How important is scientific software in bioinformatics research? A comparative study between international and Chinese research communities

Bo Yang^{a,b}, Ronald Rousseau^{c,d}, Xue Wang^a, Shuiqing Huang^{a,b*}

^aDepartment of Information Management, Nanjing Agricultural University, Nanjing 210095, China
{e-mail: mail.boyang@gmail.com; 2014114011@njau.edu.cn; sqhuang@njau.edu.cn}

^bResearch Center for Correlation of Domain Knowledge, Nanjing Agricultural University, Nanjing 210095, China

^cKU Leuven, Facultair Onderzoekscentrum ECOOM, Naamsestraat 61, Leuven B-3000, Belgium,
e-mail: ronald.rousseau@kuleuven.be

^dUniversity of Antwerp (UA), Faculty of Social sciences, Middelheimlaan 1, B-2020 Antwerp, Belgium, e-mail: ronald.rousseau@uantwerpen.be

Abstract: Software programs are among the most important tools in data-driven research. The popularity of well-known packages and corresponding large numbers of citations received bear testimony of the contribution of scientific software to academic research. Yet software is not generally recognized as an academic outcome. In this study, a usage-based model is proposed with varied indicators including citations, mentions and downloads to measure the importance of scientific software. We perform an investigation on a sample of international bioinformatics research articles, and on a sample from the Chinese community. Our analysis shows that scientists in the field of bioinformatics rely heavily on scientific software: the major differences between the international community and the Chinese example being how scientific packages are mentioned in publications and the time gap between the introduction of a package and its use. Biologists publishing in international journals tend to apply the latest tools earlier; Chinese scientists publishing in Chinese tend to follow later. Further, journals with higher impact factors tend to publish articles applying the latest tools earlier.

Keywords: scientific software; citation analysis; bioinformatics; research assessment

1. Introduction

Rapid development of data-enhanced methodologies in the world of science has led researchers to rely increasingly on information technology. It is just impossible to access, analyze, and

*Corresponding author: Shuiqing Huang, sqhuang@njau.edu.cn.

visualize large data sets without support of dedicated software (Goble, 2014). According to the results of survey-based investigations, 91% scientists believe that scientific software is important to their research and 70% of 417 researchers from 15 Russell Group universities (UK) admitted that they couldn't live without scientific software (Hannay et al., 2009; Hettrick, 2014). Nowadays it is stating the obvious when saying that many software programs have a profound influence on academic research. Taking bioinformatics as an example, the article (Altschul et al., 1990) which introduced the classic protein database search program BLAST (Basic Local Alignment Search Tool), has been explicitly cited more than 46,000 times in the Web of Science (WoS) and more than 65,000 times in Google Scholar, while Camacho (2009), introducing the latest version, BLAST+, has already received more than 1600 citations in the WoS (citation data retrieved on June 6, 2017), not to mention some uncited use (Chawla, 2016). The great success of BLAST placed the corresponding paper (Altschul et al., 1990) among the top 100 most cited papers of all time in the Web of Science (Van Noorden, 2014). Before continuing this introduction, we like to mention that in this contribution the terms software program and software package will be used as synonyms. The term algorithm refers to a step-by-step procedure for solving one specific problem.

Although it is undeniable that software is important to data-driven research and a scientist's esteem is not solely depending on publications (Piwowar, 2013), the academic achievement of developers is often invisible or comes second with respect to the scientific question which the software is designed to solve (Löffler et al., 2014). The major reason for this is that software is not always recognized as a scholarly contribution (Hafer & Kirkpatrick, 2009). Moreover, a complete software program consisting of various algorithms is often a work jointly finished by many developers with various backgrounds (Hafer & Kirkpatrick, 2009). Scientific software is often used to find and understand new problems and dependency on domain knowledge makes applications of scientific software different from that of regular commercial software (Hannay et al., 2009; Segal & Morris, 2008). Proper evaluation of the influence of scientific software is vital not only to give credit to developers for their contribution to the scientific community, but also to encourage them to invest time in software maintenance (Hong et al., 2015). The importance of scientific software developers who have a combined background in statistics, mathematics, software development and domain knowledge can't be understated (Terrel, Tobis & Thiruvathukal, 2015).

Recognizing the contribution of scientific software and its developers must include giving credit to these colleagues and sustain their academic careers. This would, among others, prevent brain drain to the world outside academia (Terrel, Tobis & Thiruvathukal, 2015; Crouch et al., 2013; Katz et al., 2014). Moreover, users in a research lab focus mostly on the effectiveness of the technology in the context of their research environment rather than on the novelty of the embedded algorithm (Faniel, 2009). However, guaranteeing the reliability of scientific software might constrain intrinsic creativity because new algorithms or technologies always come with a risk of bad performance in some untested cases (Lawrence, 2006). This significantly affects reviewers' judgment of the contribution of software developers' to a research project.

The extent to which software is reused by others is not easy to measure (Katz, 2014). Yet, it may

be possible to utilize some literature-based clues to measure to what extent a computation-enhanced investigation is positively affected by specific software components. However, the lack of an appropriate approach to identify citations to software libraries has a negative influence on funding, development, and even on the scientific careers of developers (Knepley et al., 2013). Writing a paper about a software program in order to receive citations is not a universal solution (Purcell, 2013), because not all software is suitable to be summarized in traditional venues such as regular journal publications (Katz, 2014). Scientific software is not always peer reviewed and novelty requirements reduce its chance of getting published (Jackson, 2012). The current situation leads to the fact that sharing software provides a scientific software developer few benefits if there is no associated paper to be cited (Goble, 2014).

In the open source community, citing academic publications that describe software libraries and algorithms is often seen as the way in which scientific users "pay" for the use of free software (Knepley et al., 2013). Generally, an open source package can be evaluated by its download statistics (Thelwall & Kousha, 2016), number of uses as a plug-in in other packages or linkage frequencies on line (Hafer & Kirkpatrick, 2009). Applying citation and usage based strategies to evaluate the popularity and quality of software could reward software developers for their contribution to scientific research (Knepley et al., 2013). Values for such indicators can be obtained when the data is accessible in full-text or in reference databases. The standard way to measure the impact of scientific software is counting citations. Although not every software program is linked to a corresponding paper, there are some features such as name or URL of software hidden in a citing publication rather than regular references which could potentially be used to design an alternative evaluation metric.

Biology as a discipline is an excellent case for investigating the importance of scientific software in the academic world. Its subfield of bioinformatics in particular is increasingly becoming a data and computation enhanced domain, in which the technical contribution of scientific software, as well as of developers, are worth to be rewarded by users (Howison & Bullard, 2016). As IT applications depend on the stage of inquiry and experience with the use of software (Wei & Yu, 2008), biologists from different research communities rely to varying extents on different scientific software.

During the past decade the number of Chinese publications has increased sharply (Wang, 2016). Moreover, because of different local circumstances, it can be expected that the way in which Chinese scientists use and apply software is different from that at the international stage. Therefore two geographically different research communities in the field of bioinformatics will be investigated, namely the international community (including Chinese scientists publishing in English in international journals) and its Chinese counterpart (those Chinese colleagues publishing in Chinese). It is expected that interesting citation patterns and patterns related to the application of software in publications could be revealed in these two communities which are at different stages of development. In this study, we propose a usage-based model including a set of indicators such as citations, mentions and downloads to find a quantitative answer to the question: 'How important is scientific software in academic research' and in particular in bioinformatics.

2. Related work

McCall, Richards and Walters (1977) from General Electric Company defined a metric to determine a set of indicators measuring the quality of software. More recently an ISO/IEC 9126-1 derived hierarchical set of criteria was proposed to evaluate usability, sustainability, and maintainability of software (Jackson, Crouch & Baxter, 2011). These studies focus on a general evaluation frame rather than on the assessment of software from an academic perspective. For aspects of reproducibility of software evaluation and sustainability of software we refer the reader to (Tustison et al., 2013; Venters et al., 2014).

Focusing on academic applications we first note that Howison & Herbsleb (2013) found that academic credit is a powerful motivator for the production and improvement of scientific software. Yet, in general, software sharing is more costly and complicated than sharing a regular publication or even a dataset (Howison & Herbsleb, 2011). Scientific software is usually an inseparable part of the outcome of a research project; therefore industry-oriented evaluation metrics do not constitute a first option in an informetric strategy. Knepley et al. (2013) added a function `PetscCitationsRegister()` which can be used in the PETSc numerical software libraries to indicate a citation to a piece of code or an algorithm. This function plays the same role in a software program as a regular citation string in an academic publication. For most publication related use, a uniform citation format is helpful to extract citation information from a reference list of a paper and measure the impact of a given scientific package. Although there are some recommendations from publishers and software providers for citing software (Jackson, 2012), they are not uniform and moreover difficult to follow because of complicating circumstances of how software is released and used. To facilitate accessing and reusing research data bioinformaticians tend to archive their data in repositories (Pampel et al., 2013). A well-established archiving policy or platform for datasets makes identification of a reused dataset easier before any evaluating action is performed. The ODIN project, for instance, where ODIN stands for “ORCID and DataCite Interoperability Network”, built on the ORCID and DataCite initiatives, set up a model to uniquely connect scientists and datasets (Fenner et al., 2014).

Moreover, some research communities are trying to provide platforms for registration, access, integration and citing of digital resources, such as software programs, datasets, images and videos. Taking the strategies of Figshare and ZENODO as sources of inspiration (Thaney, 2013; Chawla, 2016), a digital identifier such as a DOI assigned to a software program makes it possible to cite software registered in these types of repositories (Jackson, 2012; Bressan, 2013). With a unique identifier, scientific software becomes easily citable and traceable. In a next step a credit system covering 'all' repositories could be developed to recognize developers (Katz, 2014). However, this strategy is not perfect as it cannot deal with un-cited use of datasets or software programs. As an aside we note that the same can be said for publication citations, be it to a much lesser extent. Moreover, initiatives encouraging authors and journals to follow well-established formats for citation software in their manuscripts will not work if not enforced by publishers (Mayernik, 2012). A previous investigation through content analysis of some papers in the field of biology showed that less than 50% of software mentions are in the form of formal or informal

citations (usually simply mentioning names or URLs) (Howison & Bullard, 2016). Another investigation showed that only 22% of the investigated journals had a policy for citing software (Stodden, Guo & Ma, 2013).

For the reasons mentioned above, counting software citations is at the moment not reliable at all, in particular when compared with counting regular publication citations. Yet, there are other ways to measure the impact of software in academic papers. At present, the most promising approach is collecting software mentions from the full text of papers. Pan et al. (2015) proposed a method to automatically extract mentions of software programs in full text; they found that free software tends to attract more citations. A possible reason could be that researchers see citing free software as a method to pay for their use (Knepley et al., 2013) and consider this not a necessity in case of commercial software for which their research team, department or university has paid (Howison et al., 2015).

Some measurements related to the topic of data sharing are also promising for scientific software evaluation. Fear (2013) proposed four metrics for measuring data reuse impact. Three are citation-based metrics, while the fourth uses unique downloaders (not downloads!). These citation-based metrics are: reuse count, secondary impact (a kind of g-index applied to datasets rather than scientists), and diversity (reuse across subject areas). Number of downloads as a measure for the impact of databases was proposed by Ball & Duke (2015) and a recent investigation shows that the correlation between Scopus citations and software downloads from Google code is weak (Thelwall & Kousha, 2016), but significant.

As summarized above, common use of scientific software is drawing researchers' attention to the question how scientific software is used in data-enhanced research. Some recent investigations made significant contributions to the technical aspect of distilling usage information from the texts of publications such as development of metadata schema (Howison & Bullard, 2016), package extraction (Pan et al., 2015), and usage counting and classification of scientific software (Howison & Herbsleb, 2011; Howison & Herbsleb, 2013; Pan et al., 2016). However, the reasons behind the popularity of scientific software are still vague because popularity of a package doesn't mean high quality, and the scientific value embedded in scientific software is still an open issue.

3. Methods

3.1 Sampling

As it can be expected that the use of scientific software differs depending on the research community, we will, as a first step, study one case, namely that of bioinformatics, and this on the international level and separately on the level of local Chinese investigators. Because of differences in database infrastructure related to these two communities, we designed two different sampling strategies.

Although there are several international journals which mainly focus on bioinformatics, e.g., *BMC*

Bioinformatics and *Bioinformatics*, a large fraction of the studies in these journals deal with introducing a new algorithm or improving an existing one rather than presenting original biological research. We, however, are interested in the relationship between software and original biological research, not in introducing new algorithms. It is easily seen that most biologists acknowledge the use of scientific software in their papers, so professional biology journals are better targets to understand the behavior of biologists who apply scientific software in original studies. In this IT-enhanced domain, it can be expected that the computational skills of biologists may be varying and different analysts may use different technical strategies, as well as different packages, to conduct data analyses. To obtain a representative sample for the international community, we choose three international bioinformatics-related journals with different levels of impact factors: high (*Genome Research*, IF2014 = 14.6), medium (*BMC Genomics*, IF2014 = 4.0) and low (*Gene*, IF2014 = 2.1). For the set related to the international community we randomly picked 300 articles, 100 in each journal, among those published in first issues published from 2005 to 2014.

To increase the reliability of the data analysis, a different sampling strategy (query-based) was designed for the Chinese case to cover more bioinformatics-related documents. Possibly some biases may be introduced by using different sampling strategies for these two communities. Yet, we will try to reduce this impact in some coverage-sensitive scenarios in this study. There are several highly overlapping full-text databases indexing academic journals in China (Cheng et al., 2014). Therefore, the functionally well-built database CNKI has been chosen to collect articles and corresponding data of statistical indicators for the Chinese community. The whole process is implemented in two steps:

- (1) Building a keyword list. The keyword "bioinformatics" (actually the Chinese equivalent "Shēngwù xīnxué" as CNKI is a Chinese database) combined with a topic search was used to obtain a document list from three bioinformatics-related disciplines. Then the top 20 keywords ranked by document frequency within the top five thousand documents – the most relevant ones according to the CNKI retrieval system- are determined;
- (2) Constructing a document collection. Due to the interdisciplinary characteristics of this field, a bioinformatics article can be published in any biology-related journal. This step aims to construct a document collection that covers the journals reporting the largest amount of bioinformatics studies, while ignoring some peripheral ones. Of course the most significant word among these 20 keywords is "bioinformatics" itself. Yet, we also performed searches for the other 19 keywords and made the union of all results. This search led to 5,738 different documents. Second, a list of journals ranked by document frequencies in this collection was obtained. In this way we found the top 15 journals, incorporating 1,477 articles. Reviews, conference articles, meeting abstracts, news items and articles written in English were not included. This led to the final sample for the Chinese community. The whole sampling strategy was applied in March 2015, without any limitation on publication dates. It turned out, however, that the oldest article included in our final collection was published in 1998.

3.2 Metadata Extraction

Considering the varied ways in which scientific software is mentioned in publications and the fact that there is no standard citation format like e.g. APA or Chicago style for article references, the first priority is to extract usage information from full text. Entity extraction based on pattern recognition was applied in Pan et al.'s study to calculate the frequency of software mentioned in the full text of PLOS One papers published in 2014 (Pan et al., 2015). Their result showed that the precision of the algorithm was high (higher than 0.90) but the recall was relatively low (0.42). During our study we observed that only a fraction of used software appears in the text together with some trigger words like software, package, etc. Since an automatic strategy, based on such trigger words, is not precise enough to be applied to large-scale evaluation and as it is technically challenging to extract metadata from full text, a predefined schema for scientific software mention is essential to obtain and index usage information manually in exploratory investigations. In this study, we set up a schema based on an existing one (Howison & Bullard, 2016), in which seven facets of the mentioned software in the full-text of an article will be traced. These seven facets are: name, URL, function, version, developer, affiliated country and type of use (cited or only mentioned). All the software entities and associated metadata were manually extracted from *materials* or *methods* sections to make sure that the packages introduced made some substantial contribution to studies, rather than from full text in which some unused packages are just simply mentioned. All the human involved data-extraction work was fulfilled by a group of six pre-trained volunteers composed of two master students and three senior undergraduates from the library and information science and one PHD student from Bioinformatics. Moreover, some additional data sources such as Wikipedia, Google, Baidu and Google Scholar were used to distinguish software packages from other biological items. Finally, all volunteers have been carefully tested before we started the formal extraction work: their annotation accuracy on a randomly selected test sample of 30 articles was above 90% for all volunteers.

3.3 Statistical Model

Although software is not a citable source analogous to academic articles or patents, some derived indicators, however, can be used as indirect measurement to estimate the usage of academic software and then evaluate their influence upon academic research. First, we present some basic indicators that will be used to evaluate the influence of scientific software discovered in the two article collections and discuss the correlation between each pair of indicators.

Table 1 List of 10 indicators

Indicators	Descriptions
N	Number of packages applied in a given investigation. Only packages explicitly mentioned in the <i>Materials</i> or <i>Methods</i> section are counted.
DF	Document frequency of a given package, determined as the number of papers in which a software program is mentioned; multiple mentions of the same software in the same paper are counted only once.
CitedS	Sum of the number of citations of those articles that declare the use of a given package. Citations for the international collection and Chinese collection are collected from WoS and CNKI respectively.
CitedP	CitedS/M, where M denotes the number of different articles that declare the use of a given package. This is the number of received citations per article (for articles that use a given software package).
Cited	Number of citations received by an article at the time of retrieval. The number of

	citations refers to CNKI for the Chinese community and to WoS for the international community.
$Cited_{avg}$	Annual average number of citations of an article since its inclusion in CNKI or WoS (depending on the community).
DL	Number of downloads of an article in CNKI.
DL_{avg}	Annual average number of downloads of an article since its inclusion in CNKI.
DLS	Sum of the number of downloads of all articles that declare the use of a given software program. Download data are only collected from CNKI for the Chinese case as the WoS did not provide these data.
DLP	DLS/P (where P denotes the number of different articles that declare the use of a given package). This is the number of received downloads per article (for articles that use a given packages).

The basic variables (see Table 1), are used as indicators to measure the popularity of scientific software in the formal publishing world. Yet, we need to go a step further to set up a model to analyze the relationship between software and bioinformatics research. To fulfill this task, five composite measures (Table 2) are proposed to indicate the technical quality of a research paper.

Table 2 List of five composite measures

Measures	Descriptions
DF_{soft}	Sum of document frequencies of packages mentioned (cited) in a paper.
$Cited_{soft}$	Sum of citations of packages mentioned (cited) in a paper.
DL_{soft}	Sum of download frequencies of packages mentioned (cited) in a paper.
$Cited_{savg}$	Sum of average citations of packages mentioned (cited) in a paper.
DL_{savg}	Sum of average numbers of downloads of packages mentioned (cited) in a paper.

For a given research paper i , variables are defined as follows:

$$DF_{soft}(i) = \sum_{k=1}^N DF_k \quad (1)$$

(recall that N is the number of packages applied in a given research paper, DF_k is the document frequency of the k-th package. The variables in the following formulas are defined in a similar way.)

$$Cited_{soft}(i) = \sum_{k=1}^N CitedS_k \quad (2)$$

$$DL_{soft}(i) = \sum_{k=1}^N DLS_k \quad (3)$$

$$Cited_{savg}(i) = \sum_{k=1}^N \frac{CitedS_k}{DF_k} \quad (4)$$

$$DL_{savg}(i) = \sum_{k=1}^N \frac{DLS_k}{DF_k} \quad (5)$$

4. Results

4.1 Overview

According to the data collection strategies mentioned above, we performed our case study on the 300 articles from international journals, and the 1,477 articles from Chinese journals. Software names were normalized using the editing distance (Levenshtein, 1966) with manual

assistance. Based on Table 3 we note the following characteristics of and differences between the two communities. First, we see that a large portion of studies in the two research communities depend on at least one research software program (81% and 91% respectively). Moreover, one should be aware of the fact that although a significant proportion of studies in both communities does not mention or cite any software program, this doesn't mean they didn't use software at all because acknowledging the use of software is not required by most academic journals. Relatively speaking, international studies used more different types of software (516 in 300 articles), than the Chinese studies (607 in 1,477 articles). In the international articles 42% of the mentions of software are formal citations while this is only about 8% for the Chinese community. Chinese authors tend to use the URL to acknowledge the developers' contribution rather than formal citations and the information about developer and country is seldom mentioned in the Chinese articles (see Table 3).

Table 3 Descriptive statistics

Sources	Articles		Software						
	Count	Using software	Count	Mention	Citation	URL	Version mentioned	Developer mentioned	Country mentioned
Chinese Community	1477	1196(81%)	607	6113	485(8%)	2147(35%)	1741 (29%)	28(1%)	18(0%)
International Community	300	272(91%)	516	1174	498(42%)	230(20%)	296(25%)	129(11%)	28(2%)

Next we consider Table 4 in which we show how formal citations (the column Citation in Table 3) are actually done. We make a distinction between three types of references: references to a package, references to an article in which a package is introduced or described, and other references, mainly through a URL. The most important observation is that for the international community 342 of the 499 references (68.5%) are directly to the package, while in the Chinese community only 255 of the 535 (47.7%) are given directly to the packages. Hence, Chinese tend to give more indirect references.

Table 4. Direct and indirect citation behavior related to software packages

Sources	Cited packages	References to packages			
		To packages directly	To articles	To URLs and other types	Total
Chinese Community	485	255(47.7%)	238(44.5%)	42(7.8%)	535 (100%)
International Community	498	342(68.5%)	1272(25.5%)	30(6.0%)	499(100%)

4.2 Trends of software application

Software is the most important data-related technology for bioinformatics research. As data in most research topics increase fast and this by orders of magnitude, leading to so-called big data, biologists have been increasingly keen to use scientific software to perform large-scale data analyses. Figure 1 shows the yearly increase of new software programs in scientific articles from both communities. The data for yearly new software programs are determined by the number of

first mentioned software programs- grouped by year - based on 300 articles of each community (For the Chinese community, 20 articles published during 2005-2014 are randomly selected from each of the 15 Chinese journals.). Although quite a few packages have been developed or maintained in China (Wei & Yu, 2008), most of the software packages used by the Chinese scientists were developed by their international colleagues. Consequently, numbers in Figure 1 for the Chinese community are much smaller than those for the international community. Despite the lower application of the latest software programs in the Chinese community, the application trends of new software programs are similar in the two communities. Hence, Figure 1 illustrates that the computational ability and knowledge of Chinese biologists was at first relatively limited, but the rising line implies that, concerning the use of the latest software tools, they are more and more keeping up with their international colleagues. This is also illustrated in Figure 2. Here we see that the percentage of articles that use software is only slightly increasing for the international community, but that after a fast increase during recent years, the Chinese community has now reached the same percentage (90%) as the international community.

Figure 1. New software programs appearing during the period 2005-2014.

Figure 2. Use of software in Chinese and international articles, including trend lines.

4.3 Sooner application, better science

In order to go deeper into the development of software programs in the bioinformatics research communities, some further issues will be discussed. We note that 128 individual packages appear in both collections after excluding some general packages, such as SPSS, SAS, and MATLAB. These are shown in Appendix A. For each of the 128 packages we searched for the year of first application. For the international community we searched Google Scholar and for the Chinese community we searched CNKI (restricted to Bioinformatics). In this way, we obtained two sequences of first application years of software packages, one for the international community and one for the Chinese community. Dots in Figure 3 represent the 128 packages. Their x-value is the average of the two first application years. The y-value is the actual first application year, according to CNKI or according to Google Scholar. Calculating best fitting lines for the international community and for the Chinese leads to two lines shown in Figure 3. We randomly selected some packages and searched in the respective search engines (such as Google and Google Scholar) and found that most packages were applied to biology research by international biologists soon after their release with a delay of at most two years, while it took longer to be applied by Chinese scientists. For the packages in this software list, the delay between the Chinese community and the international one was on average 5.6 years. However, the time gap between the two communities is getting narrower over the years. We consider this convergence to be an interesting and, for the Chinese community, encouraging trend.

Figure3. Trends in applying scientific software in bioinformatic research

Additionally, another interesting fact was found in the international community, namely that the time gap of applying the latest software is correlated with the impact factor of journals. Sixty packages are used in the three international journals. Setting the earliest use as time zero we compute the time gap for every package appearing in each journal. Average time gaps are: 0.75 (*Genomic Research*, IF2014 = 14.6), 2.13 (*BMC Genomics*, IF2014 = 4.0), and 3.04 (*Gene*, IF2014 = 2.1). Based on these results we argue – tentatively - that research published in higher impact journals tends to apply the latest techniques earlier.

Inspired by these clues on the journal level, a similar question could be asked on the country level to see if the time gap of applying the latest software is correlated with research strength of countries (measured by citation-based indicators). Then, the relation between time gap and some citation-based indicators on the country level are investigated in the whole international document collection (824 papers). First, we collected the country of the first author and of the corresponding author (if different); then the top 11 countries having more than 20 papers are considered in more detail. The result shows that the Spearman correlations between the time gap and several citation-based indicators (from WoS) namely total number of citations, citations per paper and number of top papers are respectively 0.65, 0.61 and 0.71.

Hence it can be argued that the negative relation between the time gap on the one hand, and visibility and usefulness of research (including research strength of countries), on the other hand, is significant: timely application of the latest tools is possibly helpful to produce better or at least more interesting research results.

4.4 To what extent does scientific software contribute to research in bioinformatics?

Although some survey-based investigations on the importance of scientific software have been performed previously, the question to what extent scientific software contributes to original research is not yet completely answered. Hence it is not possible to extract some candidate

factors which make the quality of scientific packages measurable from the perspective of their academic meaning rather than as an industrial standard (McCall et al., 1977; Jackson, Crouch & Baxter, 2011).

In order to avoid bias, we reduced the Chinese collection of articles to the same number as the international one (300). This was achieved by randomly picking 20 articles per journal from all the 15 Chinese journals.

For the international community, among the indicators in Table 1, accumulative average citations ($\text{Cited}_{\text{avg}}$) of packages applied to an investigation is highly correlated with the quality of the paper measured by total (Cited) or average ($\text{Cited}_{\text{avg}}$) numbers of citations with correlation coefficients of 0.84 and 0.80. These two correlation strengths for the Chinese collection are 0.46 and 0.50. However, some download-based indicators, such as DLS and DLP for the case of the Chinese community, are largely uncorrelated with Cited and $\text{Cited}_{\text{avg}}$. Considering the weak correlation between Scopus citations and software downloads (Thelwall & Kousha, 2016), we suggest that using downloads for the assessment of scientific software is not a reliable method.

Although the correlation coefficients between the quality of articles (Cited) and influence of scientific software (Cited or $\text{Cited}_{\text{avg}}$) for the Chinese community are not as high as those for the international community, the correlations in both cases are statistically significant and seem reasonable from an academic point of view. Additionally, the insignificance of the relations between N and Cited, or DF_{soft} and Cited in both communities corresponds with common sense that applying more packages to a research doesn't necessarily lead to higher recognition (measured in citations) and that more popular tools (measured as document frequencies) don't always produce better outcomes.

5. Conclusion

It is generally accepted that, by and large, better science receives better recognition in the scientific world. Nowadays in data-driven fields such as bioinformatics scientific software plays an important role. Yet, researchers still can't provide convincing evidence to argue that better software statistically helps to produce better science (Goble, 2014). Despite the fact that scientific software is often not a citable object as in practice it lacks a DOI or some other identifier (Jackson, 2012; Bressan, 2013) indirect information from citable publications is valuable to evaluate scientific software and their makers. In this study, a model composed of a dozen indicators measuring the use of a scientific package is proposed and, based on this model, the relation between quality of scientific software and research papers is discussed from different aspects based on data collected from work by the international bioinformatics community. Moreover, the regional Chinese community was set up as a parallel case to test the reliability of the model. Our analytical results show that obviously scientists from both communities rely heavily on scientific software, the major difference between the two communities being the extent to which scientific software is valued and how scientific packages are mentioned in publications.

In Howison and Bullard's study (2016), 44% of software mentions can be linked to an entry in references list, which is very close to the corresponding proportion (42%) for the international community in this study. However, the collection in Howison and Bullard's study includes articles from comprehensive journals, such as *Nature* and *Science* and some of them may not focus on the technical aspect of biological research. Hence we are not surprised that in their study more articles didn't mention any software packages compared to ours (35% versus 10% for the international community). Although we didn't compare the percentage of software mentions with the impact factors of journals, we can see from the data in Table 1 that scientists from more developed research communities (the international group) are more likely to declare the use of software. A similar conclusion can be drawn from Howison and Bullard's (2016) study where it is stated that articles from the journals in higher strata show more software mentions.

The constant flow of new scientific packages justifies the hypothesis that biologists are becoming more aware of the quality of scientific software. This trend is particularly clear for Chinese biologists. The underdevelopment (in the past) in Chinese bioinformatics is confirmed by the time gap in the Chinese community between release of software and their application. Additionally, we found that publications in international journals with higher impact factor and work performed by biologists from more developed countries tend to apply the latest tools earlier. Possible reasons behind earlier application of the latest tools may be varying, such as greater awareness of latest techniques and better access (technically or financially) to cutting edge software; some advanced research teams may even hold the ownership of new developed tools and hence are their first users. In our study we considered the importance of applying the latest techniques to bioinformatics research through journals and countries.

Many scientists are inclined to trust scientific software that is used in a peer-reviewed article, rather than published on other platforms (Joppa et al., 2013). Therefore the need of evaluating the influence of scientific software in articles is of great interest, while the premature technique of entity extraction makes large-scale processing difficult.

Different from regular publications, a scientific program is not a fixed unit which can be used in evaluation exercises. Indeed, it can be distributed as an independent product or embedded into another package (Hafer & Kirkpatrick, 2009). Moreover, some programs have more than one version with different numbers or names and sometimes the boundary between scientific software working together with a data source and a pure database is vague. Though some rules are applied in this investigation, more precise definitions and enhanced practices are necessary to solve these issues conceptually as well as technically.

Scientists tend to cite important papers (either positively or negatively) and ignore the trivial ones (Garfield, 1979). Yet, citation counts are just an approximate indicator for the influence of the intellectual content of papers (Crane, 1972; Cole, 1973). These remarks also apply to the citation of software packages. When software packages are used they form an inseparable part of the intellectual content and quality of a paper. Therefore it is expected that the usefulness and the intrinsic quality of scientific packages can at least to some extent be gauged by citation-based

indicators.

Among the indicators for measuring the influence of scientific software introduced in this study the first priority is to determine which indicator would be the most promising one to evaluate the quality of scientific software. Based on our statistics the quality of a scientific package measured by the average number of citations of a paper is highly correlated to its total or average citation count for the case of the international community. Despite lower correlation values this indicator is also significant for the Chinese community. The reason is that in the past the computing infrastructure in China was rather limited and most labs didn't have professional technical staff (Wei & Yu, 2008). Nevertheless, our findings shine a light on the measurement of the contribution of scientific software to research. We conclude with some confidence that in bioinformatics better scientific software helps produce better science.

Although the correlation analysis between the earliest application and total citations of 128 packages suggests to reject the hypothesis that earlier released packages attract more citations, we didn't perform a similar investigation for the indicator DLS because download data in WoS was not available. Additionally, considering the different data collection strategies applied to reach a compromise between the two communities, the subject coverage of the sample set for the Chinese community (15 journals) is broader than that of the international counterpart (3 journals), some unexpected biases may be introduced on some of our findings and another sampling strategy would be helpful to analyze different communities in future investigations.

Acknowledgement: The authors thank Dr. Jinding Liu and Dongqing Yang for their highly appreciated suggestions as domain experts and five volunteers from the Department of Information Management at Nanjing Agricultural University for their work on metadata extraction. This work is supported by the National Social Science Fund of China under Grant No. 13CTQ031 and the fund of Creative Studies of Nanjing Agricultural University under Grant No. SKCX2016005.

References

- Altschul, S.F., Gish, W., Miller, W., Myers, E.W., & Lipman, D.J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215(3), 403-410.
- Bressan, B. (2013). The SciencePAD treasure hunt of persistent identifiers. *CERN Bulletin*, BUL-NA-2013-007.
<http://cds.cern.ch/journal/CERNBulletin/2013/05/News%20Articles/1505243>. [February 18th, 2016]
- Camacho, C., Coulouris, G., Avagyan, V. , Ma, N., Papadopoulos, J., Bealer, K., & Madden, T.L. (2009). BLAST+: architecture and applications. *BMC Bioinformatics*, 10, 421.
- Chawla, D.S. (2016). The unsung heroes of scientific software. *Nature*. 529(7),115-116.
- Cheng, W.H., Ren, S.L., & Rousseau, R. (2014). Digital publishing and China's core scientific journals: a position paper. *Scientometrics*, 98(1), 11–22.

- Cole, J.R., & Cole, S. (1973). *Social Stratification in Science*. Chicago, IL: University of Chicago Press.
- Crane, D. (1972). *Invisible colleges; diffusion of knowledge in scientific communities*. Chicago: University of Chicago Press.
- Crouch, S., Hong, N. C., Hettrick, S., Jackson, M., Pawlik, A., Sufi, S., . . . Parsons, M. (2013). The software sustainability institute: Changing research software attitudes and practices. *Computing in Science and Engineering*, 15(6), 74-80.
- Faniel, I.M. (2009). *Unrealized Potential: The Socio-Technical Challenges of Large Scale Cyberinfrastructure Initiative*. University of Michigan, School of Information, Ann Arbor, USA.
- Fear, K.M. (2013). *Measuring and anticipating the impact of data reuse*. Doctoral dissertation. University of Michigan, Michigan.
- Fenner, M., Haak, L.L., Thorisson, G.A., Ruiz, S., Vision, T.J., & Brase, J. (2014). ODIN: the ORCID and DataCite interoperability network. *International Journal of Knowledge and Learning*, 9(4), 305-325.
- Garfield, E. (1979). Is citation analysis a legitimate evaluation tool?. *Scientometrics*, 1(4), 359-375.
- Goble, C. (2014). Better software, better research. *IEEE Internet Computing*, 18(5), 4-8.
- Hafer, L., & Kirkpatrick, A.E. (2009). Assessing open source software as a Scholarly contribution. *Communications of the ACM*, 52(12), 126-129.
- Hannay, J., 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 '09*, Washington, DC, USA, IEEE Computer Society, 1-8.
- Hettrick, S. (2014). It's impossible to conduct research without software, say 7 out of 10 UK researchers.
<http://www.software.ac.uk/blog/2014-12-04-its-impossible-conduct-research-without-software-say-7-out-10-uk-researchers>. [March 3rd, 2016]
- Hong, N.C., Antonioletti, M., Carr, L. et al. (2015). Better software, better research: Providing scalable support for scientific software development. *Figshare*.
<https://dx.doi.org/10.6084/m9.figshare.1564720.v1>.
- 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.
- Howison, J., Deelman, E., McLennan, M. J. et al. (2015). Understanding the scientific software ecosystem and its impact: Current and future measures. *Research Evaluation*, rvv014:1-17.
- Howison, J., & Herbsleb, J. D. (2011). Scientific software production: incentives and collaboration. In *Proceedings of the ACM 2011 conference on Computer Supported Cooperative Work*, 513-522.
- Howison, J., & Herbsleb, J.D. (2013). Incentives and integration in scientific software production. In: *Proceedings of the 2013 conference on Computer supported cooperative work*, 459-470.
- Jackson, M. (2012). How to cite and describe software.
<http://www.software.ac.uk/how-cite-and-describe-software?> [March 4th, 2016]
- Jackson, M., Crouch, S., & Baxter, R. (2011). *Software evaluation: criteria-based assessment*. Software Sustainability Institute, The University of Edinburgh,

- <https://www.software.ac.uk/sites/default/files/SSI-SoftwareEvaluationCriteria.pdf>. [March. 15th, 2016] Ketan Maheshwari, Frank Löffler, Matthew Turk, Joppa, L.N., McInerny, G., Harper, R., Salido, L., Takeda, K., O'Hara, K., . . . Emmott, S. (2013). Troubling trends in scientific software use. *Science*, 340(6134), 814-815.
- Katz, D.S. (2014). Transitive credit as a means to address social and technological concerns stemming from citation and attribution of digital products. *Journal of Open Research Software*, 2(1):e20.
- 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). <http://openresearchsoftware.metajnl.com/articles/10.5334/jors.an/>. [June 6th, 2017]
- Knepley, M.G., Brown, J., McInnes, L.C., & Smith, B. (2013). Accurately citing software and algorithms used in publications. Workshop towards Sustainable Software for Science: Practice and Experiences, Denver, CO. http://www.mcs.anl.gov/papers/P5010-0913_1.pdf. [February 28, 2016]
- Lawrence, K.A. (2006). Walking the tightrope: The balancing acts of a large e-research project. *Computer Supported Cooperative Work (CSCW)*, 15(4), 385-411.
- Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10, 707-710.
- Löffler, F., Brandt, S.R., Allen, G., & Schnetter, E. (2014). Cactus: Issues for sustainable simulation software. *Journal of Open Research Software*, 2(1), e12.
- Mayernik, M.S. (2012). Data citation initiatives and issues. *Bulletin of the American Society for Information Science and Technology*, 38(5), 23-28.
- McCall, J.A., Richards, P.K., & Walters, G.F. (1977). Factors in software quality: Concept and Definitions of Software Quality. Volume I. GENERAL ELECTRIC CO SUNNYVALE CA.
- Pampel, H., Vierkant, P., Scholze, F., Bertelmann, R., Kindling, M., Klump, J., . . . Dierolf, U. (2013). Making research data repositories visible: the re3data.org registry. *PloS One*, 8(11), e78080.
- Pan, X.L., Yan, E., Wang, Q.Q., & Hua, W.N. (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.
- Pan, X.L., Yan, E., & Hua, W.N. (2016). Disciplinary differences of software use and impact in scientific literature. *Scientometrics*, 109(3), 1593-1610.
- Purcell, A. (2013). Standing on the shoulders of software developers. <http://home.cern/about/updates/2013/01/standing-shoulders-software-developers>. [February 18th, 2016]
- Segal, J., & Morris, C. (2008). Developing scientific software. *IEEE Software*, 25(4), 18-20.
- Stodden, V., Guo, P., & Ma, Z. (2013). Toward reproducible computational research: an empirical analysis of data and code policy adoption by journals. *PloS One*, 8(6), e67111.
- Terrel, A., Tobis, M., & Thiruvathukal, G.K. (2015). Scientific software communities. *Computing in Science & Engineering*, 17(8), 8-10.
- Thaney, K. (2013). Code as a research object: a new project. <https://mozillascience.org/code-as-a-research-object-a-new-project>. [Mar.1st, 2016]
- Thelwall, M., & Kousha, K. (2016). Academic Software Downloads from Google Code: Useful Usage Indicators?. *Information Research: An International Electronic Journal*, 21(1), n1.

- Tustison, N.J., Johnson, H.J., Rohlfing, T., Klein, A., Ghosh, S.S., Ibanez, L., & Avants, B.B. (2013). Instrumentation bias in the use and evaluation of scientific software: recommendations for reproducible practices in the computational sciences. *Frontiers in Neuroscience*, 7, 162.
- Van Noorden, R.V., Maher, B., & Nuzzo, R. (2014). The top 100 papers. *Nature*, 514(30), 550-553.
- Venters, C.C., Lau, L., Griffiths, M.K., Holmes, V., Ward, R.R., Jay, C., . . . Xu, J. (2014). The blind men and the elephant: Towards an empirical evaluation framework for software sustainability. *Journal of Open Research Software*. 2(1), e8.
- Wang, L. (2016). The structure and comparative advantages of China's scientific research: Quantitative and qualitative perspectives. *Scientometrics*, 106(1), 435-452.
- Wei, L., & Yu, J. (2008). Bioinformatics in China: A personal perspective. *PLoS Computational Biology*, 4(4), e1000020.

Appendix

Appendix A. Packages used by the two communities (in alphabetical order)

Affymetrix GCOS	Gene Structure Display Server(GSDS)	Motif scan
AgBase	Genemapper	MrBayes
AmiGO	GeneMark	MUSCLE
Arlequin	GeneSpring	NetNGlyc/NetOGlyc
Augustus	GeneTool	NetPhos
BeadStudio	Genetyx	Oligo
BiNGO	Genomatix	ORF Finder
BioEdit	GenomeStudio	PAL2NAL
Biomart	GeNorm	PAML
Blast	genscan	Pathway Studio
Bowtie	GOEAST	PAUP
CAP3	Gorilla	Phrap
CDART	Graphpad prism	phylip
Chromas	GSEA	PHYML
Circos	HMMER	PLACE
Clone Manager	HMMTOP	Primer Premier
Clustal	Image J	PrimerBLAST
Clustal Omega	Interpro	Protfun
Cluster	InterPro Scan	ProtParam
CodonW	Jalview	PSIPRED
Conserved Domain Search (CDS)	JoinMap	PSORT
ContigExpress	Jpred	Pymol
Crossmatch	MacClade	Quantity ONE
cufflinks	MACS	Rasmol
cytoscape	MAFFT	RepeatMasker
DAMBE	MapChart	RNAfold
DAVID	MapDraw	Scanprosite
DNA Sequence Polymorphism/DnaSP	Mascot	SeqMan
DNAMAN	MEGA	SEQUEST
DNAssist	MEGABLAST	
DNAstar	MegAlign	SIGNALP
DNAStrider	MEME	SMART
Editseq	Methprimer	SOAPdenovo
EMBOSS	Mfold	SOSUI
FASTA	miRanda	Spidey
FASTX-Toolkit	Mireap	SSRHunter
FGENESH	MISA	TargetP
Gene Ontology	Mobyle	TargetScan
GENE RUNNER	Modeller	TatP

TESS(Transcription Element Search System)	TopHat	velvet
	Tree puzzle	Vienna RNA
TFSEARCH	Treeview	WebLogo
TMHMM	UCSF Chimera	
Tmpred	Vector NTI	

Appendix B. The top 50 (actually 53 because of ties) packages ranked by document frequencies for the international community.

Package Name	DF	CitedS	CitedP
BLAST	127	6632	52
ClustalW	48	1302	27
UCSC Genome Browser	47	3416	73
RepeatMasker	22	1229	56
MEGA	21	459	22
R language	21	321	15
Primer Premier	15	276	18
PAML	14	1053	75
BLAT	13	586	45
Microsoft Excel	11	172	16
Significance Analysis of Microarrays (SAM)	11	465	42
PhyML	10	108	11
MrBayes	9	197	22
SPSS	9	166	18
SIGNALP	8	319	40
TopHat	8	280	35
TreeView	8	222	28
BioEdit	7	178	25
Blast2GO	7	99	14
Bowtie	7	306	44
Cufflinks	7	200	29
EMBOSS	7	478	68
HMMer	7	161	23
MEGABLAST	7	336	48
RNAfold	7	890	127
SAS	7	163	23
MUMmer	6	132	22
MUSCLE	6	923	154
PHRED	6	310	52
tRNAscan-SE	6	82	14
DnaSP	5	29	6
ImageJ	5	98	20
InterProScan	5	104	21

Matlab	5	228	46
mfold	5	104	21
PAUP	5	240	48
RPS-BLAST	5	260	52
Weblogo	5	50	10
Artemis	4	694	174
Artemis Comparison Tool	4	626	157
exonerate	4	280	70
Genespring	4	274	69
Glimmer	4	666	167
GraphPad Prism	4	63	16
HaploView	4	57	14
MySQL	4	121	30
NetNGlyc	4	10	3
Newbler	4	78	20
PHAST	4	347	87
PHYLP	4	78	20
Sequencher	4	172	43
SMART	4	59	15
TMHMM	4	59	15

Appendix C. The top 50 packages ranked by document frequencies for the Chinese community.

Package Name	DF	CitedS	DLS	CitedP	DLP
BLAST	514	1446	66278	2.8	128.9
MEGA	366	741	39238	2.0	107.2
ProtParam	322	643	33493	2.0	104.0
SianalP	314	749	34590	2.4	110.2
CluatalX	292	794	38343	2.7	131.3
DNA MAN	285	685	30676	2.4	107.6
DNA star	267	832	31381	3.1	117.5
Primer Premier	231	455	24330	2.0	105.3
TMHMM	225	494	24768	2.2	110.1
ProtScale	200	507	21539	2.5	107.7
Swiss Model	161	263	16982	1.6	105.5
ORF Finder	155	431	19516	2.8	125.9
PSORT	112	242	12510	2.2	111.7
SMART	96	184	9417	1.9	98.1
BioEdit	92	241	11811	2.6	128.4
NetPhos	83	182	7892	2.2	95.1
Tmpred	83	180	7935	2.2	95.6
SOPMA	79	191	8977	2.4	113.6
ExPASy	77	152	8206	2.0	106.6

PREDICT PROTEIN	74	164	7235	2.2	97.8
TargetP	50	135	6178	2.7	123.6
CDD	45	111	5276	2.5	117.2
Scanprosite	45	86	5187	1.9	115.3
Conserved DomainS	43	85	4211	2.0	97.9
Oligo	43	88	4350	2.0	101.2
Prosite	41	105	3740	2.6	91.2
InterPro Scan	40	99	4167	2.5	104.2
Vector NII	36	98	4972	2.7	138.1
Protfun	31	78	3364	2.5	108.5
Wolf Psort	31	58	3606	1.9	116.3
genscan	31	126	4627	4.1	149.3
Pfam	27	41	3004	1.5	111.3
SPSS	26	59	2651	2.3	102.0
NetGlyc	25	55	2448	2.2	97.9
Motif scan	24	44	1985	1.8	82.7
PSIPRED	24	21	1532	0.9	63.8
Plant CARE	23	52	2134	2.3	92.8
ProtComp	23	29	2634	1.3	114.5
Anthprot	21	73	2913	3.5	138.7
GOR	21	58	2331	2.8	111.0
MEME	21	28	2244	1.3	106.9
COILS	20	93	2897	4.7	144.9
Compute pI/MW	20	36	1963	1.8	98.2
PLACE	19	44	2930	2.3	154.2
Genedoc	18	56	2064	3.1	114.7
MegAlign	18	22	1810	1.2	100.6
phylip	17	67	3525	3.9	207.4
CAP3	15	55	1968	3.7	131.2
Swiss Pbdviewer	14	59	1949	4.2	139.2
DNA SP	13	38	1695	2.9	130.4