Journal of Informetrics 12 (2018) 87–100

Contents lists available at ScienceDirect

Journal of Informetrics

journal homepage: www.elsevier.com/locate/joi

Co-mention network of R packages: Scientific impact and clustering structure

Kai Li*, Erjia Yan

College of Computing and Informatics, Drexel University, Philadelphia, PA 19104, United States

ARTICLE INFO

Article history:
Received 8 November 2017
Received in revised form
30 November 2017
Accepted 1 December 2017
Available online 5 December 2017

Keywords:
R
Open software
Scientometrics
Network analysis
Co-mention analysis

ABSTRACT

Despite its rising position as a first-class research object, scientific software remains a marginal object in studies of scholarly communication. This study aims to fill the gap by examining the co-mention network of R packages across all Public Library of Science (PLoS) journals. To that end, we developed a software entity extraction method and identified 14,310 instances of R packages across the 13,684 PLoS journal papers mentioning or citing R. A paper-level co-mention network of these packages was visualized and analyzed using three major centrality measures: degree centrality, betweenness centrality, and PageRank. We analyzed the distributive patterns of R packages in all PLoS papers, identified the top packages mentioned in these papers, and examined the clustering structure of the network. Specifically, we found that the discipline and function of the packages can partly explain the largest clusters. The present study offers the first large-scale analysis of R packages' extensive use in scientific research. As such, it lays the foundation for future explorations of various roles played by software packages in the scientific enterprise.

1. Introduction

Software is a central player in contemporary data-driven scientific activities, because "data doesn't do anything of itself" (Berry, 2016, p. 51). Software plays important, yet often invisible or overlooked, roles—not only in scientific research, but also our daily decision-making. These roles are a central topic in the field of software studies, which aims at recognizing the "cultural and social forces that are shaping the development of software itself" (Manovich, 2013, p. 11). Findings from this area of research have affirmed that software has significant relevance to how data is included, manipulated, and evaluated in scientific studies, as well as what data is available to researchers (Driscoll & Walker, 2014; Gillespie, 2014; Ruhleder, 1994, 1995).

Until recently, however, research in the field of information science has not given software this same degree of attention. Studies to date have revealed that software entities are heavily mentioned in scientific publications but inconsistently cited (e.g., Howison & Bullard, 2015; Li, Yan, & Feng, 2017; Pan, Yan, Wang, & Hua, 2015). Nonetheless, we are still unable to answer many questions about the relationship between software and scientific research, questions whose scope and complexity extends well beyond bulk citation frequency. One such question will be addressed by the present study: how do software entities, or the mention of such entities, co-occur in scientific outputs?

Network analysis offers promising tools for approaching such a problem. In information science, network analysis methods have already been used to trace the intellectual structure of a given field, based on co-citation relationships between

* Corresponding author.
E-mail address: kl696@drexel.edu (K. Li).

https://doi.org/10.1016/j.joi.2017.12.001
1751-1577/

scientific papers and authors. Such work has enhanced our understanding of the relationships between these entities in a variety of contexts (e.g., Chen, 1999; Nerur, Rasheed, & Natarajan, 2008; White & McCain, 1998; Yan & Ding, 2009). However, as a new type of research object in information science, scientific software entities have not yet been examined from this angle. It is our goal to build a framework to quantitatively study the roles of software in scientific research. This framework takes software entities as the unit of analysis, employs bibliometric relationships such as co-occurrence and citation context as the research instrument, and adopts network-based indicators as the method to broaden a research frontier that is currently dominated by the analysis of scholarly units such as authors, papers, and journals.

With this vision in mind, this study uses network analysis tools in order to understand how R packages have been co-mentioned by scientific papers. R, like some other open source software products (e.g., Python), is built upon the idea of a software ecosystem (German, Adams, & Hassan, 2013). In such an ecosystem, the core software forms a platform, which can be supplemented by extensions developed by various stakeholders. In the case of R, its core statistical-computing and graphics functionality has been extended by packages developed by third parties. It is these packages that directly facilitate countless scientific tasks, making R applicable to—and popular within—many knowledge domains (e.g., Boettiger, Chamberlain, Hart, & Ram, 2015; Gentleman et al., 2004; Marwick, 2016; Mens, Claes, & Grosjean, 2014; Tippmann, 2014). These packages can be used, individually or collectively, in scientific research and then inscribed in the resultant papers. Thus, the co-mention network formed by R packages is a strong indication of their relationship to scientific research as well as their relationship to other R packages within research contexts.

Extending our previous study (Li et al., 2017), this paper conducts a large-scale analysis on the involvement of R packages in scientific texts. Specifically, we answer the following three questions:

- What is the scientific impact of the R packages mentioned or cited in papers from the Public Library of Science (PLoS)?
- What is the clustering structure of the co-mention network of R packages?
- What does the clustering structure mean in the context of scholarly communication?

To answer these question, we employ an entity extraction algorithm to automatically identify all R packages in the sampled PLoS papers. Based on the identification results, we plot the co-mention network, use centrality measures to understand the attributes of the network, and explore the clustering structure of the network. This is the first extensive study to examine the distribution of software entities in scientific publications using network analysis. As such, it serves as a cornerstone for a deeper understanding of the relationships among scientific software, scientific activities, and scientific texts.

2. Literature review

2.1. Software as a research object in studies of scholarly communication

During the past two decades, the scientific enterprise has been shifting towards the data-driven scientific paradigm (Hey, Tansley, Tolle, & et al., 2009). In this new paradigm, data and software are becoming increasingly important vehicles for producing and reproducing scientific knowledge. Researchers of scholarly communication have also started to pay closer attention to software as a research object. For example, Ding and colleagues (Ding et al., 2013) have proposed the concept of "entitymetrics" to systematically incorporate digital research objects into the research agenda of scientometrics. Under the influence of these broader trends, we are also witnessing more empirical studies pursuing questions related to scientific software.

A major topic under this umbrella is the inconsistency of software citation, an issue to which Ince, Hatton, & Graham-Cumming (2012) have called attention. After examining 90 biological articles, Howison and Bullard (2015) concluded that many software entities are cited with insufficient information to support the functions of citation. Li, Greenberg, & Lin (2016) found that the citation of LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) fails to reveal much meaningful information about how the software is used in the reported studies. Similarly, Li et al. (2017) identified highly variant practices in the citation of R software and packages, from official citation to unofficial mentioning. All these studies make it urgent to develop a framework to properly cite software entities in research outputs, a need recently recognized by the FORCE11 Software Citation Working Group (Katz & Smith, 2015; Niemeyer, Smith, & Katz, 2016; Smith, Katz, & Niemeyer, 2016).

Researchers have also aimed to understand the scientific impact of software in scientometric terms. For example, a group of researchers (Pia, Basaglia, Bell, & Dressendorfer, 2009; Pia, Basaglia, Bell, & Dressendorfer, 2010; Pia, Basaglia, Bell, & Dressendorfer, 2012) has studied the scientific impact of software entities implementing different versions of the Monte Carlo method. The authors confirmed the importance of the software represented by the papers they analyzed and demonstrated that these tools are used across-disciplines. More recently, Pan and colleagues (Pan, Yan, & Hua, 2016; Pan et al., 2015) have worked to understand the scientific significance of software in both general and disciplinary contexts by using a bootstrapping method to extract software entities. They listed the most highly cited software based on papers published in PLoS ONE in 2014 and concluded that open software is more likely to be mentioned than its proprietary counterparts. Another avenue of research is devoted to assessing the impact of software using altmetric concepts (Piwowar & Priem, 2016; Priem & Piwowar, 2013; Zhao & Wei, 2017). These papers have extended our understanding of the impact

of scientific software by using a combination of altmetrics indicators (e.g., number of mentions in social media and number of downloads). Together, these studies epitomize the growing importance of software in scholarly communication studies, forming a vital background for the current work.

2.2. R as a research object

R has become an indispensable application for data science because of its ability to support a full spectrum of tasks in the data-driven paradigm of decision making. R's success is represented by its rising position in programming language indices, such as Muenchen's ranking of major data analysis software (Muenchen, 2012), TIOBE Index,[1] and RedMonk Programming Language Rankings.[2] The growing list of publications offering advice on R's use in data science provides further evidence of the software popularity (e.g., Baumer, Kaplan, & Horton, 2017; Pathak, 2017; Wickham, 2016; Zumel & Mount, 2016). Moreover, R has gained widespread adoption among researchers in various disciplines (Tippmann, 2014), notably including bioinformatics (Gentleman et al., 2004), ecology (Boettiger et al., 2015; Mens et al., 2014), and archaeology (Marwick, 2016).

R's status as a full-fledged software ecosystem is a major factor contributing to its success (German et al., 2013). In this ecosystem, anyone can build new components to extend the core functionalities of the software. These building blocks, the most basic functional modules in the R ecosystem, are called packages. They are composed of multimedia resources, such as code, data, and documentation (Wickham, 2015, p. 3), which are bundled together to streamline the process of finishing certain tasks. R packages are normally deposited at online repositories, so that they can be found and reused by others. The most famous of these repositories include CRAN (the official R package repository), Bioconductor, and R-Forge; Github, a comprehensive code-sharing website, is also heavily used by programmers to share R packages and code (Decan, Mens, Claes, & Grosjean, 2015; Gentleman et al., 2004).

Despite existing efforts to study the R ecosystem from the perspective of software development and distribution (Claes, Mens, & Grosjean, 2014; Decan et al., 2015; German et al., 2013; Mens et al., 2014), few studies have examined the roles played by R packages in scholarly communication. As we are gaining more knowledge about how R software entities are cited in research outputs (Li et al., 2017), this study is designed to better understand another aspect of the relationship between the R ecosystem and research outputs: the co-mention patterns of R packages in scientific papers.

2.3. Co-citation analysis

Co-citation analysis is an important research method in scientometric studies. Since its development in the early 1970s, it has been seen as a valuable method in that it is able to reveal inter-document relationships that are established by the citing authors (Small, 1973). From a quantitative point of view, it is commonly assumed that the strength of the link between two documents is a positive indicator of the similarity of these documents (Gmür, 2003), even though which measurement to use is not an unquestionable issue (Ahlgren, Jarneving, & Rousseau, 2003; McCain, 1990). Another notable feature of co-citation analysis is its ability to transform co-citation relationships into two-dimensional networks, or a specialty map of a given knowledge domain (Chen, 1999; Chen & Paul, 2001; Small & Crane, 1979; White & McCain, 1998). Different approaches to the interpretation of co-citation networks have been developed, from using the similarity or relationship between entities (Mombers, Van Heeringen, Van Venetië, & Le Pair, 1985; Small, Sweeney, & Greenlee, 1985), to the development stage of such specialties (Bellardo, 1980).

Like other scientometric methods, co-citation analysis can be conducted on multiple levels of entities, such as document, journal, and even country of authorship. Author co-citation analysis (ACA) is the most popular example of this research method. First introduced by White & Griffith (1981), ACA coordinates document-based statistics with a selected list of researchers, with the goal of identifying the most influential authors in a field and the relationships among them. In their famous work on visualizing these relationships in information science, White & McCain (1998) selected the top 120 authors from the DIALOG system, analyzed their specialties based on a clustering analysis of co-citation relationships, and studied how those relationships have changed over time.

These author-level analyses, along with studies on the journal level (e.g., Ding, Chowdhury, & Foo, 2000; Hu, Hu, Gao, & Zhang, 2011; Liu, 2005; Tsay, Xu, & Wu, 2003) and the country level (Hou & Chen, 2011), show the productivity and applicability of co-citation analysis. Use of this method has deepened our understanding of the relationship between many scholarly objects. However, as an emerging research object in scholarly communication, software entities have not yet benefited from co-citation analysis—hence the present study.

3. Methods

3.1. Data collection

Our approach to data collection resembles that described in our previous study (Li et al., 2017). We selected PLoS as our data source and used its public API to search for papers that cited or mentioned the software R. PLoS was selected because, as an open-access database, it permits the retrieval and analysis of full-text data.

[1] https://www.tiobe.com/tiobe-index/.
[2] http://redmonk.com/sogrady/category/programming-languages/.

The query was itself conducted using an R package: "rplos" (Chamberlain, Boettiger, & Ram, 2016). We adapted the query used last time by including some additional terms uniquely connected to R. The following query was used in the present study:

> 'everything:"A Language and Environment for Statistical Computing" OR everything:"www.r-project.org" OR everything:"cran.r-project.org" OR everything:"RStudio" OR everything:"R Foundation for Statistical Computing"'

We considered but discarded some other search terms, such as "R software" and "R packages", because they bring too many irrelevant results. Our search was performed in August 8, 2017. With only full papers selected, we acquired 13,684 records of papers from PLoS.[3] After the metadata of all papers was retrieved, their full texts were also downloaded for the next step of our analysis.

3.2. Software entity identification

An automatic name-entity recognition algorithm was designed and implemented using the following steps.

In the first step, we took the names of all R packages from its two most important package repositories, CRAN[4] and Bioconductor.[5] As of August 8, 2017, there are 11,203 packages indexed by CRAN and 1381 packages by Bioconductor. No package was included in both repositories. The list of all 12,584 packages was later used as a dictionary to match package entities in the full paper texts.

In the second step, we developed an algorithm based on the package-name dictionary in order to identify all the packages mentioned in PLoS papers. We first randomly selected and analyzed four groups of 30 papers. Each group of papers was manually coded to identify linguistic patterns where an R package is mentioned. These identified rules were implemented in a piece of self-developed code in R, then applied to each group of papers to test the accuracy rates using an iterative approach. After the introduction of the fourth group, the accuracy rates remained stabilized (precision = 95%; recall = 93.5%). We therefore considered our algorithm to be finished.

Our algorithm comprises the following procedures:

1. Full-text data is preprocessed:
   a Common English stop words are removed based on the dictionary built into the "tm" package of R.
   b Every word is transformed into lowercase.
   c Shortened forms of "version", such as "v." and "ver.", are restored to the full spelling.
2. Only those sentences meeting all the requirements below are tokenized and analyzed:
   a The sentence has more than four words;
   b The sentence contains at least one of the following words: "software", "www.r", "project.org", "cran.r", "bioconductor", "project.org", "r", "cran", "library", "libraries", "package", "packages", "library", and "libraries"; and,
   c The sentence contains at least one of the package names collected from CRAN and Bioconductor.
3. Each word from the selected sentences is graded according to the following guideline. For each item in the guideline, a score of 1 is given if the criterion is met; 0 otherwise:
   a The word is in the package name list (**V1**);
   b There is an instance of "package" or "library" in the position immediately before or after the word (**V2**);
   c There is an instance of "packages" and "libraries" in position −8 to −1 before the target word, and an instance of a package name in position −4 to −1 before the same word (**V3**);
   d There is an instance of "packages" and "libraries" in position +1 to +8 after the target word, and an instance of a package name in position +1 to +4 after the same word (**V4**);
   e There is an instance of one of the following terms ("package", "packages", "library", "libraries") in position −2 to +2 around the target word and an instance of one of the following terms ("software", "www.r", "project.org", "cran.r", "bioconductor", "project.org", "r", "cran", "library", "libraries") immediately adjacent to the word (**V5**).

For each word, a final score is calculated using the following formula:

- score = (V2 + V3 + V4 + V5) * V1

If the final score is larger than 0, then the target word is determined to be an R package. The scoring of an exemplary sentence is attached as the Appendix. Thus, the packages identified in our study are not necessarily cited by the papers:

[3] As reported in our previous paper (Li et al., 2017), it is difficult to collect all papers about R using normal queries. Following a similar testing method by using the query of *'everything:ŕ softwareÖR everything:ŕ package¨*, we estimated that there are about 12,000 papers mentioning R so unofficially that they cannot be retrieved using any meaningful combination of search terms. This bias towards more official mentions of R is inevitable, but it does not affect the validity of our results in this study.

[4] https://cran.r-project.org/web/packages/available_packages_by_name.html.

[5] https://www.bioconductor.org/packages/release/bioc/.

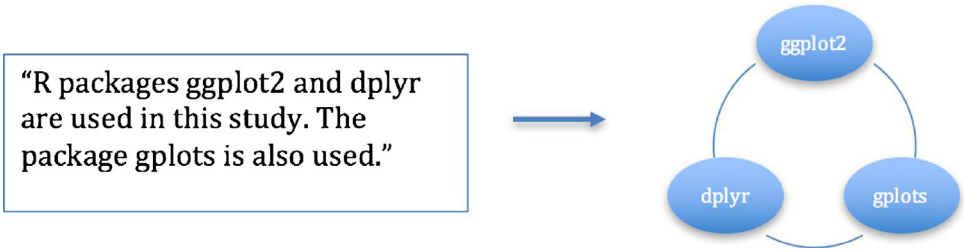

Fig. 1. The construction of the co-mention network.

many of them are simply mentioned without any official citation. Given this fact, we use the term *co-mention* to categorize this relationship for the rest of this paper.

For each of the package identified, we also examined if it is mentioned with a reference. We tested if there is any textual pattern of "bracket + number + bracket" after the package name in the sentence where the package is mentioned, by using the regular expression method. If this pattern is available, then the mentioned R package is determined to be accompanied by a reference.

3.3. Evaluation of the accuracy of entity identification

After the algorithm was implemented in R, it was applied to the full texts of all 13,684 retrieved papers. Before conducting descriptive and network analyses, we randomly selected and manually coded another 100 papers to test our results. On the package level, the precision of our code is 84.4% and the recall is 87.5%. On the paper level, 83 out of the 100 papers have completely correct results in terms of both precision and recall. Six additional papers do not have any mistakenly identified packages, even though some packages mentioned in these papers were unable to be identified by our algorithm.

The same 100 papers were also tested using Stanford Pattern-based Information Extraction and Diagnostics (SPIED), a pattern-based entity extraction and visualization software developed by the Stanford Natural Language Processing Group (Gupta & Manning, 2014). This algorithm is able to automatically learn the textual patterns of the seed terms and use machine learning methods to detect similar terms in the texts. We extracted the sentences mentioning R software in the 100 papers, following steps 1–2 in our algorithm. The texts including these sentences were fed into SPIED, along with the names of the five most commonly used packages in this sample of papers. The following specifications of the system (maximum number of iterations = 15; thresholds of learning phrases and patterns = 0.3) were used. In total, 38 software entities were identified by SPIED, 21 of which were correct based on our manual examination. As such, the precision of SPIED based on this test sample is 23.9%, and the recall is 55.2%—both much lower than our algorithm.

We also examined the cases of misidentification of our in-house algorithm after these baseline tests. A common mistake made by our algorithm is quoted below (with the target package name emphasized by the authors):

> "A paired-end **DNA** library (Illumina) was made according to manufacturer's instructions and sequenced to produce 74.9 million 100 bp x 2 read pairs." (Mantooth et al., 2017)

In this case, "dna" happens to be the name of a package and to be next to the word "library", even though it is not an R package in this context. However, given the nature of our algorithm, such contexts cannot be detected. Because of its favorable performance compared to other common name-entity identification packages, we decided to use this method as-is.

3.4. Network analysis

We created the co-mention network of all identified R packages based on our data. A basic illustration of how this network is constructed in shown in Fig. 1: if any two packages are mentioned in the same paper, even if they are not present in the same sentence, they are seen as being co-mentioned, and are connected through an undirected link. The number of links between two nodes were summed up as the edge weight between two software packages.

Based on the established network, a two-step network analysis was performed. In the first step, we used a few centrality measures, including degree centrality, betweenness centrality, and PageRank, to understand individual R packages' role in connecting with one another. We used the R package "igraph" (Csardi & Nepusz, 2006) to calculate the centrality of the network.

According to the popular interpretations offered by Hanneman & Riddle (2005), **degree centrality** is the total amount of connections an actor has. Degree centrality is the simplest and most direct indicator of the importance of a node in the network. **Betweenness centrality**, on the other hand, is the level to which an actor lies between other pairs of actors in the same network. In his classic interpretation of this concept, Freeman (1978) defined betweenness as an "index of the potential of a point for control of communication" (p. 224). Originally proposed by Brin and Page (1998), **PageRank** is an indicator of the importance of a node obtained by calculating the quality of its inlinks. It was originally designed for directed

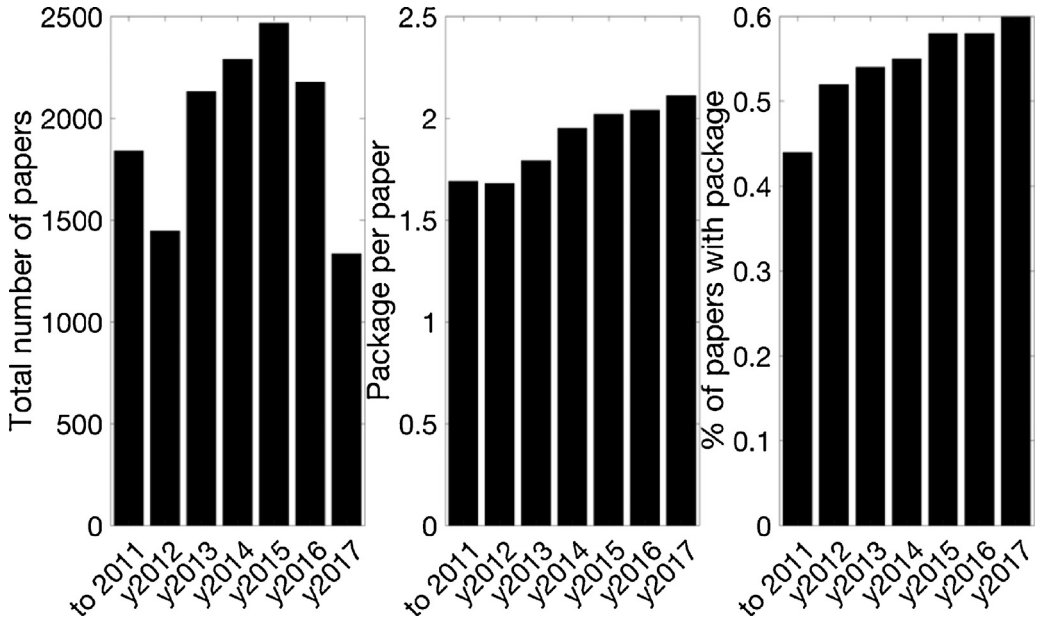

Fig. 2. Total number of papers, mean number of packages per paper with package mention, and percentage of papers with package mention in all papers mentioning R by publication year.

networks but has since been applied to undirected networks in various studies (Grolmusz, 2015; Iván & Grolmusz, 2010; Perra & Fortunato, 2008).

In the second step, we employed a modularity-based clustering technique as implemented in version 1.6.5 of VOSviewer (Van Eck & Waltman, 2010) to understand the stratification and grouping characteristics of the co-mention network. VOSviewer is based on the technique of *visualization of similarities*, which depicts the distance between two objects as the representation of their similarities (Van Eck & Waltman, 2007); as a result, objects more similar to each other are clustered closer in the graph.

4. Result

4.1. Overview of the papers and packages

Of the 13,684 paper we analyzed, 7463 papers (54.5%) mention at least one R package. This number is similar to that found in our prior research (Li et al., 2017), wherein 223 out of 391 papers (57%) mentioned at least one R package. Within these 7463 papers, there are 14,310 *package cases* of 1838 unique packages mentioned or cited. The average number of package cases per paper (1.92) is also very similar to the result reported in our previous study (1.85).

We followed the general approach of our previous study in classifying all PLoS papers based on publication year. Given the relatively small number of papers published before 2012, all the papers up to 2011 are categorized into a single group. Every year from 2012 onward has its own category. Fig. 2 summarizes the total number of papers, the percentage of papers in which any package is identified, and the mean number of packages in papers with any package identified in each group. As shown in this graph, the latter two numbers have increased during the past few years: in line with our previous analysis of a smaller sample of PLoS papers (Li et al., 2017), these results suggest the growing impact of R packages in PLoS journals.

It is worth mentioning that, similar with our previous study (Li et al., 2017), we also identified a strong imbalance among knowledge domains in our dataset. “Biology and life sciences” has a dominant presence in all the papers we collected, with 96.7% of papers falling into this category. “Medicine and health sciences” (60.3%) and “Research and analysis methods” (51.3%) are the other two categories that have more than 50% of papers.

Moreover, we tested the ratio of the number of package cases with reference to all package cases identified. In total, 10,084 package cases were found to have a reference, accounting for 70.5% of all 14,310 package cases. This number is very similar to our previous study (72.1%). We also tested if this ratio is subject to change over time. As shown in Fig. 3, the percentage of papers with reference has only increased slightly during the past few years based on the larger sample of papers collected in this study, from 63.5% by 2011–74.6% in 2017. This gradual increasing trend is also generally similar with the result last time (Li et al., 2017).

Table 1 presents the frequencies of the top 10 packages in terms of total frequency and their relative sizes compared to the total number of package mentions in all analyzed papers.

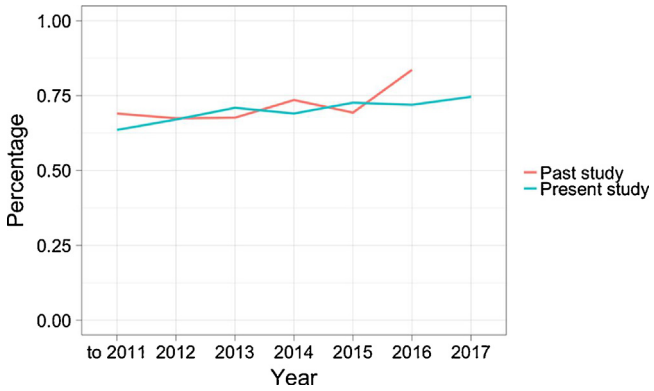

Fig. 3. Package reference-to-mention ratio over time.

Table 1
Top 10 packages identified in our dataset and their relative sizes.

Package	Count	Percentage
lme4	834	5.8%
vegan	738	5.2%
nlme	426	3%
limma	392	2.7%
MASS	301	2.1%
ape	226	1.6%
mgcv	207	1.4%
ggplot2	204	1.4%
survival	200	1.4%
MuMIn	181	1.3%

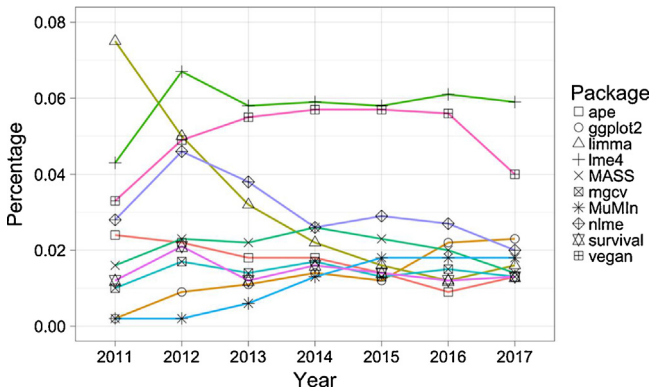

Fig. 4. Change of the relative sizes of the top 10 packages by year.

Fig. 4 shows the change in the relative sizes of the top 10 packages shown in Table 1. Most of these packages have been mentioned in a relatively stable manner throughout the history of PLoS. One distinguishable exception is the package "limma": even though it is still an important R package used in scientific studies, its relative size has decreased dramatically since the early years of PLoS journals. What should be noted is that the pattern of limma is not unique among packages from Bioconductor. The second most frequently used Bioconductor package, "affy," has gone through a similar trend: its size has dropped from 2.9% before 2012–0.4% in 2017. On the other hand, both "ggplot2" and "MuMIn" have become significantly more popular in PLoS papers during the same period. The reasons for such radical changes, however, require further investigation.

4.2. The network of package co-mentions and centrality measures

Fig. 5 is an illustration of the distribution of papers with a given number of packages. In total, we found 3410 papers (24.9% of all papers examined) mentioning at least two packages. These papers are the basis for the network analysis reported in this section.

Overall, we identified 14,615 co-mention pairs among 1612 unique R packages. 1576 of these packages (97.8% of those co-mentioned with any other package) are interconnected with each other. Plotting these as nodes in a network graph using

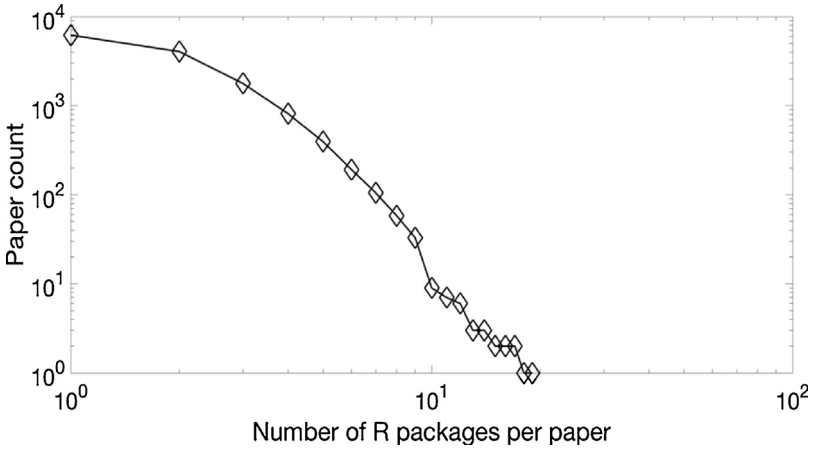

Fig. 5. Distribution of the number of packages per paper.

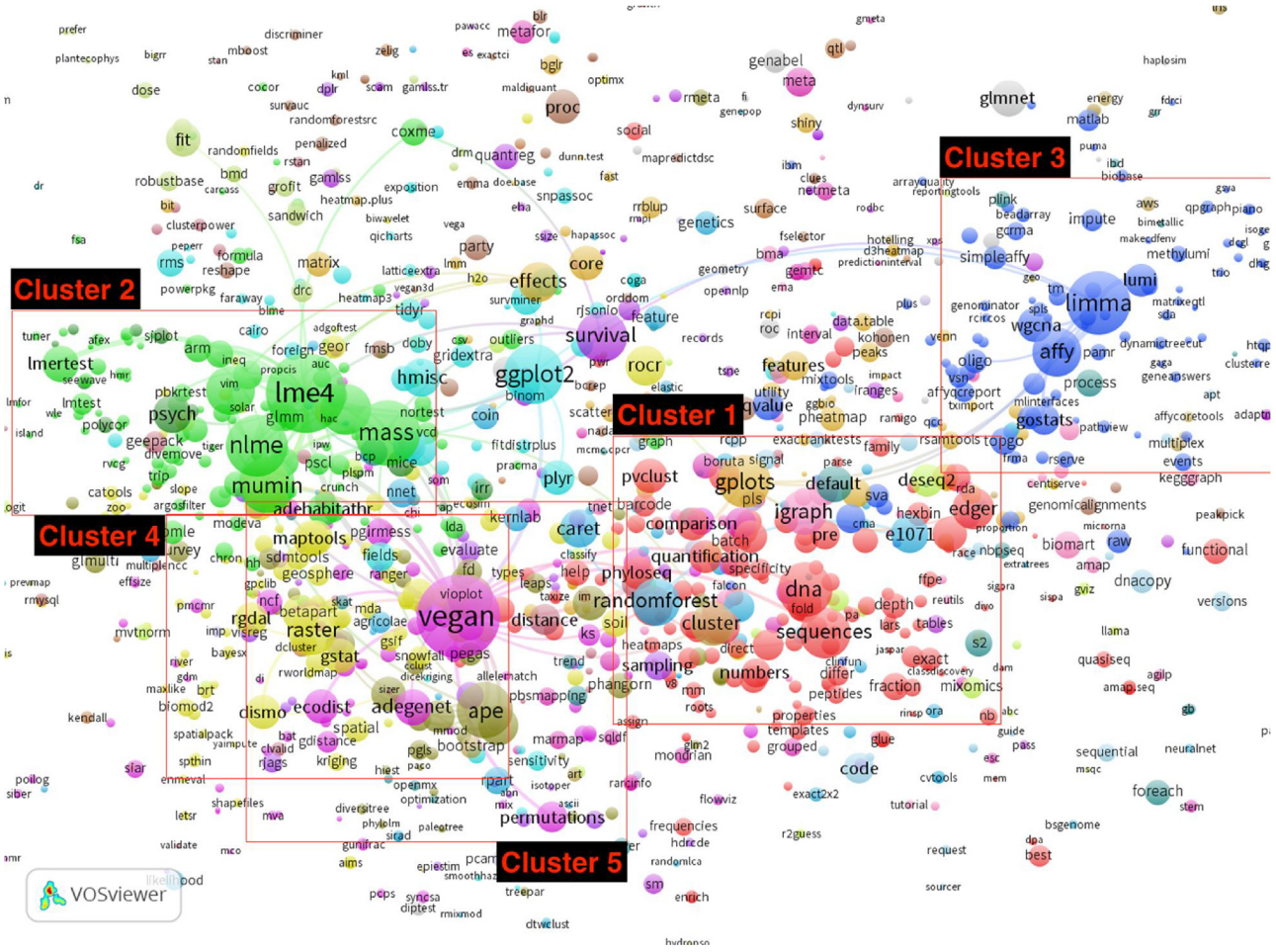

Fig. 6. Co-mention network of all R packages in PLoS database.

VOSviewer (Fig. 6) allows us to see and interpret their clustering behavior. In VOSviewer, we used the clustering function with 50 iterations and minimal cluster size 10. Moreover, for the sake of readability, we elected to show only the top 100 links between all nodes. In total, 21 clusters are identified by VOSviewer, each of which is marked in a distinct color. Fig. 5 also highlights the five largest clusters identified by VOSviewer in terms of total number of nodes. These are discussed in greater detail below.

Table 2
Top clusters and their corresponding top nodes in the network.

Cluster	Top five nodes (Number of links)	Description of the package based on CRAN
1	dna: Differential Network Analysis (203)	Package for conducting differential network analysis from microarray data.
	sequences: Generic and Biological Sequences (148)	Educational package used in R courses to illustrate object-oriented programming and package development. Using biological sequences (DNA and RNA) as a working example.
	edgeR: Empirical Analysis of Digital Gene Expression Data in R (112)	Differential expression analysis of RNA-seq expression profiles with biological replication.
	diveRsity: A Comprehensive, General Purpose Population Genetics Analysis Package (104)	Allows the calculation of both genetic diversity partition statistics, genetic differentiation statistics, and locus informativeness for ancestry assignment.
	Information: Data Exploration with Information Theory (89)	Performs exploratory data analysis and variable screening for binary classification models using weight-of-evidence (WOE) and information value (IV).
2	lme4: Linear Mixed-Effects Models using 'Eigen' and S4 (283)	Fit linear and generalized linear mixed-effects models.
	MASS: Support Functions and Datasets for Venables and Ripley's MASS (264)	Functions and datasets to support Venables and Ripley, "Modern Applied Statistics with S" (4th edition, 2002).
	nlme: Linear and Nonlinear Mixed Effects Models (184)	Fit and compare Gaussian linear and nonlinear mixed-effects models.
	car: Companion to Applied Regression (146)	Functions and Datasets to Accompany J. Fox and S. Weisberg, An R Companion to Applied Regression, Second Edition, Sage, 2011.
	MuMIn: Multi-Model Inference (136)	Model selection and model averaging based on information criteria (AICc and alike).
3	limma: Linear Models for Microarray Data (231)	Data analysis, linear models and differential expression for microarray data.
	Affy: Methods for Affymetrix Oligonucleotide Arrays (115)	The package contains functions for exploratory oligonucleotide array analysis.
	WGCNA: Weighted Correlation Network Analysis (71)	Functions necessary to perform Weighted Correlation Network Analysis on high-dimensional data.
	qvalue: Q-value estimation for false discovery rate control (62)	This package takes a list of p-values resulting from the simultaneous testing of many hypotheses and estimates their q-values and local FDR values.
	multtest: Resampling-based multiple hypothesis testing (51)	Non-parametric bootstrap and permutation resampling-based multiple testing procedures (including empirical Bayes methods) for controlling the family-wise error rate (FWER), generalized family-wise error rate (gFWER), tail probability of the proportion of false positives (TPPFP), and false discovery rate (FDR).
4	raster: Geographic Data Analysis and Modeling (131)	Reading, writing, manipulating, analyzing and modeling of gridded spatial data.
	mgcv: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation (122)	Generalized additive (mixed) models, some of their extensions and other generalized ridge regression with multiple smoothing parameter estimation by (Restricted) Marginal Likelihood, Generalized Cross Validation and similar.
	ROCR: Visualizing the Performance of Scoring Classifiers (100)	ROC graphs, sensitivity/specificity curves, lift charts, and precision/recall plots are popular examples of trade-off visualizations for specific pairs of performance measures.
	sp: Classes and Methods for Spatial Data (82)	Classes and methods for spatial data; the classes document where the spatial location information resides, for 2D or 3D data.
	maps: Draw Geographical Maps (69)	Display of maps.
5	vegan: Community Ecology Package (368)	Ordination methods, diversity analysis and other functions for community and vegetation ecologists.
	adegenet: Exploratory Analysis of Genetic and Genomic Data (78)	Toolset for the exploration of genetic and genomic data.
	ecodist: Dissimilarity-Based Functions for Ecological Analysis (66)	Dissimilarity-based analysis functions including ordination and Mantel test functions, intended for use with spatial and community data.
	settings: Software Option Settings Manager for R (50)	Provides option settings management that goes beyond R's default 'options' function.
	permutations: Permutations of a Finite Set (47)	Manipulates invertible functions from a finite set to itself. Can transform from word form to cycle form and back.

The five largest clusters shown above are analyzed further in Table 2, with the titles and descriptions (taken from their repositories) of the top five packages in each cluster (in terms of the total number of links) listed. It is obvious that these clusters can be partly explained by the functions and disciplines of the packages. For example, the top four packages of the first cluster ("dna", "sequences", "edgeR", and "diveRsity") are related to genetic analysis. However, this cluster might also be subject to misidentification as discussed in the Methods section: "dna" and "sequences" are the two terms found to be most frequently misidentified in our posttest. The third cluster is also related to DNA sequencing analysis; here, however, most of the top packages are from Bioconductor rather than from CRAN. Both of these two clusters can be interpreted by

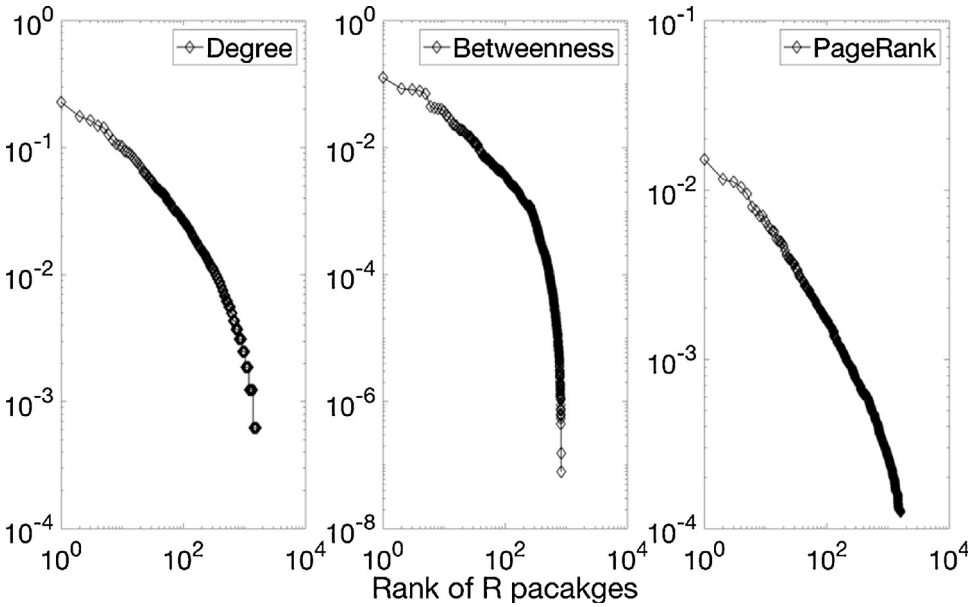

Fig. 7. Distributions of the centrality measures.

Table 3
Top 20 packages and their centrality measures.

Package	Degree (Rank)	Betweenness (Rank)	PageRank (Rank)
lme4	0.1757 (2)	0.0812 (3)	0.0116 (2)
vegan	0.2284 (1)	0.1268 (1)	0.0152 (1)
nlme	0.1142 (7)	0.0376 (10)	0.0076 (7)
limma	0.1433 (5)	0.0846 (2)	0.0104 (4)
MASS	0.1639 (3)	0.0785 (4)	0.0112 (3)
ape	0.1068 (8)	0.0403 (9)	0.0071 (8)
mgcv	0.0757 (18)	0.019 (19)	0.0048 (19)
ggplot2	0.149 (4)	0.0705 (5)	0.0095 (5)
survival	0.1024 (10)	0.0444 (6)	0.007 (9)
MuMIn	0.0844 (15)	0.015 (26)	0.0052 (15)
gplots	0.1036 (9)	0.0316 (11)	0.0065 (10)
multcomp	0.0633 (24)	0.0117 (34)	0.0041 (22)
car	0.0906 (13)	0.023 (14)	0.0057 (14)
dna	0.126 (6)	0.0419 (7)	0.008 (6)
affy	0.0714 (20)	0.0204 (17)	0.005 (16)
randomForest	0.0875 (14)	0.0301 (12)	0.0059 (12)
ade4	0.0739 (19)	0.0156 (24)	0.0047 (20)
igraph	0.0937 (11)	0.0406 (8)	0.0061 (11)
raster	0.0813 (16)	0.0187 (20)	0.005 (16)
adegenet	0.0484 (36)	0.009 (38)	0.0032 (35)

the fact that most of the papers we analyzed belong to biology and life sciences. The second, fourth, and fifth clusters largely correspond to the functions of linear mixed-effects modeling, data analysis, and ecological data analysis, respectively.

Fig. 7 shows the distributions of degree centrality, betweenness centrality, and PageRank for all packages in the network. Consistent with findings on author co-citation analysis (e.g., Yan & Ding, 2009), all three measures are bound by power laws. Based on the context of this co-mention network and the definitions of the three measures discussed in the Methods section, we operationalized **degree centrality** as the frequency with which an R package is co-mentioned with any other R package, **betweenness centrality** as the level to which an R package is co-mentioned with any other distinct R package, and PageRank as the level of total importance (in terms of the total number of links in the case of undirected network) of all R packages that are co-mentioned with a specific package.

A correlation test was conducted between the total count of packages mentions and the three centrality measures adopted in this analysis. All four measures are strongly positively correlated with each other, with the value of r ranging from 0.83 (count-degree) to 0.99 (degree-PageRank). This result suggests a strong consistency among all these indicators of the importance of an R package.

Table 3 shows the three centrality measures as well as the rankings in all these measures of the top 20 packages in terms of total count (the top 10 of which were shown in Table 1). This table lends further support to the conclusion that our measures for the top R packages are relatively consistent with each other.

Obviously, however, the measures are not perfectly consistent; discernibly different patterns exist among the packages. For example, "ggplot2", "dna", and "igraph" are the three packages whose centrality rankings are higher than their total counts, suggesting that these packages are more likely to be co-mentioned with other R packages. Among these three, "ggplot2" and "igraph" are the only two packages in the top 20 list that are dedicated to visualization tasks. On the other hand, a few packages have significant lower rankings in centrality measures than in terms of total frequency. Many of these packages, including "nlme", "mgcv", and "multcomp", are used for modeling-related tasks. The connection between a package's function/discipline and the measures of its importance **requires** further study.

5. Conclusions

This paper explores, for the first time, the attributes of a network formed by software entities being co-mentioned in scientific papers. We first developed an in-house algorithm to identify R packages from the full text of all PLoS journals that cited or mentioned R. Based on the results of name-entity extraction, we plotted a co-mention network composed of R packages. We applied three centrality measures to understand the importance of the central nodes in this network. Moreover, we examined the clustering structure of the network to propose some tentative explanations for the co-occurrence patterns of R packages.

The most notable contribution of this study is methodological: we established a framework to examine the impact of software entities based not only on citation counts, but on relationships with other software entities. This novel framework subjects scientific software entities to the combined methods of scientometrics and network analysis. In doing so, it extends the scope of scientometrics and scholarly communication research by including a new and important type of research object. Moreover, our framework has strong potential to be extended in the future to answer more sociologically-oriented questions about the interrelationship of software, scientific activities, and scientific publication. For example, one can expect to better understand how software entities related to different knowledge domains and how scientific tasks are connected to each other and potentially to datasets, in the space of scientific publication. This pattern could reveal useful information of the roles played by software in the full pipeline of scientific knowledge production: how they are used in scientific activities and how these uses are inscribed in scientific texts (Knorr, 1981; Swales, 1990).

As the first large-scale scientometric study focusing on entities in the same software ecosystem, our results establish that the patterns of software co-mention are similar to those found for other types of scholarly objects. For example, the distribution of centrality measures is similar to what has been found in author co-citation analysis (e.g., Liu, Bollen, Nelson, & Van de Sompel, 2005; Yan & Ding, 2009). Despite the similarities between a software co-mention network and an author co-citation network, the implications of software entities as a research object in scientometrics and network analysis can only be fully understood by more comparative studies. At least two considerations motivate such further research: first, centrality measures in a software co-mention network bear meanings that are distinct from those of functionally similar measures in other types of co-citation network (e.g., author or paper networks). Furthermore, these meanings might be influenced by the functional and disciplinary attributes of the software entities. This knowledge will better posit software entities in the family of scientific objects, and help us reach more solid conclusions about the roles they play in scientific research.

In addition to descriptive analysis, we also interpreted the clustering structure of the R package co-mention network. Examining the network graph, we observed that functional and disciplinary connections among packages seem to be two important factors contributing to some of the more prominent clusters. This suggests that, in general, co-mentioned R packages have functional and/or disciplinary similarities or connections—a conclusion which sheds light on how scientific software is used in laboratory activities, at least in the context of biology and life sciences. Our observations about the clustering structure of the R package co-mentioning network might be used to facilitate existing and future open software package recommendation systems, so that researchers can find useful computational tools for their research. We freely admit, however, that our interpretations of the clusters are preliminary and must be revisited by future case studies using both qualitative and quantitative methods. An especially significant reason for more case studies is that as most of the papers in our sample are within the domain of biology and life sciences, it should be expected that scientists in other knowledge domains might use different software packages in different ways.

This study is only the first step in understanding how scientific software is embedded in scientific activities from a quantitative perspective. In our future research, we will try to expand our investigation into other scientific paper databases and software ecosystems, in order to establish a more broadly comparative view of the relationship between scientific papers and software. In addition, we will focus on the micro-level processes by which scientific software is used in research and is described in scientific outputs, which will help us better understand the findings reported in this paper. Last but not the least, we will refine our name-entity recognition algorithm by introducing more advanced NLP techniques, to more accurately identify software entities based on the contexts in which they are mentioned in papers.

Author contributions

Kai Li: Conceived and designed the analysis; Collected the data; Contributed data or analysis tools; Perfomed the analysis; Wrote the paper.

Erjia Yan: Conceived and designed the analysis; Contributed data or analysis tools; Proofread the article.

Appendix A. : An example of scoring a sentence

We take the following sentence from Jiménez-Gómez, Wallace, and Maloof (2010) as an example of how our algorithm scores terms, as described in Section 3.2:

> "We obtained trait indexes for each RIL in the sun and shade environments fitting mixed effect models using the lme4 package in R [60], [61]." (p. 10)

The final scores of all the terms in this sentence are shown in Table A1:

Table A1
The scoring of an exemplary sentence using our algorithm.

Term	Score
we	0
obtained	0
trait	0
indexes	0
ril	0
sun	0
shade	0
environments	0
fitting	0
mixed	0
effect	0
models	0
using	0
lme4	1
package	0

References

Ahlgren, P., Jarneving, B., & Rousseau, R. (2003). Requirements for a cocitation similarity measure, with special reference to Pearson's correlation coefficient. *Journal of the American Society for Information Science and Technology; Hoboken*, *54*(6), 550–560.

Baumer, B., Kaplan, D., & Horton, N. J. (2017). *Modern data science with R*.

Bellardo, T. (1980). The use of co-citations to study science. *Library Research*, *2*(3), 231–237.

Berry, D. (2016). *The philosophy of software: Code and mediation in the digital age*. Springer. Retrieved from. https://books.google.com/books?hl=en&lr=&id=GeYgDAAAQBAJ&oi=fnd&pg=PR1&dq=philosophy+of+software&ots=268PNRJJtr&sig=1eRNhPKVht_ZOlHqGt5RzKLb8og

Boettiger, C., Chamberlain, S., Hart, E., & Ram, K. (2015). Building software, building community: Lessons from the ROpenSci project. *Journal of Open Research Software*, *3*(1). Retrieved from. http://openresearchsoftware.metajnl.com/articles/10.5334/jors.bu/?toggle_hypothesis=on

Brin, S., & Page, L. (1998). *The anatomy of a large-scale hypertextual Web search engine*. pp. 107–117. *Computer networks and ISDN systems* (Vol. 30) Elsevier. Retrieved from. http://cat.inist.fr/?aModele=afficheN&cpsidt=2304904

Chamberlain, S., Boettiger, C., & Ram, K. (2016). *Rplos: Interface to the search 'API 'for 'PLoS' journals. R package version 0.6. 4*.

Chen, C., & Paul, R. J. (2001). Visualizing a knowledge domain's intellectual structure. *Computer*, *34*(3), 65–71.

Chen, C. (1999). Visualising semantic spaces and author co-citation networks in digital libraries. *Information Processing & Management*, *35*(3), 401–420.

Claes, M., Mens, T., & Grosjean, P. (2014). On the maintainability of CRAN packages. In *2014 software evolution week-IEEE conference on software maintenance, reengineering and reverse engineering (CSMR-WCRE)* (pp. 308–312). Retrieved from. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6747183

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research InterJournal. *Complex Systems*, *1695*(5), 1–9.

Decan, A., Mens, T., Claes, M., & Grosjean, P. (2015). On the development and distribution of R packages: an empirical analysis of the R ecosystem. In *Proceedings of the 2015 european conference on software architecture workshops* (p. 41). Retrieved from. http://dl.acm.org/citation.cfm?id=2797476

Ding, Y., Chowdhury, G., & Foo, S. (2000). Journal as markers of intellectual space: Journal co-citation analysis of information retrieval area, 1987–1997. *Scientometrics*, *47*(1), 55–73.

Ding, Y., Song, M., Han, J., Yu, Q., Yan, E., Lin, L., & Chambers, T. (2013). Entitymetrics: Measuring the impact of entities. *PUBLIC LIBRARY OF SCIENCE*, *8*(8), e71416. http://dx.doi.org/10.1371/journal.pone.0071416

Driscoll, K., & Walker, S. (2014). Big data, big questions| working within a black box: Transparency in the collection and production of big twitter data. *International Journal of Communication*, *8*, 20.

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social Networks*, *1*(3), 215–239.

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., et al. (2004). Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, *5*(10), 1.

German, D. M., Adams, B., & Hassan, A. E. (2013). The evolution of the R software ecosystem. In *2013 17th european conference on software maintenance and reengineering (CSMR)* (pp. 243–252). Retrieved from. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6498472

Gillespie, T. (2014). *The relevance of algorithms. media technologies: Essays on communication, materiality, and society*. pp. 167. Retrieved from. http://books.google.com/books?hl=en&lr=&id=zeK2AgAAQBAJ&oi=fnd&pg=PA167&dq=info:jgo7uoqGxjUJ:scholar.google.com&ots=GngEQ-U0Ai&sig=QJknz9uFTfCu5bDut_SSypTc_6o

Gmür, M. (2003). Co-citation analysis and the search for invisible colleges: A methodological evaluation. *Scientometrics*, *57*(1), 27–57.

Grolmusz, V. (2015). A note on the pagerank of undirected graphs. *Information Processing Letters*, *115*(6), 633–634.

Gupta, S., & Manning, C. D. (2014). *Spied: Stanford pattern-based information extraction and diagnostics*. pp. 38. Idibon: Sponsor.

Hanneman, R. A., & Riddle, M. (2005). *Introduction to social network methods*. University of California Riverside. Retrieved from. https://www.researchgate.net/profile/Robert_Hanneman/publication/235737492_Introduction_to_Social_Network_Methods_Vol._13/links/0deec52261e1577e6c000000.pdf

Hey, T., Tansley, S., Tolle, K. M., et al. (2009).. *The fourth paradigm: data-intensive scientific discovery* (Vol. 1) Redmond, WA: Microsoft research. Retrieved from. https://www.fh-potsdam.de/fileadmin/user_upload/fb-informationswissenschaften/bilder/forschung/tagung/isi_2010/isi_programm/TonyHey_-_eScience_Potsdam__Mar2010____complete_.pdf
Hou, J., & Chen, H. (2011). Countries co-citation network and research fronts of international energy technology. In *2011 international conference on advances in social networks analysis and mining (ASONAM)* (pp. 551–552). Retrieved from. http://ieeexplore.ieee.org/abstract/document/5992659/
Howison, J., & Bullard, J. (2015). 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. http://dx.doi.org/10.1002/asi.23538
Hu, C.-P., Hu, J.-M., Gao, Y., & Zhang, Y.-K. (2011). A journal co-citation analysis of library and information science in China. *Scientometrics*, *86*(3), 657–670.
Ince, D. C., Hatton, L., & Graham-Cumming, J. (2012). The case for open computer programs. *Nature*, *482*(7386), 485–488. http://dx.doi.org/10.1038/nature10836
Iván, G., & Grolmusz, V. (2010). When the Web meets the cell: Using personalized PageRank for analyzing protein interaction networks. *Bioinformatics*, *27*(3), 405–407.
Jiménez-Gómez, J. M., Wallace, A. D., & Maloof, J. N. (2010). Network analysis identifies ELF3 as a QTL for the shade avoidance response in arabidopsis. *PLOS Genetics*, *6*(9), e1001100. http://dx.doi.org/10.1371/journal.pgen.1001100
Katz, D. S., & Smith, A. M. (2015). Transitive credit and JSON-LD. *Journal of Open Research Software*, *3*(1). Retrieved from. http://openresearchsoftware.metajnl.com/articles/10.5334/jors.by/
Knorr, K. D. (1981). *The manufacture of knowledge an essay on the constructivist and contextual nature of science*.. Retrieved from. https://philpapers.org/rec/KNOTMO-2
Li, K., Greenberg, J., & Lin, X. (2016). Software citation, reuse and metadata considerations: an exploratory study examining LAMMPS. *Proceedings of the 79th ASIS&T annual meeting*, *Vol. 53*. Retrieved from. http://dl.acm.org/citation.cfm?id=3017519
Li, K., Yan, E., & Feng, Y. (2017). How is R cited in research outputs? Structure, impacts, and citation standard. *Journal of Informetrics*, *11*(4), 989–1002. http://dx.doi.org/10.1016/j.joi.2017.08.003
Liu, X., Bollen, J., Nelson, M. L., & Van de Sompel, H. (2005). Co-authorship networks in the digital library research community. *Information Processing & Management*, *41*(6), 1462–1480.
Liu, Z. (2005). Visualizing the intellectual structure in urban studies: a journal co-citation analysis (1992–2002). *Scientometrics*, *62*(3), 385–402.
Manovich, L. (2013).. *Software takes command* (Vol. 5) A&C Black. Retrieved from. https://books.google.com/books?hl=en&lr=&id=RarTbJ84EnMC&oi=fnd&pg=PR9&dq=manovich+software+takes+command&ots=j_b-bIdDxr&sig=hOLWTFwtmkWCqpG1yduDxgD-6sw
Mantooth, K., Hadziabdic, D., Boggess, S., Windham, M., Miller, S., Cai, G., et al. (2017). Confirmation of independent introductions of an exotic plant pathogen of Cornus species, Discula destructiva, on the east and west coasts of North America. *PUBLIC LIBRARY OF SCIENCE*, *12*(7), e0180345. http://dx.doi.org/10.1371/journal.pone.0180345
Marwick, B. (2016). Computational reproducibility in archaeological research: Basic principles and a case study of their implementation. *Journal of Archaeological Method and Theory*, 1–27.
McCain, K. W. (1990). Mapping authors in intellectual space: A technical overview. *Journal of the American Society for Information Science*, *41*(6), 433.
Mens, T., Claes, M., & Grosjean, P. (2014). ECOS: Ecological studies of open source software ecosystems. In *2014 software evolution week-IEEE conference on software maintenance, reengineering and reverse engineering (CSMR-WCRE)* (pp. 403–406). Retrieved from. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6747205
Mombers, C., Van Heeringen, A., Van Venetië, R., & Le Pair, C. (1985). Displaying strengths and weaknesses in national R&D performance through document cocitation. *Scientometrics*, *7*(3–6), 341–355.
Muenchen, R. A. (2012). *The popularity of data analysis software*.. URL Http://R4stats. Com/Popularity. Retrieved from. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.565.3929&rep=rep1&type=pdf
Nerur, S. P., Rasheed, A. A., & Natarajan, V. (2008). The intellectual structure of the strategic management field: An author co-citation analysis. *Strategic Management Journal*, *29*(3), 319–336.
Niemeyer, K. E., Smith, A. M., & Katz, D. S. (2016). The challenge and promise of software citation for credit, identification, discovery, and reuse. *ArXiv Preprint ArXiv:1601.04734*. Retrieved from http://arxiv.org/abs/1601.04734.
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.
Pan, X., Yan, E., & Hua, W. (2016). Disciplinary differences of software use and impact in scientific literature. *Scientometrics*, *109*(3), 1593–1610.
Pathak, M. A. (2017). *Beginning data science with R*.
Perra, N., & Fortunato, S. (2008). Spectral centrality measures in complex networks. *Physical Review E*, *78*(3), 036107.
Pia, M. G., Basaglia, T., Bell, Z. W., & Dressendorfer, P. V. (2009). Geant4 in scientific literature. In *2009 IEEE nuclear science symposium conference record (NSS/MIC)* (pp. 189–194). Retrieved from. http://ieeexplore.ieee.org/abstract/document/5401810/
Pia, M. G., Basaglia, T., Bell, Z. W., & Dressendorfer, P. V. (2010). The impact of Monte Carlo simulation: A scientometric analysis of scholarly literature. *ArXiv Preprint ArXiv:1012.3305*. Retrieved from https://arxiv.org/abs/1012.3305.
Pia, M. G., Basaglia, T., Bell, Z. W., & Dressendorfer, P. V. (2012). Publication patterns in HEP computing. *Journal of Physics: Conference Series*, *396*, 062015. IOP Publishing. Retrieved from. http://iopscience.iop.org/article/10.1088/1742-6596/396/6/062015/meta
Piwowar, H., & Priem, J. (2016). *Depsy: Valuing the software that powers science*.. Retrieved from. https://github.com/Impactstory/depsy-research/blob/master/introducing_depsy.md
Priem, J., & Piwowar, H. (2013). *Toward a comprehensive impact report for every software project*. pp. 790651. Figshare.
Ruhleder, K. (1994). 'Pulling down' books vs.'pulling up' files: Textual databanks and the changing culture of classical scholarship. *The Sociological Review*, *42*(S1), 181–195.
Ruhleder, K. (1995). Reconstructing artifacts, reconstructing work: From textual edition to on-line databank. *Science, Technology, & Human Values*, *20*(1), 39–64.
Small, H., & Crane, D. (1979). Specialties and disciplines in science and social science: An examination of their structure using citation indexes. *Scientometrics*, *1*(5–6), 445–461.
Small, H., Sweeney, E., & Greenlee, E. (1985). Clustering the science citation index using co-citations. II. mapping science. *Scientometrics*, *8*(5–6), 321–340.
Small, H. (1973). Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, *24*(4), 265–269. http://dx.doi.org/10.1002/asi.4630240406
Smith, A. M., Katz, D. S., & Niemeyer, K. E. (2016). Software citation principles. *PeerJ Computer Science*, *2*, e86. http://dx.doi.org/10.7717/peerj-cs.86
Swales, J. (1990). *Genre analysis: English in academic and research settings*. Cambridge University Press.
Tippmann, S. (2014). Programming tools: Adventures with r. *Nature*, *517*(7532), 109–110. http://dx.doi.org/10.1038/517109a
Tsay, M., Xu, H., & Wu, C. (2003). Journal co-citation analysis of semiconductor literature. *Scientometrics*, *57*(1), 7–25.
Van Eck, N. J., & Waltman, L. (2007). VOS: A new method for visualizing similarities between objects. In *Advances in data analysis*. pp. 299–306. Springer. Retrieved from. https://link.springer.com/chapter/10.1007/978-3-540-70981-7_34
Van Eck, N. J., & Waltman, L. (2010). Software survey: VOSviewer, a computer program for bibliometric mapping. *Scientometrics*, *84*(2), 523–538.
White, H. D., & Griffith, B. C. (1981). Author cocitation: A literature measure of intellectual structure. *Journal of the Association for Information Science and Technology*, *32*(3), 163–171.
White, H. D., & McCain, K. W. (1998). Visualizing a discipline: An author co-citation analysis of information science, 1972–1995. *Journal of the American Society for Information Science*, *49*(4), 327–355.

Wickham, H. (2015). *R packages*. O'Reilly Media, Inc. Retrieved from. https://books.google.com/books?hl=en&lr=&id=DqSxBwAAQBAJ&oi=fnd&pg=PR3&dq=r+packages+wickham&ots=am14LUQFHb&sig=3eFJlkvBBo3lHRqjSPlkPgMOFWc
Wickham, H. (2016). *R for data science: Visualize, model, transform, tidy, and import data*. O'Reilly Media.
Yan, E., & Ding, Y. (2009). Applying centrality measures to impact analysis: A coauthorship network analysis. *Journal of the Association for Information Science and Technology*, *60*(10), 2107–2118.
Zhao, R., & Wei, M. (2017). Impact evaluation of open source software: An Altmetrics perspective. *Scientometrics*, *110*(2), 1017–1033. http://dx.doi.org/10.1007/s11192-016-2204-y
Zumel, N., & Mount, J. (2016). *Practical data science with R*. Shelter Island, NY: Manning Publications Co.