONE* in 2014. In this study, we focus on studying how these software entities are

used and cited in a variety of disciplines. Citations in this article refer to formal citations associated with an entry in a references list. Specifically, we address the following questions:

1. How much software is used in scientific literature across diverse disciplines?
2. How much software is cited in scientific literature across diverse disciplines? What are the disciplinary differences of software use and citation?
3. What types of software are more likely to receive citations and why?

The answers to the above questions are valuable in two ways. First, they provide insights into the importance of software in science. Second, they help lay a foundation for designing hybrid metrics to assess the full-spectrum impact of software and help build a more inclusive scientific evaluation system that incorporates digital outputs.

Literature Review

While some scientists argue that software plays a secondary role in scientific research (Howison & Herbsleb, 2010), others hold a different view that software plays a central role, for instance, in fields such as bioinformatics (Huang et al., 2013). Nevertheless, there is almost universal agreement in the scientific community that software plays an important role and that software "can be a source of innovation and can enhance science" (Howison& Herbsleb, 2014, p. 2). Despite its value, software is typically uncounted or discounted in current research evaluations that prioritize traditional publications more than non-traditional research outputs (Hafer & Kirkpatrick, 2009). In recent years, we have witnessed that more non-traditional outputs such as software have been created as the end products of various scientific inquiries and they have been widely adopted, used, and reused in the scientific community (Pan, Yan, Wang, & Hua, 2015; Howison & Bullard, 2016). A survey of the use of software and database has found that 97.7% of BMC Bioinformatics papers contained software/database (Duck et al., 2013).

Previous studies on software largely focused on investigating the motivations of software development and sharing (Crowston, Howison, & Wiggins, 2010). It is found that academic reputation and monetary rewards motivate scientists to make their software free for academic use (Hann, Roberts, & Slaughter, 2004; Poisot, 2015). There is a belief that

scientists participate in developing and sharing scientific software for extrinsic benefits such as earning citations and advancing careers (Roberts, Hann, & Slaughter, 2006; Howison & Herbsleb, 2011). Meanwhile, studies have demonstrated that intrinsic motivations, along with learning (Huang et al., 2013) and use value (Howison & Herbsleb, 2013) can also become the drives to develop and share software (Lakhani & Wolf, 2003).

In addition to these motivation studies, scientists have recently embarked on the issue of software use and impact. A study in 2013 has found that scientists tend to choose software that is widely used by others in their community and prefer software that is free for academic use (Huang et al., 2013). Studies on the scientific software ecosystem have suggested that the use of scientific software is influenced by its visibility, availability, sustainability, reproducibility, and citation (Howison and Herbsleb, 2014; Howison et al., 2015; Huang et al., 2013). Studies also have suggested that software developers are interested to know the use and impact of their software because "software use matters to them for funding purposes" (Howison et al., 2015; Trainer, Chaihirunkarn, Kalyanasundaram, & Herbsleb, 2015, p. 428).

Recent studies on data impact have led to the discussions on software citation and evaluation, as a parallel can be drawn between software and data in scientific literature (Piwowar, Carlson, & Vision, 2011; Howison & Bullard, 2016). It is suggested that the numbers of mentions and citations in literature can be used to measure the impact of software (Huang et al., 2013; Pan et al., 2015). Yet, it is argued that "the practices of citation to software vary considerably from field to field and appear to miss significant software" (Howison et al., 2015, p. 478). One study examining the use of software in scientific articles in biology has found that more than half of the software mentions did not include references (Howison & Bullard, 2016). Thus, it validates the need to use alternative metrics in addition to citations when assessing software impact, such as the numbers of downloads, registered users, subscribers, user reviews, and artifacts inserted in literature (Howison et al., 2015).

Software users and developers yearn for information about the use and impact of software on science. Moreover, funding agencies also benefit from having access to software impact data (Piwowar, 2013). In this paper, we examine how software is used

across different disciplines and demonstrate software's popularity and impact in scientific literature.

Data and Methods

All articles published in a multidisciplinary journal *PLOS ONE* in 2014 were downloaded for analysis. The access point for this data set is provided by the PubMed Central Open Access Subset (http://www.ncbi.nih.gov/pmc/tools/openftlist/) which is freely accessible to the public. Jsoup, a java HTML parser, was used to extract the text of papers from the HTML files (Jsoup). The methods/methodology sections of these articles were selected as the intermediate data set to learn software entities because our previous study found that most items of software were mentioned in these sections (Pan et al., 2015). This data set, which contains 9,571 papers, was used as the input to extract software entities. An improved bootstrapping method proposed in our previous work (Pan et al., 2015) was used to learn software entities from the data set. This bootstrapping method is a weakly supervised method that required a small number of seed terms and an unlabeled text corpus as input. It began by generating candidate patterns using seed terms (e.g., BibExcel, LIBSVM, SPSS, and SAS). Then, candidate patterns were sorted and the top-ranked patterns were used to identify candidate entities. Next, candidate entities were scored and high scoring entities were selected as learned entities. After that, the learned entities were used to generate patterns that can extract more entities in an iterative way. To improve the method performance, we employed a pattern accuracy measure and multiple entity features to filter unlabeled entities. A random sample of 386 papers was selected as the test set and the 470 manually labeled software entities in this set were considered as the gold standard for evaluating the performance of the bootstrapping method. The precision and recall scores of the method by the end of the iteratively learning process were 0.94 and 0.42 respectively. Overall, this method had the highest F1 score of 0.58 and outperformed the baseline methods. Then, this method was used to extract software entities from the 9,571 papers and 2,342 unique software entities were learned from this data set.

Papers without a pre-assigned *PLOS ONE* category were discarded, resulting in

9,548 papers. We identified 2,334 unique software entities from these papers. Table 1 reports the distribution of papers in each of the 24 *PLOS ONE* categories. As shown in Table 1, there are substantial differences in the number of papers among the categories (a possible typo in the downloaded file was found: the last category "Biology and life gsciences" might be "Biology and life sciences").

TABLE 1. The distribution of papers across disciplines.

Rank	Discipline	Papers	Rank	Discipline	Papers
1	Biology and life sciences	6288	13	Mathematics	157
2	Medicine and health sciences	4569	14	Agriculture	138
3	Biology	1675	15	Chemistry	134
4	Research and analysis methods	1653	16	Computer science	109
5	Medicine	1346	17	Engineering	107
6	Physical sciences	879	18	Veterinary science	90
7	Ecology and environmental sciences	645	19	Physics	84
8	Social sciences	519	20	Science policy	80
9	Computer and information sciences	445	21	People and places	68
10	Earth sciences	423	22	Materials science	42
11	Engineering and technology	361	23	Astronomical sciences	1
12	Social and behavioral sciences	199	24	Biology and life gsciences	1

Considering that the 24 categories provided by *PLOS ONE* still have room for refinement, we further grouped them into 12 disciplines based on disciplinary similarities. Table 2 summarizes the information of the new 12 collapsed disciplines. This refinement helps us conduct the analysis and draw concise conclusions.

TABLE 2. Summary of the 12 new disciplinary categories.

Collapsed category	Original categories	No. of papers
Biology	Biology; Biology and life sciences; Biology and life gsciences; Veterinary science	7,971

Medicine and health sciences	Medicine and health sciences; Medicine	5,915
Research and analysis methods	Research and analysis methods	1,653
Physics	Physics; Astronomical sciences; Physical sciences	964
Social sciences	Social sciences; Social and behavioral sciences; People and places; Science policy	785
Ecology and environmental sciences	Ecology and environmental sciences	645
Computer and information sciences	Computer science; Computer and information sciences	554
Engineering	Engineering and technology; Engineering; Materials science	496
Earth sciences	Earth sciences	423
Mathematics	Mathematics	157
Agriculture	Agriculture	138
Chemistry	Chemistry	134

In this article, we count the number of mentions and number of citations to assess the impact of software on science. A citation in *PLOS ONE* is represented as square brackets with an integer that is reference ID, such as "[1]". For example, in the sentence "In this paper, Webometric Analyst 2.0 and Weka 3.0 were used to extract and analyze the statistical data of each paper [1] [2]", the number of citations of "Webometric Analyst" and that of "Weka" are one, because a citation occurred after each software. A random sample of 100 sentences, which contain one or more software entities and at least a citation occurring after the software entities, was used to test the accuracy of the assumption. We manually checked if a citation occurred in the substring that starting from a software entity to the end of the sentence is the citation to the software—this has been true for all the 100 sentences.

We use the sentence and article as the counting unit separately. The following two formulas are used to calculate the numbers of mentions and citations when we use the sentence as the counting unit. The number of mentions of $software_i$ in a discipline is calculated as

$$Mentions_{software_i} = \sum_{p=1}^{n} \sum_{s=1}^{m_p} MScore(software_i)$$

where n is the number of articles in a discipline and m_p is the number of sentences in article p. If a sentence contains $software_i$, $MScore(software_i)$ is 1; otherwise, it equals 0. Similarly, the number of citations is calculated using the following formula:

$$Citations_{software_i} = \sum_{p=1}^{n} \sum_{s=1}^{m_p} CScore(software_i)$$

We made an assumption that if a sentence contains $software_i$ and there is a citation in the substring that starting from $software_i$ to the end of this sentence, $CScore(software_i)$ equals 1 and 0 otherwise. That is to say, for each software entity mentioned in an article, we counted the number of sentences that mentioned it. Then, we counted the number of sentences as the number of software mentions of this article. Finally, we aggregated the number of software mentions of each article belonging to a discipline as the number of software mentions of this discipline. For each mention we also assessed whether there is a citation, and again, we aggregated citations at the article and discipline level as the number of software citations.

Additionally, we also used the article as the counting unit: if a software entity occurs in an article, no matter the number of occurrences, its number of mentions is one; otherwise, its number is zero. When we counted the number of citations of a software entity, all sentences that mentioned the software in an article will be assessed whether there is a citation: if there is a citation, no matter the number of occurrences, its number of citations is one. Then, we aggregated the mention and citation numbers at the discipline level.

Article #1 contains three sentences that mentioned software:

Sentence 1: In this paper, Webometric Analyst 2.0 and Weka 3.0 were used to extract and analyze the statistical data of each paper [1] [2].

Sentence 2: Each query was submitted to an application programming interface using Webometric Analyst to get the statistical data of each paper.

Sentence 3: The Weka [2], VOSViewer [8] and CiteSpace [9] software were employed to analyze the dataset.

Results:

Four software entities were mentioned in Article #1: Webometric Analyst, Weka, VOSViewer, and CiteSpace.

Using the sentence as counting unit: the number of software mentions of Article #1 is 6; the number of software citations is 5.

Using the article as counting unit: the number of software mentions of Article #1 is 4; the number of software citations is 4.

Fig. 1. An example of how we count the numbers of software mentions and citations

A program was written to count the number of mentions and citations for every learned software entity. This program matched 2,342 software entities from the data set. Some constrains were put on the matching process to improve the accuracy of counting the number of mentions and citations of software entities. When a learned software entity contains capital letters, the matched term should contain at least one uppercase letter or be in certain context that contains positive trigger words (i.e., package, program, software, tool, toolbox, and toolkit) or version numbers. If a learned software entity does not contain capital letters, the matched term is not required to contain capital letters.

It is worth noting that the unit of analysis is at the sentence level and one entity's multiple occurrences within a sentence are counted only once. For instance, for the sentence "We used the SPSS (SPSS for Windows, Version 18.0, Chicago, IL, USA) to analyze the dataset", the number of mentions is one and the number of citations is zero. In addition, for simplicity, version information of software entities is ignored. For example, SAS 9.2 and SAS 9.3 are consolidated as SAS. Variants of a software entity are also consolidated. For example, ImageJ and Image J are consolidated as ImageJ.

Three sample sets are used to explore what types of software are more likely to receive citations. First, we take a random sample of 30 software entities that did not

receive any formal citations and manually check whether these software entities are commercial software. Second, top 10 most frequently mentioned software entities in each discipline are selected as a sample set to test whether software that is free for academic use is more likely to receive citations. These software entities are classified into two groups based on whether they are commercial and the differences of software uncitedness (the ratio of the number of software mentions minus the number of software citations to the number of software mentions) for the two groups are assessed using IBM SPSS statistics (SPSS, version 20; IBM Corp., Armonk, NY). Third, top 10 most highly cited software entities of every discipline are selected and classified into two groups based on whether they are commercial. The software entities of noncommercial group are divided into two smaller groups based on whether the developers of these software entities request users to make a citation to their software or related publication. The average of software uncitedness of each group is calculated and compared with that of other groups.

Results

In the first subsection, for each of the 12 disciplines, we present results on (1) the distribution of items of software, (2) software use and citation, and (3) software uncitedness. In the second subsection, we explore what types of software are more likely to receive citations using top mentioned and cited software in each discipline.

Disciplinary characteristics of software use and citation

Figure 2 shows the distribution of the 2,334 software entities across the 12 disciplines. The 361 (15.47%) software entities are used in one field; 683 (29.26%) are used in two fields and 474 (20.31%) are used in three fields. Twelve pieces of software (i.e., ArcGIS, ClustalW, Cluster X, ESTIMATES, ImageJ, JMP, MATLAB, Microsoft Access, Microsoft Excel, SAM, SAS, SPSS) are used in all 12 disciplines.

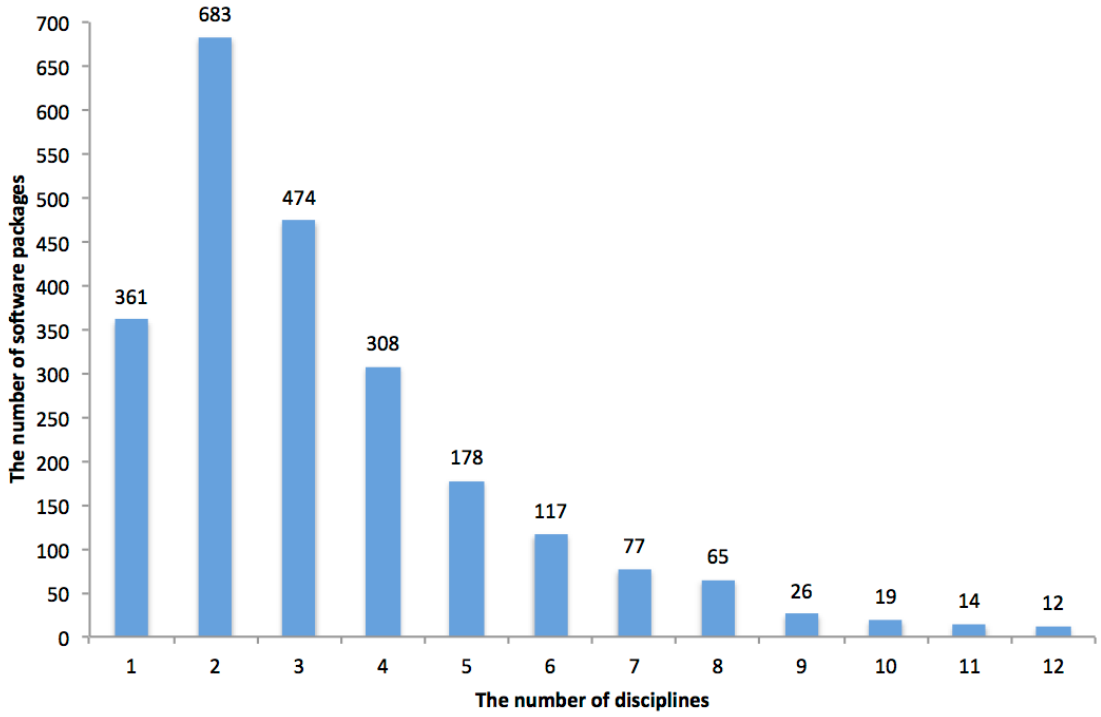

Fig. 2. The number of items of software vs. the number of disciplines

Among the 9,548 papers, 7,602 papers (79.62%) mentioned software. Table 3 reports the distribution of software entities across disciplines.

TABLE 3. The distribution of items of software across disciplines (disciplines ordered by the percentage of papers mentioned software).

Disciplines	Papers	Papers that mentioned software	Percentage of papers mentioned software
Agriculture	138	118	86%
Medicine and health sciences	5,915	4,795	81%
Biology	7,971	6,400	80%
Research and analysis methods	1,653	1,318	80%
Ecology and environmental sciences	645	474	73%
Chemistry	134	98	73%

Engineering	496	351	71%
Physics	964	676	70%
Earth sciences	423	285	67%
Social sciences	785	494	63%
Computer and information sciences	554	342	62%
Mathematics	157	96	61%

A disciplinary difference in the distribution of software is found: while 86% of the agriculture articles contained software, 61% of articles in mathematics contained software. It seems that software is more widely used in some disciplines (e.g., Agriculture, Medicine and health sciences, and Biology) than the others (e.g., Mathematics, Computer and information sciences, and Social sciences). A previous study found that 65% of the 90 sampled biology papers had software mentions (Howison & Bullard, 2016), while our study shows that 80% of biology articles mentioned software entities.

The 2,334 software entities are mentioned in the 7,602 papers and they are in total mentioned 25,860 times and cited 7,381 times. On average, an article mentioned software 3.40 times and cited software 0.97 times. The numbers of software mentions of each discipline using the sentence as counting unit are shown in Table 4. It is worth noting that papers with no software mentions are ignored when we calculated the mentions per article of each discipline.

TABLE 4. The mention of software in the scientific lecture across disciplines using the sentence as counting unit (disciplines ordered by number of mentions).

Disciplines	Papers	Total mentions	Mean mentions	Median mentions	Mode mentions
Biology	6,400	23,392	3.66	2	1
Medicine and health sciences	4,795	13,268	2.77	2	1
Research and analysis methods	1,318	3,951	3.00	2	1
Physics	676	2,086	3.09	2	1
Ecology and environmental sciences	474	1,928	4.07	2	1

Computer and information sciences	342	1,425	4.17	3	1
Social sciences	494	1,268	2.57	2	1
Engineering	351	1,089	3.10	2	1
Earth sciences	285	909	3.19	2	1
Agriculture	118	565	4.79	3	1
Chemistry	98	346	3.53	2	1
Mathematics	96	274	2.85	2	1

Note: Papers indicates the number of papers that mentioned software; Mean mentions = Total mentions/Papers.

As shown in Table 4, the mean software mentions varies from one discipline to another ranging from 2.57 (Social sciences) to 4.79 (Agriculture). Ten out of 12 disciplines have a median of 2. Only Agriculture and Computer and information sciences have a higher median of 3. Before we conducted pairwise comparison of disciplines for software mentions, papers belonging to two disciplines are removed from the paper list of each discipline to ensure accuracy. Because of the non-normal distribution of these disciplines based on software mentions, a series of Mann-Whitney U-tests are employed to identify which disciplines mentioned software significantly different from the others (Table 5).

TABLE 5. Mann-Whitney U-tests for comparison of disciplinary differences in software mentions.

Discipline	Phy	Che	Bio	Soc	Med	Com	Mat	Eng	Ear	Eco	Res
Agr	**0****	**0****	0.415	**0****	**0****	**0****	**0****	**0****	**0****	**0****	**0****
Phy		0.797	0**	**0****	0.008**	**0.004****	**0.001****	0.189	**0.012***	0.203	0.050*
Che			0.003**	**0****	0.237	**0.001****	**0****	**0.004****	**0.002****	0.360	0.251
Bio				**0****	**0****	**0****	**0****	**0****	**0****	0.126	**0****
Soc					0**	0.174	0.662	0.004**	0.041*	0**	0**
Med						**0****	**0****	**0****	**0****	0.101	0.395
Com							0.08	0.967	0.265	0.062	0.008**
Mat								0.023*	0.360	0.001**	0**
Eng									**0.005****	0.679	0.799

Ear	0.002**	0.066
Eco		**0.008****

Note. *Singificant at p = 0.05; **significant at p = 0.01; p value that is displayed in bold indicates the discipline in column is lower than discipline in row (e.g., the p value in the 2th row and 2th column that is displayed in bold means that physics is lower than agriculture in the number of software mentions).

Table 5 shows that agriculture significantly differs from the other disciplines (with the exception of biology) in software mentions. Significant differences are also found between biology and the other disciplines (with the exception of agriculture and ecology and environmental sciences) in terms of the number of software mentions. Scientists in agriculture and biology are more likely to mention software in their articles, while scholars in social sciences and mathematics are less likely to do so.

Table 6 shows the number of software mentions and citations of each discipline using articles as the counting unit. The modes of software mentions and citations of all the disciplines are one. A widespread uncitedness is found in our data set; only between 22% (Medicine and health sciences) and 54% (Ecology and environmental sciences) of the software mentions included references. In biology, 66% of software mentions did not receive any formal citations. In contrast to Howison & Bullard's (Howison & Bullard, 2016) finding (which reported an uncitedness of 56%), our uncitedness is higher. This might be explained by that 24% of the journals that Howison and Bullard used in their research had explicit policies on how to cite software but *PLOS ONE* did not have that in 2014.

TABLE 6. The mention and citation of software in the scientific lecture across disciplines using the article as counting unit (disciplines ordered by number of mentions).

Disciplines	Papers	Total mentions	Total citations	Mean(Median) mentions	Mean(Median) citations	Uncitedness
Biology	6,400	18,257	6,285	2.85 (2)	0.98 (2)	0.66
Medicine and health sciences	4,795	10,842	2,364	2.26 (2)	0.49 (1)	0.78
Research and analysis methods	1,318	3,125	784	2.37 (2)	0.59 (1)	0.75

Physics	676	1,569	534	2.32 (2)	0.79 (1)	0.66
Ecology and environmental sciences	474	1,425	771	3.01 (2)	1.63 (2)	0.46
Social sciences	494	963	277	1.95 (1)	0.56 (1)	0.71
Computer and information sciences	342	888	437	2.60 (2)	1.28 (1)	0.51
Engineering	351	776	219	2.21 (2)	0.62 (1)	0.72
Earth sciences	285	677	299	2.38 (2)	1.05 (1)	0.56
Agriculture	118	440	192	3.73 (2)	1.63 (3)	0.56
Chemistry	98	254	71	2.59 (2)	0.72 (1)	0.72
Mathematics	96	189	61	1.97 (2)	0.64 (1)	0.68

Note: Papers indicates the number of papers that mentioned software; Mean mentions = Total mentions/Papers; Mean citations = total citations/Papers; Uncitedness = (Mean mentions – Mean citations)/Mean mentions.

A series of Mann-Whitney U tests are used to assess whether there are differences between disciplines in citing software. Scientists in ecology and environmental sciences, and computer and information sciences are more likely to cite software when they mention software in their articles, while scientists in medicine and health sciences and research and analysis methods are less likely to make a formal citation for software that they mentioned in their articles

TABLE 7. Mann-Whitney U-tests for comparison of disciplinary differences in uncitedness.

Discipline	Phy	Che	Bio	Soc	Med	Com	Mat	Eng	Ear	Eco	Res
Agr	0.112	0.038*	0.453	0.025*	0**	**0.049***	0.339	0.023*	0.170	**0.002****	0**
Phy		0.211	0.640	0.327	0**	**0****	0.560	0.353	**0****	**0****	0**
Che			0.376	0.776	0.005**	**0****	0.408	0.709	**0.002****	**0****	0.127
Bio				0.075	0**	0.471	0.497	0.099	0.910	0.159	0**
Soc					0**	**0****	0.988	0.727	**0****	**0****	0**
Med						**0****	**0****	**0****	**0****	**0****	**0****
Com							0.004**	0**	0.416	0.129	0**
Mat								0.472	**0.003****	**0****	0.010**
Eng									**0****	**0****	0.007**
Ear										**0.003****	0**

Eco	0**

Note. *Singificant at p = 0.05; **significant at p = 0.01; p value that is displayed in bold indicates the discipline in column is lower than discipline in row.

Figure 3 shows the software mention ratio (mean mentions per article) and citation ratio (mean citations per articles) of each discipline using the article as counting unit.

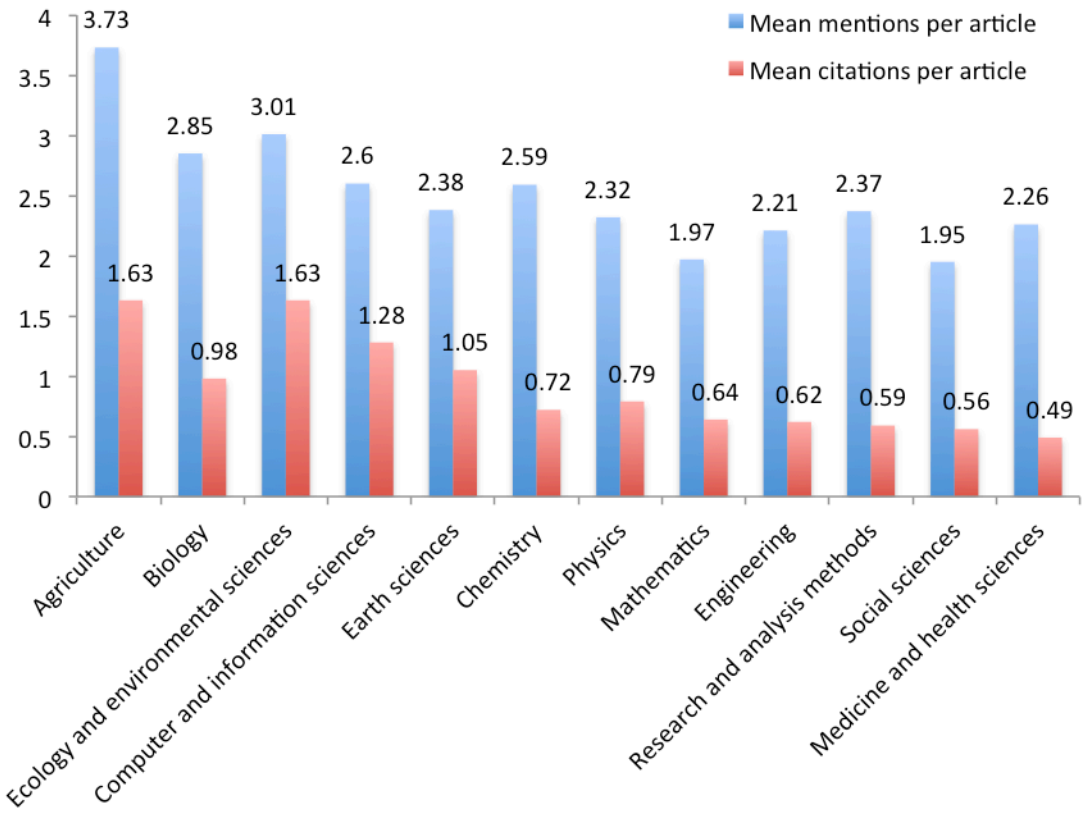

Fig. 3. The mean mention ratio and mean citation ratio of the 12 disciplines using the article as counting unit.

The 12 disciplines were separately sorted in descending order based on the two ratios. The top six disciplines with the highest software mention ratio and the bottom six disciplines were classified into the high mention ratio group and low mention ratio group,

respectively. Similarly, the top and bottom six disciplines based on mean citation ratio were separately classified into the high citation ratio group and low citation ratio group. The 12 disciplines were assigned into four groups:

- high mention ratio and high citation ratio: Agriculture, Biology, Ecology and environmental sciences, Computer and information sciences, and Earth sciences;
- high mention ratio and low citation ratio: Chemistry;
- low mention ratio and high citation ratio: Physics; and
- low mention ratio and low citation ratio: Mathematics, Engineering, Research and analysis methods, Social sciences, and Medicine and health sciences.

Table 8 shows the numbers of software entities that are mentioned and cited in each field. Percentages of software entities that received no citations in each discipline are also calculated. This metric shows that software citation is practiced to a greater extent in fields such as environmental sciences, computer and information sciences, and earth sciences. On the other hand, more than 60% of the mentioned software received no citation in chemistry. Our results demonstrate the need to take into considerations the number of software mentions in full texts when assessing the impact of software on science.

TABLE 8. The number of software entities in each discipline (disciplines ordered by the percentage of uncited software entities).

Discipline	Mentioned software entities	Cited software entities	Percentage of uncited software entities
Chemistry	165	55	67%
Mathematics	119	53	55%
Engineering	358	160	55%
Research and analysis methods	876	400	54%
Social sciences	313	147	53%
Medicine and health sciences	1611	792	51%
Physics	599	308	49%
Agriculture	257	136	47%
Biology	2251	1317	41%

Computer and information sciences	436	276	37%
Earth sciences	289	186	36%
Ecology and environmental sciences	480	314	35%

Characters of software that is more likely to receive citations

We randomly selected 30 software entities that were never cited in the reference list to assess whether they are commercial. After manually checked all the 30 unique software entities, we found that 18 (60%) software entities were commercial and 12 (40%) were free for academic use. It seems reasonable that commercial software entities are less likely to receive citations because they usually have no citation targets like publications. To demonstrate this assumption, the top 10 most frequently mentioned software entities in each discipline are selected as a sample set. Table 9 lists these software entities and their numbers of mentions. Among the 44 unique software entities in Table 9, 26 items of software (59%) are free for academic use. We grouped the 44 software entities into two classes based on whether they are commercial. Then, we counted the uncitedness of each software entities for the two classes. Because of the uncitedness distribution is skewed, the Mann-Whitney U test is employed to assess the difference between the two groups for uncitedness. Commercial software entities have a significantly greater uncitedness than those that are free for academic use (two tailed Mann-Whitney U test: $p < 0.05$). It means that commercial software is less likely to receive citation than software that is free for academic use. We also found that a few statistical software (e.g. SPSS, SAS) and image processing software (e.g. ImageJ) are widely used across several fields because of their marked applicability. For each discipline, more than three out of the top 10 most frequently mentioned software entities are free for academic use. Our results show the popularity of free software in different disciplines.

TABLE 9. Top 10 most frequently mentioned software in each discipline using the sentences as the counting unit

Discipline	Top 10 most frequently mentioned software (the number of mentions)
Agriculture	SPSS (24); **MEGA** (15); **BLAST** (14); JMP (14); SAS (13);

	STRUCTURE (10); **BLASTX** (9); **RDP** (9); **PRIMER** (8); AxioVision (8)
Biology	SPSS (1330); **ImageJ** (1011); SAS (431); MATLAB (417); **BLAST** (398); **MEGA** (356); EXCEL (344); Stata (305); FlowJo (258); **PRISM** (242)
Chemistry	SPSS (19); SigmaPlot (10); **ImageJ** (10); SAS (9); AMBER (9); MOE (8); **ENM** (8); JMP (7); EXCEL (7); **GROMACS** (7)
Computer and information sciences	MATLAB (77); **SPM** (30); **Pfam** (29); SPSS (29); **PSI-BLAST** (22); **Weka** (22); **GSEA** (21); **BLAST** (18); ArcGIS (18); SAS (17)
Earth sciences	SPSS (55); ArcGIS (52); SAS (30); **Mothur** (27); **MEGA** (20); **ImageJ** (18); **QIIME** (17); **MaxEnt** (17); EXCEL (16); MATLAB (15)
Ecology and environmental sciences	SPSS (80); ArcGIS (66); SAS (52); **VEGAN (44)**; **QIIME** (43); **MEGA** (40); **BLAST** (39); **ImageJ** (36); **ARLEQUIN** (35); **Mothur** (32)
Engineering and technology	MATLAB (82); SPSS (61); **ImageJ** (57); **SPM** (34); SAS (32); **FSL** (19); SVS (18); **GSEA** (17); EXCEL (16); **fastICA** (14)
Mathematics	SAS (18); **SPM** (15); SPSS (15); MATLAB (15); Stata (15); **Pfam** (11); PLS (7); **Globaltest** (7); **STAR** (6); **SAM** (6)
Medicine and health sciences	SPSS (1461); **ImageJ** (644); Stata (553); SAS (448); MATLAB (237); EXCEL (230); FlowJo (195); **PRISM** (186); **SPM** (183); Adobe Photoshop (139)
Physics	SPSS (107); MATLAB (107); Stata (85); **ImageJ** (74); SAS (41); EXCEL (39); **SPM** (31); **FSL** (19); **REVIEW MANAGER** (19); **BLAST** (18)
Research and analysis methods	SPSS (321); **ImageJ** (216); Stata (159); MATLAB (127); SAS (97); EXCEL (88); Adobe Photoshop (54); **PRISM** (49); Ingenuity Pathway Analysis (47); **BLAST** (46)
Social sciences	SPSS (144); **SPM** (91); Stata (83); MATLAB (68); SAS (46); EXCEL (26); Adobe Photoshop (26); E-Prime (25); **Talairach** (19); **SAM** (18)

Note: Software that is free for academic use is displayed in bold.

Table 10 shows the top 10 most highly cited software of each discipline. There are 62 unique software entities in Table 10 and 50 (81%) are free for academic use. Free software is more likely to receive citations, which may be explained by that developers of free software typically request users to make a citation to their software or citable publications. To further investigate this phenomenon, we manually checked information

for these highly cited software entities. We find that developers of 30 pieces of software (60%) provided information on how to cite their software in their websites. Our results suggest that scientists expect a proper acknowledgment of their work—be the publications or digital outputs—and thusly have a keen interest in the impact of their software, which in turn provides evidence for the value of their work in the scientific community. Moreover, the 62 unique software entities are classified into three groups: the first group contains the 12 pieces of commercial software; the second group contains the 30 pieces of software whose developers provided information on how to cite their software in their websites; the third group contains the remained 20 pieces of software. We count the average uncitnedness for the three groups: the first group is 0.66, the second is 0.47, and the third is 0.36. This might be explained by software that is free for academic use usually has related publications and its developers are more likely to tell the users how to cite the software. That means providing citation target and the information about how to cite the software might improve the practice of software citations.

TABLE 10. Top 10 most highly cited software of every discipline using the sentences as the counting unit

Discipline	Top 10 most highly cited software (number of mentions)
Agriculture	***MEGA*** (9); **BLAST** (6); ***Mothur*** (5); ***STRUCTURE*** (5); ***PHYLIP*** (5); ***RDP*** (4); ***ARLEQUIN*** (4); **Blast2GO** (3); ***Pfam*** (3); ***QIIME*** (3)
Biology	***MEGA*** (233); **ImageJ** (115); **BLAST** (106); ***MUSCLE*** (96); ***Clustal W*** (94); ***ARLEQUIN*** (81); **MrBayes** (75); **BioEdit** (68); ***STRUCTURE*** (66); ***Bowtie*** (62)
Chemistry	**Modeller** (5); AMBER (5); ***REFMAC*** (4); MOE (4); ***PHENIX*** (3); **PyMOL** (2); ***Mothur*** (2); HKL-2000 (2); PHASER (2); **CO2SYS** (2)
Computer and information sciences	**PSI-BLAST** (10); MATLAB (9); **SPM** (8); ***Weka*** (8); **GROMACS** (7); AMBER (7); ***LIBSVM*** (6); **SAM** (6); **BLAST** (6); **MaxEnt** (6)
Earth sciences	***MEGA*** (15); **MaxEnt** (14); ArcGIS (14); ***Mothur*** (13); ***VEGAN*** (9); ***QIIME*** (7); **Random Forests** (7); ***PAST*** (6); ***TNT*** (5); ***PRIMER*** (5)
Ecology and environmental sciences	***VEGAN*** (32); ***MEGA*** (31); ***ARLEQUIN*** (25); ***Mothur*** (22); **MaxEnt** (20); ***STRUCTURE*** (17); ArcGIS (17); ***QIIME*** (15); **MrBayes** (15); ***RDP*** (15)

Engineering and technology	**SPM** (11); **ImageJ** (10); MATLAB (8); ***FSL*** (8); SVS (5); ***MEGA*** (5); ***Refmac*** (4); ***RDP*** (4); ASA (4); ***Mothur*** (3)
Mathematics	**SPM** (7); **STAR** (3); **PSI-BLAST** (3); TAC (3); ***EMBOSS*** (2); **SAM** (2); MATLAB (2); **Globaltest** (2); ***SPINE-X*** (2); **MaxEnt** (2)
Medicine and health sciences	**ImageJ** (60); Stata (53); ***MEGA*** (52); **SPM** (42); ***PLINK*** (35); MATLAB (34); SPSS (30); ***FSL*** (29); SAS (28); ***Haploview*** (27)
Physics	***VMD*** (15); MATLAB (14); AMBER (13); ***Refmac*** (12); **ImageJ** (11); ***FSL*** (11); Stata (10); **SPM** (10); ***PHENIX*** (9); ***CHARMM*** (9)
Research and analysis methods	**ImageJ** (30); Stata (23); MATLAB (18); ***BWA*** (15); **TopHat** (13); **BLAST** (11); ***MEGA*** (10); ***MUSCLE*** (10); **SAMtools** (10); **Cufflinks** (10)
Social sciences	**SPM** (18); Stata (18); MATLAB (16); ***EEGLAB*** (8); **SAM** (8); SPSS (7); ***Talairach*** (6); ***FreeSufer*** (5); ***FSL*** (5); **REST** (4)

Note: Software that is free for academic use is marked in bold; Software, whose developers mentioned how to cite their software, is marked in italics.

Discussions and conclusions

In this article, we analyzed the use and impact of software in scientific literature across a variety of disciplines. We classified our data set of 9,548 articles published in *PLOS ONE* in 2014 into 12 disciplines. A bootstrapping method proposed in our previous work (Pan et al., 2015) was used to extracted software entities from the data set and 2,334 software entities has been learned. We examined how these software entities are used in the 12 disciplines through metrics that include the number of mentions and the number of citations.

The distribution of software entities across diverse disciplines provides evidence that software is widely used across different scientific disciplines represented in our data set. The 2,334 items of software were mentioned 25,860 times across the 12 disciplines. We found that up to 80% of the 9,548 articles contained software mentions and more than 60% of the articles in each field include at least one software mention. More data sources are needed to examine the use of scientific software to generalize our findings. Disciplinary differences in the distribution of software in scientific articles were also found. More than

80% of the articles in agriculture and medicine and health sciences contained software mentions, while only about 60% of the papers in mathematics, computer and information science, and social science contained software mentions. These findings answered the first research question on how much software is used across different scientific disciplines.

We counted the numbers of software mentions and citations in full texts using both sentences and article counting units for each discipline. A series of Mann-Whitney U-tests were used to assess the disciplinary differences for the number of software mentions and uncitedness. Evidence revealed disciplinary differences in the number of software mentions: Scientists in agriculture and biology are more likely to mention software, while scholars in social sciences and mathematics are less likely to do so. Disciplinary differences also existed in software citations: software citation is more consistently practiced in fields such as environmental sciences and computer and information sciences. In addition, the results showed that more than 30% of mentioned software received no citation in each discipline. These findings suggested that the number of software mentions in full texts should be taken into account when assessing the impact of software on science. These findings addressed the second research question on how much software is cited in scientific literature across diverse disciplines.

Last, the top 10 most mentioned and cited items of software in each discipline were identified to explore what types of software are more likely to receive citations. A statistically significant difference in the uncitedness between commercial software and noncommercial software was found. Software that is free for academic use was more likely to receive citations. We also found that the average uncitedness of software that was provided with information on how cite the software in the websites was 0.36, much lower than that of commercial software and noncommercial software without such citation guide information. These findings suggested that providing software citation targets and citation approaches can improve the practice of software citation. These findings address the third research question. In addition, this study also found that 60% of the 50 highly cited free items of software were provided with information on how to cite the software in their websites. It indicates that scientists who developed the software have

a deep interest in the popularity and impact of their products. This finding, in turn, substantiates the need to build a more inclusive scientific evaluation system that incorporates both publications and digital outputs.

One limitation of this study is that *PLOS ONE* is selected as the only data source. The mentions and uncitedness of software revealed in *PLOS ONE* are likely to be different from those of other journals, which has a higher or lower journal impact factor. A study of the use of software in biology has found that journals with higher journal impact factors are more likely to mention software and cite software formally (Howison & Bullard, 2016). There also might be differences in software citations between the journals that make a requirement of specific forms of software citations and *PLOS ONE* which did not make such requirement until 2015. Our future work includes using more data sources to demonstrate the findings of this study. Another future research direction is to exploring why scientists cite some software entities but do not cite the others.

Acknowledgments

Xuelian Pan is supported by the Program B for Outstanding PhD candidate of Nanjing University. Erjia Yan is supported by the National Consortium for Data Science (NCDS) Data Fellows program for the project "Assessing the Impact of Data and Software on Science Using Hybrid Metrics". Also, we are grateful to the reviewers for very helpful comments.

References

Candela, L., Castelli, D., Manghi, P., & Tani, A. (2015). Data journals: A survey. *Journal of the Association for Information Science and Technology*, *66*(9), 1747–1762. doi:10.1002/asi.23358

Chao, T. C. (2011). Disciplinary reach: Investigating the impact of dataset reuse in the earth sciences. *Proceedings of the ASIST Annual Meeting*, *48*. doi:10.1002/meet.2011.14504801125

Crowston, K., Howison, J., & Wiggins, A. (2010). Free/Libre Open Source Software Development: What We Know and What We Do Not Know. *ACM Computing*

Surveys, *40*(2), 1–37. doi:10.1145/2089125.2089127

Duck, G., Nenadic, G., Brass, A., Robertson, D. L., & Stevens, R. (2013). bioNerDS: exploring bioinformatics' database and software use through literature mining. BMC bioinformatics, 14(1), 1.

Hafer, L., & Kirkpatrick, A. E. (2009). Assessing open source software as a scholarly contribution. *Communications of the ACM*, *52*, 126. doi:10.1145/1610252.1610285

Hann, I.-H., Roberts, J., & Slaughter, S. (2004). Why Developers Participate in Open Source Software Projects: An Empirical Investigation. CIS 2004 Proceedings, 66.

Hannay, J. E., MacLeod, C., Singer, J., Langtangen, H. P., Pfahl, D., & Wilson, G. (2009). How do scientists develop and use scientific software? *Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering, SECSE 2009*, 1–8. doi:10.1109/SECSE.2009.5069155

Howison, J., & Bullard, J. (2016). Software in the scientific literature: Problems with seeing, finding, and using software mentioned in the biology literature. *Journal of the Association for Information Science and Technology*, 67(9), 2137–2155. doi:10.1002/asi.23538

Howison, J., Deelman, E., Mclennan, M. J., da Silva, R. F., & Herbsleb, J. D.(2015). Understanding the scientific software ecosystem and its impact : Current and future measures. *Research Evaluation*, 24(4), 454-470.

Howison, J., & Herbsleb, J. (2010). Socio-technical logics of correctness in the scientific software development ecosystem. *Workshop on Changing Dynamics of Scientific Collaboration Workshop at CSCW 2010*. Retrieved from http://repository.cmu.edu/isr/496/?utm_source=repository.cmu.edu/isr/496&utm_medium=PDF&utm_campaign=PDFCoverPages

Howison, J., & Herbsleb, J. D. (2011). Scientific software production. *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work - CSCW '11*, 513–522. doi:10.1145/1958824.1958904

Howison, J., & Herbsleb, J. D. (2013). Incentives and integration in scientific software production. *Proceedings of the 2013 Conference on Computer Supported*

Cooperative Work - CSCW '13, 459. doi:10.1145/2441776.2441828

Howison, J. and Herbsleb, J. (2014). The sustainability of scientific software : ecosystem context and science policy. Working Paper. University of Texas at Austin. Retrieved from http://james.howison.name/pubs/HowisonHerbsleb-Sustainability.pdf

Huang, X., Ding, X., Lee, C. P., Lu, T., Gu, N., & Hall, S. (2013). Meanings and Boundaries of Scientific Software Sharing. *Proc. Conf. Computer Supported Cooperative Work (CSCW)*, 423–434. doi:10.1145/2441776.2441825

Hedley, J. Jsoup: Java HTML Parser. Version 1.7.3 [software]. [cited 2015 Oct 16]. Available from: https://jsoup.org/.

IBM Corp. SPSS. Version 20 [software]. [cited 2015 Oct 16]. Available from: http://www-01.ibm.com/software/cn/analytics/spss/downloads.html.

Katz, D. S., Choi, S.-C. T., Lapp, H., Maheshwari, K., Löffler, F., Turk, M., … Venters, C. (2014). Summary of the First Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE1). *Journal of Open Research Software*, *2*(1), 1–21. doi:10.5334/jors.an

Katz, D. S., Choi, S.-C. T., Wilkins-Diehr, N., Chue Hong, N., Venters, C. C., Howison, J., … Littauer, R. (2015). Report on the Second Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE2), (1), 1–30. Retrieved from http://arxiv.org/abs/1507.01715

Lakhani, K., & Wolf, R. G. (2003). Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. *SSRN Electronic Journal*. doi:10.2139/ssrn.443040

Pan, X., Yan, E., Wang, Q., & Hua, W. (2015). Assessing the impact of software on science: A bootstrapped learning of software entities in full-text papers. *Journal of Informetrics*, *9*(4), 860–871. doi:10.1016/j.joi.2015.07.012

Piwowar, H. A. (2013). Value all research products. *Nature*, *493*, 159. doi:10.1038/493159a

Piwowar, H. A., Carlson, J. D., & Vision, T. J. (2011). Beginning to track 1000 datasets from public repositories into the published literature. *Proceedings of the ASIST Annual Meeting*, *48*(1), 1–4. doi:10.1002/meet.2011.14504801337

Piwowar, H. A., & Chapman, W. W. (2010). Public sharing of research datasets: A pilot study of associations. *Journal of Informetrics*, *4*(2), 148–156. doi:10.1016/j.joi.2009.11.010

Poisot, T. (2015). Best publishing practices to improve user confidence in scientific software. *Ideas in Ecology and Evolution*, *8*, 50–54. doi:10.4033/iee.2015.8.8.f

Prabhu, P., Zhang, Y., Ghosh, S., August, D. I., Huang, J., Beard, S., … Walker, D. (2011). A survey of the practice of computational science. *State of the Practice Reports on - SC '11*, 1. doi:10.1145/2063348.2063374

Roberts, J. A., Hann, I.-H., & Slaughter, S. A. (2006). Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects. *Management Science*, *52*(7), 984–999. doi:10.1287/mnsc.1060.0554

Robinson-García, N., Jiménez-Contreras, E., & Torres-Salinas, D. (2015). Analyzing data citation practices using the data citation index. *Journal of the Association for Information Science and Technology*, n/a–n/a. doi:10.1002/asi.23529

Rolland, B., & Lee, C. (2013). Beyond trust and reliability: reusing data in collaborative cancer epidemiology research. *Proceedings of the ACM 2013 Conference on Computer Supported Cooperative Work*, 435–444. doi:10.1145/2441776.2441826

Segal, J., & Morris, C. (2008). developing scientific software. *Software,ieee*, *25*(4), 18–20.

Tenopir, C., Allard, S., Douglass, K., Aydinoglu, A. U., Wu, L., Read, E., … Frame, M. (2011). Data sharing by scientists: practices and perceptions. *PLOS ONE*, *6*(6), e21101.

Trainer, E. H., Chaihirunkarn, C., & Herbsleb, J. D. (2013). The Big Effects of Short-term Efforts: A Catalyst for Community Engagement in Scientific Software. *Workshop in Sustainable Software for Science: Practice and Experience (WSSSPE)*, 1–4. doi:10.6084/m9.figshare.790754

Trainer, E. H., Chaihirunkarn, C., Kalyanasundaram, A., & Herbsleb, J. D. (2015). From Personal Tool to Community Resource: What's the Extra Work and Who Will Do It?. In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing(pp. 417-430). ACM.

Velden, T., Bietz, M. J., Diamant, E. I., Herbsleb, J. D., Howison, J., Ribes, D., & Steinhardt, S. B. (2014). Sharing, Re-use and Circulation of Resources in Cooperative Scientific Work. *Proceedings of the Companion Publication of the 17th {ACM} Conference on Computer Supported Cooperative Work & Social Computing*, 347–350. doi:10.1145/2556420.2558853