revtools: bibliographic data visualization for evidence synthesis in R

Martin J. Westgate

martin.westgate@anu.edu.au

Fenner School of Environment & Society, The Australian National University, Acton ACT 2601

Abstract

Evidence synthesis (ES) is the process of summarizing scientific information, incorporating a range of methods including systematic reviews, systematic maps, and meta-analyses. ES is critical for translating scientific information into recommendations for management or policy, but is becoming increasingly challenging due to rapid growth in publication of scientific literature. From a computational perspective, the ES process can be viewed as a sequence of operations on bibliographic data, including importing, de-duplicating, and classifying large numbers of articles or reports. These are common tasks in natural language processing and machine learning, yet few software tools are available to help researchers access these techniques for manipulating bibliographic data in a systematic way. Here, I present 'revtools', an R package for exploratory investigation of bibliographic data during reviews and evidence syntheses. It provides tools for the import and de-duplication of bibliographic data formats, and cluster analysis and visualization of article titles, abstracts and keywords using topics models. The key function of 'revtools' is an interactive viewer window that allows users to view and select and export the articles, terms or topics most relevant to their study. Rather than generating lists of text commonly provided by other article sorting software, 'revtools' displays this content as points in an ordination cloud, allowing rapid, intuitive assessment of patterns within the corpus. These tools will assist users to identify key topics and outlying articles or terms, increasing the navigability and ease of processing of bibliographic datasets.

Introduction

The goal of linking scientific information with management practice to achieve 'evidence-based management' has a long history, and is prevalent in a range of disciplines including healthcare (Sackett 1997), social policy (Young *et al.* 2002), and environmental conservation (Sutherland *et al.* 2004). Central to achieving evidence-based management is 'evidence synthesis' (ES); the collation of relevant scientific information into tractable recommendations (Collaboration for Environmental Evidence 2013). The range of approaches available to support ES has grown enormously in recent years, with systematic reviews, systematic maps, and meta-analysis among the most commonly-used methods (Cook *et al.* 2017; Dicks *et al.* 2017). However, this same period has seen scientific publication increase at ~3% per year (Bornmann & Mutz 2015), with over 50 million articles now thought to exist (Jinha 2010). Unsurprisingly, the task of completing ES projects in this high-growth environment has become exceedingly onerous. In healthcare, for example, the average systematic review takes a team of five people 67 weeks to complete (Borah *et al.* 2017), equivalent to approximately 12,000 person hours. Consequently, the combination of large quantities of literature and manual sorting of article lists is driving the growth of the 'synthesis gap'; an incapacity of the scientific establishment to synthesize its' own research for application to societal problems (Westgate *et al.* 2018).

One way to close the synthesis gap is by drawing on machine learning to automate parts of the ES process (Hemens & Iorio 2017). For example, it is common for academic databases and search engines to return some content that is not relevant to the users' needs, because any set of search terms will tend to include words that can be used in different ways within the scientific literature (i.e. they are homonyms; Westgate & Lindenmayer 2017). Recently, Roll, Correia and Berger-Tal (2017) demonstrated a method for classifying articles in the presence of widespread homonymy, using artificial neural networks to differentiate between the medical and ecological literatures discussing the term 'restoration'. This is just one example of a group of methods that have been developed for classifying scientific information in ES projects (reviewed by O'Mara-Eves *et al.* 2015). ES

practitioners now have access to a range of software options, some of which include machine-learning-based classification as standard (see Kohl *et al.* 2018 for a discussion of available software).

What is currently lacking is an ES support tool that draws on machine learning methods in the R environment, which is a problem for several reasons. First, ecologists are increasingly accustomed to drawing on novel techniques in open source programming languages such as R or Python (Mislan, Heer & White 2016), meaning standard statistical methods can remain rarely used in these fields simply because they have not been implemented in these languages. This appears to be the case for machine-supported ES, as the only R package to support article sorting (metagear; Lajeunesse 2016) does not implement machine learning, despite providing a range of other useful features. Second, as a widely-used open source platform, developments in R can occur rapidly and still be thoroughly evaluated by potential users (Wallace *et al.* 2017), which is difficult for software whose source code is unavailable. Finally, the development of packages that incorporate Shiny apps (Chang *et al.* 2016) allows the flexibility of R's command line interface (CLI) to be combined with the benefits of a graphical user interface (GUI) to facilitate uptake among new users (e.g. Kass *et al.* 2018). This combined approach is difficult to achieve in standalone software.

In this paper, I present 'revtools' (short for 'review tools'; Westgate 2018), an R package for importing and manipulating bibliographic data, combined with machine learning and data visualization tools to facilitate article sorting during ES projects. 'revtools' is designed to be as intuitive and automated as possible, while still allowing users to investigate the underlying models and data (see Fig. 1 for a flow diagram of core functions and their relations). Below I describe the functionality of 'revtools' in order of a typical ES workflow.

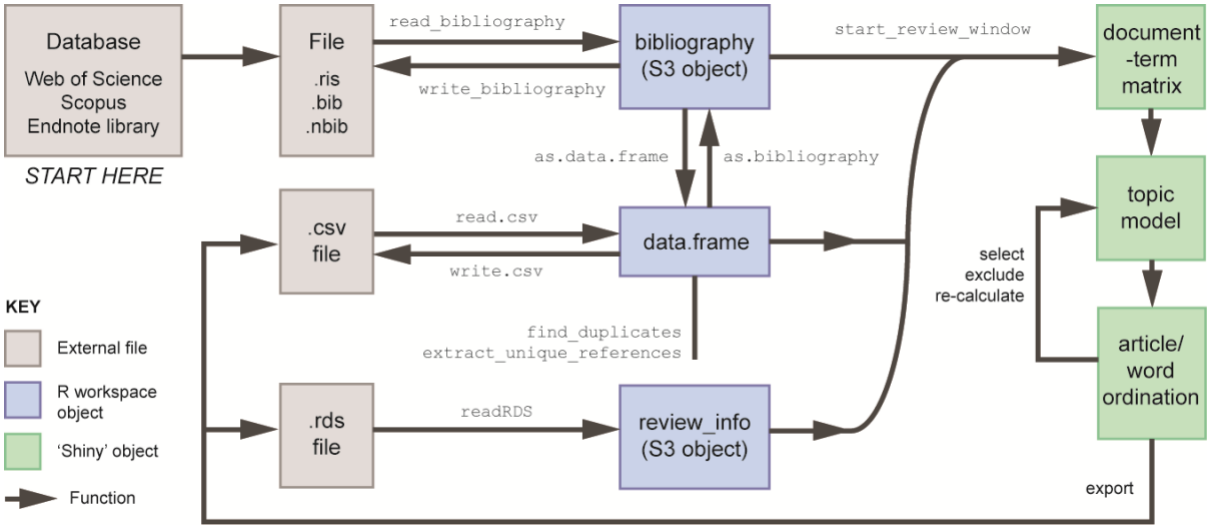

Figure 1: *Flow diagram showing connections between key objects and functions in 'revtools'.*

Import and data storage

When running an ES project, it is common to have bibliographic data stored in files generated by a bulk download of some kind, typically from an academic database such as Web of Science or Scopus. However, bibliographic data can be stored in a range of formats with differing properties (with BibTeX and RIS formats among the most common). Although there are existing formats for importing BibTeX data (e.g. RefManageR; McLean 2017), there is no available function to identify the format of a bibliographic data file and import it in a consistent manner. This is problematic for tagged styles for which no standard formatting exists, meaning that two files might have identical content but different filename suffixes (such as .ciw and .ris), or have the same suffix but different formats. There is also no existing format that can combine different bibliographic data formats in a sensible way.

'revtools' solve these problems by providing a new function ('read_bibliography') that auto-detects tag data and article delimiters, and storing them in a new format called class 'bibliography'. Intended as a simple object class for dealing with bibliographic data, this class uses lists to retain a nested structure like that of the RIS or BibTeX formats that it stores. Each 'bibliography' object is a list in which each entry is an article, consisting of another list storing standard bibliographic data such as the

title, author or keywords for that reference. The list format is useful because it allows storage of vectors of different lengths; for example, each article typically only has one title but may have many authors. Class 'bibliography' contains standard 'print', 'summary' and subset functions, as well as an 'as.data.frame' function and the converse 'as.bibliography' to back-convert for export to reference management software (via 'write_bibliography').

De-duplication

A key component of systematic review protocols is that ES practitioners should search a number of databases to ensure that all potentially-relevant literature is detected (Collaboration for Environmental Evidence 2013). This approach is very thorough, but has the side effect of introducing duplicates into the resulting bibliography, all of which need to be removed prior to article screening. From a computational perspective, this de-duplication is challenging because of the large number of entries that need to be checked, and difficulties in balancing sensitivity and specificity of the algorithm (Rathbone *et al.* 2015).

In 'revtools', the function 'find_duplicates' has been built to locate duplicate articles within large bibliographies. This function works on any 'data.frame' containing usable bibliographic information, and returns a second 'data.frame' that has the same number of rows as the input, but also contains an extra column (named 'group') that groups duplicates according to shared integer values. The subsequent function 'extract_unique_references' allows the user to extract the reference from each group that contains the most information. In combination, these functions not only allow removal of duplicates, but also provide information about the source of those duplicates, allowing the user to check which articles were available in each database (e.g. Fig. 2).

The function 'find_duplicates' works by fuzzy matching of article titles, journal titles, and publication years. The algorithm starts by create a lookup table of potentially synonymous journal titles. For example, society journals are sometimes listed with a description after the journal title (e.g. 'A journal of the society for X'), and may also have differences in capitalization and formatting. 'revtools' accounts for this by looking for long journal titles whose key terms are identical (and in the same

order) to those of shorter journal titles. Any such matches are labelled as potential synonyms and passed to the title matching sequence. This is followed by a second stage, which uses a 'while' loop to identify potentially matching article titles. Starting with the first article, the algorithm identifies a candidate set of all other articles that appear to be in the same journal (including potentially synonymous journals identified previously), and are within 1 year of the specified publication date. To avoid mistakenly excluding incomplete records, the candidate set also includes all articles that are lacking journal and publication year data. The algorithm then calculates the similarity of the target title to all article titles in the candidate set (using 'stringdist'; Van der Loo 2014), after converting all titles to lower case. Candidate titles that differ from the target title by 10 characters or fewer (i.e. 'string distance' of ≤10) are labelled as synonymous, and are removed from later consideration.

This combined, fuzzy matching approach increases effectiveness of the de-duplication process in several ways. First, by restricting comparisons to a subset of relevant journals and publication years it greatly reduces the number of pairwise comparisons that need to be made by 'stringdist'. This is important because a method that compared all titles at once would be prohibitively slow when applied to large datasets. Second, distance-based matching as employed here avoids the problem of failing to match titles that are otherwise identical, but have different formatting (such as extra punctuation or capitalization). Third, 'while' loops are efficient because 'matched' articles are excluded from consideration, meaning that the loop only runs as many times as there are unique articles, and not as many times as there are rows in the dataset. In this way, the number of articles compared in each iteration of the loop decreases as the algorithm progresses. In contrast, a 'for' loop or vectorized approach (using 'apply' functions) would check every article against every other article in each iteration, which is considerably slower.

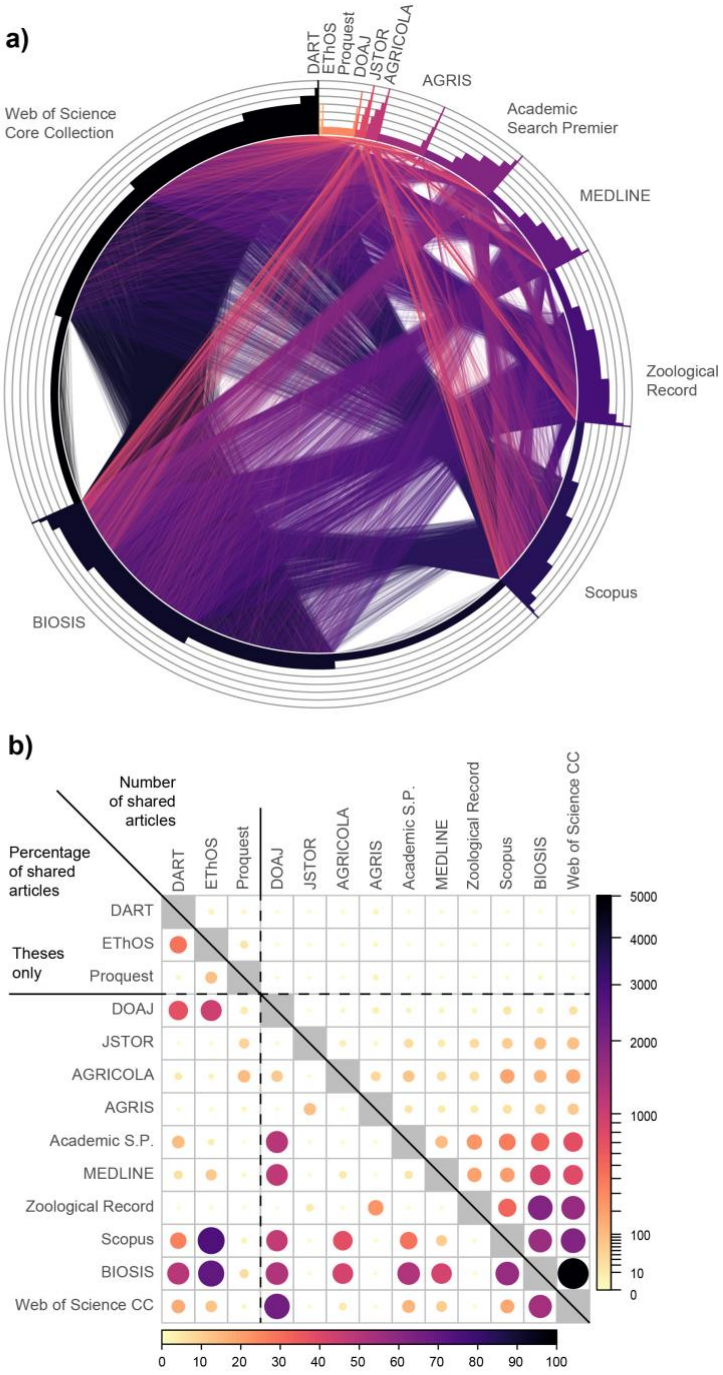

Figure 2: *Example of de-duplication analyses, using data from a systematic review conducted by Mistra EviEM (Stockholm Environment Institute; http://www.eviem.se/en/projects/Buffer-strips/). Panel (a) shows each search result (n= 31,369) as a point on the circumference of the circle, with articles that are present in more than on dataset linked by straight lines (n= 22,311). Following de-duplication, this dataset was estimated to contain 18,433 unique articles (i.e. 41% were duplicates). Lines are colored according to the smaller of the two linked datasets. Marginal histograms show the number of databases each article is shared among. Panel (b) shows the same data, but in summarized form. Databases are arranged in order of their contribution to the bibliography.*

Topic modelling

The core concept underpinning 'revtools' is that visualizing article similarity as a cloud of points can be easier to navigate, and more intuitive, than sorting lists of article titles. Drawing this point cloud requires a method for quantifying article similarity, for which 'revtools' uses existing machine learning methods – specifically 'topic models' (Blei 2012). The key new function for building these topic models is 'make_DTM', which constructs a document-term matrix (DTM), the standard input to many machine learning algorithms. Specifically, 'make_DTM' takes whatever data is available from titles, keywords and abstracts of each article, concatenates them, then applies several standard text processing functions from the 'tm' package (Meyer, Hornik & Feinerer 2008). These include conversion to lower case, and removal of punctuation, numbers and 'stop' words (such as 'the', 'and' etc). 'make_DTM' also performs stemming on all supplied terms; for example the terms 'manager' and 'management' contain the same core concept or 'stem', which in this case is 'manag'. After collating all words that contain the same stem, however, this function returns whichever of the original terms was most common in the dataset, rather than the stem itself (i.e. 'manag' might be returned as 'management'). This increases the interpretability of the resulting ordination, without affecting the data used in the topic model; although it does affect the interpretation of model results. Finally, words shorter than three characters, or that are present in <1% of articles, are automatically excluded. Topic models are calculated using the 'topicmodels' package (Grün & Hornik 2011), while the resulting ordinations are calculated from posterior model estimates using correspondence analysis (from the 'ade4' package; Dray & Dufour 2007).

Interactive visualization

The most important feature of 'revtools' is the interactive data interface, built using 'shiny' (Chang *et al.* 2016) and 'shinydashboard' (Chang & Borges Ribeiro 2017), and which is called by the function 'start_review_window' (Fig. 3). The main window (center) shows a 2D or 3D ordination in which each point represents one article (or optionally, word), and which are colored according to the highest-weighted topic for that article. On the top right is a bar chart showing the number of articles within each topic, which also serves as a key to the meaning of each topic in the main plot. If article

abstracts are available in the dataset, these are also shown in the right-hand panel. Finally, an expandable sidebar gives access to user options, including changing between article or word-based ordinations, changes to the color scheme (using the viridis family of palettes; Garnier 2017), or saving progress in a range of formats.

Figure 3: *Screenshot of the GUI for article sorting in the 'revtools' package.*

From a practical perspective, the function 'start_review_window' is designed to be as user-friendly as possible. This function accepts objects of three classes (Fig. 1): class 'bibliography' for directly imported data; class 'data.frame' if the user has merged different forms of data, or has attempted to de-duplicate the dataset; and 'review_info', a new S3 class that stores information produced by 'start_review_window' itself, allowing the user to store and reload their earlier progress. Because plots in 'revtools' are drawn with 'plotly' (Sievert *et al.* 2017), both plots are interactive; points or bars that are selected bring up further information that can be used to select or exclude a given article, word, or topic. One critical feature of this function is that all data are linked, so changes in one plot are passed to the next. This can be useful for excluding irrelevant but influential words, such as

copyright information, or for excluding entire topics if deemed irrelevant. The 'plotly' architecture also ensures that redrawing only occurs if the underlying topic model is recalculated. Basic changes such as selecting an alternate color scheme will not redraw the entire plot, a feature that is particularly important because it ensures that viewing attributes (such as zoom or view angle) are preserved as points are selected or excluded.

Rather than seeking to visualize topic model outputs in the most accurate way possible (the goal of the structurally similar 'LDAvis' package, for example; Sievert & Shirley 2015), 'revtools' focusses on helping the user to select or exclude individual points (articles, words or topics). This requires that the user is able re-run the topic models within the browser at any time. This re-calculation can be slow, and it is atypical for a 'shiny' interface to re-calculate models in this way. However, this feature is critical if the user is to properly engage with the data. For example, outlying articles may take up a large proportion of the ordination space, and recalculating the ordination after excluding this content can be highly informative of the distribution of topics within the corpus of interest. However, the default settings for topic models in 'revtools' prioritize speed over efficiency, by selecting a small number of topics (n=5) and iterations (2,000) for the initial ordination. This is adequate for a cursory investigation, but users will need to consider how to parameterize the underlying model before progressing too far, particularly if using they are excluding whole topics at once or if robust decisions are desired.

Discussion

In this paper I have presented 'revtools', a new platform for applying machine learning tools to evidence synthesis projects in the R environment. This package fills a key feature gap by allowing users to quickly engage with complex bibliographic data, and make meaningful conclusions about the information that is most relevant to them. These features can be used to summarize topics and locate relevant information within any set of article abstracts, irrespective of academic discipline.

The features provided by 'revtools' are likely to be particularly powerful when combined with those of related software packages. Following article sorting in 'revtools', for example, a user could

plausibly use features from 'metagear' (Lajeunesse 2016) to download and extract information from article PDFs, and conduct meta-analysis on the resulting data using 'metagear', 'metafor' (Viechtbauer 2010), or 'openMEE' (Wallace *et al.* 2017). However, deciding on which tools to adopt for ES projects is notoriously difficult, and potential users of 'revtools' should consider the ecosystem of available software before making a choice (see Marshall & Brereton 2015; Kohl *et al.* 2018). One important consideration is that 'revtools' assumes that improved visualization of bibliographic data will lead to increased efficiency, with little or no corresponding increase in article selection bias; but this assumption that has yet to be empirically tested. This is important because avoidance of potential bias is a key feature of systematic review methodology (Collaboration for Environmental Evidence 2013). Indeed some other software packages make greater efforts than 'revtools' to reduce potential bias, for example by providing user interfaces that deliberately hide author and journal names from view (e.g. 'metagear'; Lajeunesse 2016). In the short term, therefore, ES practitioners should consider whether the increased ease-of-use provided by 'revtools' is right for their situation, or conversely, whether a more traditional and bias-averse (but more labor-intensive) approach to systematic review is more fitting.

Finally, it is important to recognize that the field of machine learning in general – and the task of visualizing outputs form machine learning algorithms in particular – are active areas of research. Consequently, users should expect the range of algorithms and display options available in 'revtools' to increase over time. One particularly useful feature would be dynamic article recommendation, in which user choices are used to inform recommendations about potentially relevant information. This approach is already implemented within 'colandr', for example (www.colandrapp.com). In principle this is not especially difficult in R, although the manner and speed with which user decisions are used to recommend new articles can strongly affect the outcome, meaning careful testing would be required (O'Mara-Eves *et al.* 2015). A more immediate concern is that the current implementation of 'revtools' takes a long time to calculate topic models – and becomes more difficult to interpret – when many articles are included (i.e. >1000). This is ironic as it is reviews of this size where data

visualization methods become increasingly valuable. Consequently, finding new ways to calculate and present data from very large reviews will be a key focus of future research and development.

Acknowledgements

I am grateful to N. Haddaway for substantial discussion and insight regarding the design of 'revtools', and also for the bibliographic dataset shown in Fig. 2. Those discussions were supported by an International Outgoing Visiting Fellowship to MW, awarded by the Australian Research Councils' Centre of Excellence for Environmental Decisions. P. Barton provided comments that improved an earlier version of this manuscript.

References

Blei, D.M. (2012) Probabilistic topic models. *Commun. ACM*, **55,** 77-84. doi: 10.1145/2133806.2133826

Borah, R., Brown, A.W., Capers, P.L. & Kaiser, K.A. (2017) Analysis of the time and workers needed to conduct systematic reviews of medical interventions using data from the PROSPERO registry. *BMJ Open*, **7**. doi: 10.1136/bmjopen-2016-012545

Bornmann, L. & Mutz, R. (2015) Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *Journal of the Association for Information Science and Technology*, **66,** 2215-2222. doi: 10.1002/asi.23329

Chang, W. & Borges Ribeiro, B. (2017) shinydashboard: Create Dashboards with 'Shiny'. R package version 0.6.1 https://cran.r-project.org/package=shinydashboard

Chang, W., Cheng, J., Allaire, J., Xie, Y. & McPherson, J. (2016) Shiny: Web Application Framework for R. R Package version 0.14.2. https://cran.r-project.org/package=shiny

Collaboration for Environmental Evidence (2013) Guidelines for systematic review and evidence synthesis in environmental management. Version 4.2. Environmental Evidence, Bangor, United Kingdom. doi:

Cook, C.N., Nichols, S.J., Webb, J.A., Fuller, R.A. & Richards, R.M. (2017) Simplifying the selection of evidence synthesis methods to inform environmental decisions: A guide for decision makers and scientists. *Biological Conservation*, **213,** 135-145. doi: https://doi.org/10.1016/j.biocon.2017.07.004

Dicks, L., Haddaway, N., Hernández-Morcillo, M., Mattsson, B., Randall, N., Failler, P., Ferretti, J., Livoreil, B., Saarikoski, H. & Santamaria, L. (2017) Knowledge synthesis for environmental decisions: an evaluation of existing methods, and guidance for their selection, use and development: a report from the EKLIPSE project. doi:

Dray, S. & Dufour, A.B. (2007) The ade4 package: implementing the duality diagram for ecologists. *Journal of Statistical Software*, **22,** 1-20. doi: 10.18637/jss.v022.i04

Garnier, S. (2017) viridis: Default Color Maps from 'matplotlib'. R Package version 0.4.0 https://cran.r-project.org/package=viridis

Grün, B. & Hornik, K. (2011) topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, **40,** 1-30. doi: 10.18637/jss.v040.i13

Hemens, B.J. & Iorio, A. (2017) Computer-Aided Systematic Review Screening Comes of Age. *Annals of internal medicine*, **167,** 210-211. doi: 10.7326/M17-1295

Jinha, A.E. (2010) Article 50 million: an estimate of the number of scholarly articles in existence. *Learned Publishing*, **23,** 258-263. doi: 10.1087/20100308

Kass, J., Vilela, B., Aiello-Lammens, M., Muscarella, R., Merow, C. & Anderson, R.P. (2018) Wallace: a flexible platform for reproducible modeling of species niches and distributions built for community expansion. *Methods in Ecology and Evolution*. doi: 10.1111/2041-210X.12945

Kohl, C., McIntosh, E.J., Unger, S., Haddaway, N.R., Kecke, S., Schiemann, J. & Wilhelm, R. (2018) Online tools supporting the conduct and reporting of systematic reviews and systematic maps: a case study on CADIMA and review of existing tools. *Environmental Evidence*, **7,** 8. doi: 10.1186/s13750-018-0115-5

Lajeunesse, M.J. (2016) Facilitating systematic reviews, data extraction and meta-analysis with the metagear package for r. *Methods in Ecology and Evolution*, **7,** 323-330. doi: 10.1111/2041-210X.12472

Marshall, C. & Brereton, P. (2015) Systematic review toolbox: a catalogue of tools to support systematic reviews. *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, pp. 23. ACM. doi: 10.1145/2745802.2745824

McLean, M.W. (2017) RefManageR: Import and Manage BibTeX and BibLaTeX References in R. *The Journal of Open Source Software*, **2,** 338. doi: 10.21105/joss.00338

Meyer, D., Hornik, K. & Feinerer, I. (2008) Text mining infrastructure in R. *Journal of Statistical Software*, **25,** 1-54. doi: 10.18637/jss.v025.i05

Mislan, K.A.S., Heer, J.M. & White, E.P. (2016) Elevating The Status of Code in Ecology. *Trends in Ecology & Evolution*, **31,** 4-7. doi: 10.1016/j.tree.2015.11.006

O'Mara-Eves, A., Thomas, J., McNaught, J., Miwa, M. & Ananiadou, S. (2015) Using text mining for study identification in systematic reviews: a systematic review of current approaches. *Systematic Reviews*, **4,** 5. doi: 10.1186/2046-4053-4-5

Rathbone, J., Carter, M., Hoffmann, T. & Glasziou, P. (2015) Better duplicate detection for systematic reviewers: evaluation of Systematic Review Assistant-Deduplication Module. *Systematic Reviews*, **4,** 6. doi: 10.1186/2046-4053-4-6

Roll, U., Correia, R.A. & Berger-Tal, O. (2017) Using machine learning to disentangle homonyms in large text corpora. *Conservation Biology*. doi: 10.1111/cobi.13044

Sackett, D.L. (1997) Evidence-based medicine. *Seminars in perinatology*, pp. 3-5. Elsevier. doi: 10.1016/S0146-0005(97)80013-4

Sievert, C., Parmer, C., Hocking, T., Chamberlain, S., Ram, K., Corvellec, M. & Despouy, P. (2017) plotly: Create Interactive Web Graphics via 'plotly.js'. R package version 4.7.1. https://cran.r-project.org/package=plotly

Sievert, C. & Shirley, K. (2015) LDAvis: Interactive Visualization of Topic Models. R package version 0.3.2. https://cran.r-project.org/package=LDAvis

Sutherland, W.J., Pullin, A.S., Dolman, P.M. & Knight, T.M. (2004) The need for evidence-based conservation. *Trends in Ecology and Evolution*, **19,** 305-308. doi: 10.1016/j.tree.2004.03.018

Van der Loo, M.P. (2014) The stringdist package for approximate string matching. *The R Journal*, **6,** 111-122. doi:

Viechtbauer, W. (2010) Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, **36,** 1-48. doi:

Wallace, B.C., Lajeunesse, M.J., Dietz, G., Dahabreh, I.J., Trikalinos, T.A., Schmid, C.H. & Gurevitch, J. (2017) OpenMEE: Intuitive, open-source software for meta-analysis in ecology and evolutionary biology. *Methods in Ecology and Evolution*, **8,** 941-947. doi: 10.1111/2041-210X.12708

Westgate, M.J. (2018) revtools: Tools to support evidence synthesis. R package version 0.2.1 https://cran.r-project.org/package=revtools

Westgate, M.J., Haddaway, N.R., Cheng, S.H., McIntosh, E.J., Marshall, C. & Lindenmayer, D.B. (2018) Software support for environmental evidence synthesis. *Nature Ecology & Evolution*, **in press**. doi:

Westgate, M.J. & Lindenmayer, D.B. (2017) The difficulties of systematic reviews. *Conservation Biology*, **31,** 1002-1007. doi: 10.1111/cobi.12890

Young, K., Ashby, D., Boaz, A. & Grayson, L. (2002) Social Science and the Evidence-based Policy Movement. *Social Policy and Society*, **1,** 215-224. doi: 10.1017/S1474746402003068