# 1 medrxivr: an R package for systematically searching biomedical preprints

## 1.1 Main points (to be removed)

The main points I want to get across in this Chapter are:

* Why searching preprints is important
* Why existing methods are not sufficient
* What the medrxivr tool adds

## 1.2 Lay summary

Preprints are copies of academic manuscripts that are posted online in advance of being formally published by an academic journal. They represent an important source of scientific literature. A new software program called medrxivr was created to allow researchers to find preprints related to their research in a transparent and reproducible way. Development of this tool was an essential part of this thesis, as preprints represent a key source of information needed for the research reported in future chapters.

## 1.3 Introduction

Preprints represent an increasingly important source of scientific information. Defined by the Committee on Publication Ethics (COPE) as ‘scholarly manuscript[s] posted by the author(s) in an openly accessible platform, usually before or in parallel with the peer review process’1, preprints serve several purposes. They are used to establish primacy when submitting to a journal where the peer-review process may take several months;2 to rapidly disseminate research findings, as occurred during the COVID-19 pandemic;3 and to make available publications that may not have been accepted elsewhere in an attempt to combat publication bias or the “file-drawer” effect.4

As a result, repositories of preprinted articles should be considered a distinct but complementary information source when reviewing the evidence base as part of a systematic review. The two key repositories in the health science are bioRxiv, established in 2013,5 and medRxiv, which evolved to replace the “Epidemiology” and “Clinical Trial” categories of bioRxiv, which launched in 2019.6

Searching these preprints as part of the systematic review described in Chapter ?? was a necessity, as many of the existing reviews on the topic of lipids and dementia have not considered this important source of grey literature. At the time of writing, however, the bioRxiv.medRxiv websites allow only simple search queries as opposed to the often complex Boolean logic (AND/OR/NOT) that information specialists use to query other major databases.[@bramer2018a;@gusenbauer2020] Additionally, the best available extraction mechanism for obtaining references for all records returned by a search were to go through each record, one-by-one, downloading individual citations. As the scale of these preprint databases increase, particularly in light of the massive expansion of the medRxiv repository as a result of COVID, this already time-consuming and error-prone method is no longer feasible.

This chapter outlines the development and key functionality of medrxivr (version 0.0.5), a tool created to facilitate the searching of medRxiv and bioRxiv preprints. The factors that necessitated the development of this tool in the context of this thesis are outlined, and the use of medrxivr in external projects and by other researchers is discussed. As the majority of work on this aspect of this thesis is represented by lines of code, this Chapter is a high-level summary of the work done on this project. The GitHub repository[[1]](#footnote-22) for the medrxivr contains a complete record of the development of this tool, including discussion with other members of the systematic review community.



Figure 1.1: **Role of medrxivr in a systematic review workflow:** medrxivr allows for systematic searching of biomedical preprints as part of the initial literature searching. Following title and abstract screening, reviewers can then programmatically retrieve a copy of the PDF of included records to facilitate the full-text screening stage (similar to Endnote’s “Find Full Text” feature).

## 1.4 Development

### 1.4.1 Success criteria

The tool was developed to meet three success criteria,7 influenced both by the functionality required to perform systematic searches as part of the review in Chapter ??, discussion with information specialist colleagues, and an informal survey of the evidence synthesis and health librarian communities on Twitter. The criteria were as follows:

1. reliable, reproducible and transparent search functionality, allowing for Boolean (AND/OR/NOT) operator logic;
2. support for bulk export of references returned by the search to a file type that can be readily imported into a reference manager (e.g., *.bib* or *.ris*); and
3. automated retrieval of the full-text PDFs of relevant records, similar to the Find Full Text feature offered by EndNote.

### 1.4.2 Alternative medRxiv/bioRxiv interfaces

Prior development of this tool, an audit of existing tools for accessing medRxiv and bioRxiv metadata was conducted. While none address the success criteria described above, two of these tools are useful to consider to highlight the additional functionality that medrxivr contributes.

The first, a platform called Rxivist,8 allows users to search preprints using keywords. However, the core functionality of the Rxivist platform is focused around exploring the number of times a preprint has been downloaded and/or shared on Twitter, to allow researchers to find the most popular papers related to their topic. The search interface[[2]](#footnote-27) does not allow for complex search strategies using Boolean operators and there is no option to batch-export the results of a search.

The second tool, search.bioPreprint, allows users to search for terms across a range of preprint servers, including medRxiv and bioRxiv, but also journals which use a post-publication peer-review process such as F1000Research.9 However, similar to the Rxivist platform, this tool is designed for researchers aiming to keep up to date with recent developments in their fields rather than systematically assess the entirety of the available literature. As such, the platform only returns the most recent 1,000 records by publication date.

Finally, neither tool provides an easy way to programmatically download a copy of the PDF of relevant preprints as part of the preparation for the full-text screening stage of a systematic review.

### 1.4.3 Early versions

Work on the medrxivr tool began in Summer 2019, and initially consisted of a development of set of R scripts to allow for searching medRxiv and bioRxiv as part of the systematic search outlined in Chapter ??. Following interest from other researchers in using the *ad-hoc* web-scraping scripts, additional development work took place in 2019/2020, allowing for improved searching and exporting functionality and the initial version of the medrxivr R package was released in February 2020.

Early versions of the tool had a reliance on scraping data directly from the repository website. Web-scraping is a fragile mechanism for extracting data, as it is entirely dependent on consistent website design and underlying code structure remaining unchanged.10,11. In the case of medrxivr, constant maintenance work was required to ensure the web-scraping script performed as expected, as the repository website was regularly updated.

However, an Application Programming Interface (API) for the medRxiv and bioRxiv repositories was made public in early 2020 by the institution responsible for managing these preprint repositories, the Cold Springs Harbor Laboratory. This allowed for newer versions of the medrxivr package to engage in active “fault prevention” and provide a more robust interface to the data by removing the reliance of web-scraping.11

### 1.4.4 Package infrastructure

The medrxvir package was written in R using RStudio,12 and followed development best-practice, including detailed documentation, a robust unit testing framework (99% of all code lines within the package are formally tested across multiple platforms including Windows, MacOS, and Linux), and in-depth code review by two experienced, independent reviewers.

## 1.5 Usage

The medrxivr R package is split into two component parts:

* an interface to the Cold Springs Harbor Laboratory API, which imports medRxiv and bioRxiv metadata into R; and
* a collection of functions for working with the imported metadata, with an explicit focus on searching this data as part of a systematic review or evidence synthesis project.

The standard workflow is to download a copy of all metadata contained in the repository, and then to perform searches on this local copy. This is a workaround as the Cold Springs Harbor Laboratory API does not provide any functionality to search the database.

While the package allows for users to search both medRxiv and bioRxiv, as the process is identical for both, the examples in the sections are limited to the medRxiv repository.

### 1.5.1 Installation

medrxivr has been released to the Comprehensive R Archive Network (CRAN), and can be installed with the following code:

install.packages("medrxivr")

Alternatively, the development version of the package can be installed from GitHub:

# install.packages("devtools")   
devtools::install\_github("ropensci/medrxivr")

### 1.5.2 Importing preprint metadata

medrixvr provides two ways to access medRxiv data. The first, via the mx\_api\_content() function, creates a local copy of all data available from the medRxiv API at the time the function is run.

# Get a copy of the database from the live medRxiv API endpoint  
mx\_data <- mx\_api\_content()

The second, via the mx\_snapshot() function, provides access to a maintained static snapshot of the database, created each morning at 6am using mx\_api\_content(). Allowing users of medrxivr to access a maintained snapshot removes any dependency on the API, which can become unavailable during peak usage times. The relationship between the two methods for accessing the data contained in the medRxiv database is summarized in Figure 1.2.

# Import a copy of the medRxiv data from the snapshot  
mx\_data <- mx\_snapshot()



Figure 1.2: **Overview of medrxivr data sources:** Users can either access the API directly via mx-api\_content(), or can import a maintained snapshot of the database, taken each morning at 6am, via the mx\_snapshot() function. Note: due to the size of bioRxiv, only a maintained snapshot of the medRxiv repository is available via mx\_snapshot().

### 1.5.3 Performing a search

Once a local copy of the metadata is created, the first step in searching it is to create a search strategy. Search terms to be combined with the OR operator are contained in vectors (c(...)), while topics to be combined with the AND operator are contained in lists (list(…)).

# Create the search query  
topic1 <- c("dementia","alzheimer's") # Combined with OR  
topic2 <- c("lipids","statins") # Combined with OR  
  
myquery <- list(topic1, topic2) # Combined with AND

For example, when written in standard syntax, the search contained in the myquery object above would be: “((dementia **OR** alzheimer’s) **AND** (lipids **OR** statins))”. There is now limit to the number of search terms that can be included in each topic, nor in the number of topics that can be search for. Search terms can also contain common syntax used by systematic reviewers and health librarians, including the use of NEAR statements which allows for identification of co-localised terms, and wild-cards, which allow for alternate spellings, e.g. “randomi*s*ation” vs “randomi*z*ation”.

Once a strategy has been defined, it is passed along with the local copy of the database to the mx\_search() function.

# Run the search  
results <- mx\_search(mx\_data,  
 myquery)

### 1.5.4 Refining your search

An important argument of the mx\_search() is report, which outputs a structured table with each search strategy presented on an individual line and the number of records associated with this strategy.

results <- mx\_search(mx\_data,  
 c("dementia", "alzheimer's"), # Combined with OR  
 report = TRUE)

## Found 203 record(s) matching your search.  
## Total topic 1 records: 203  
## dementia: 203  
## Total topic 2 records: 0  
## alzheimer’s: 0

This allows users to discover which terms in their search are contributing most to the total number of results returned. This is important as part of developing a search strategy,13 as it allows for the key terms related to each topic to be discovered. It also aids in identifying misspelled search terms, which will frequently return no results.

### 1.5.5 Exporting to a bibliography file

In line with the second success criteria (Section 1.4.1), one of the key features of the medrxivr is the ability for users to easily export the results of their systematic search to a reference manager. While it is a seemingly simple request, this is is one of the key ways in which medrxivr is set apart for other preprint search tools, including the native medRxiv/bioRxiv website search functionality.

For example, the results of our simple search above can be exported to the "medrxiv\_export.bib" file using the following code:

mx\_export(results,   
 file = "medrxiv\_export.bib",  
 report = TRUE)

### 1.5.6 Downloading the PDFs of relevant records

medrxivr alos allows users to download the full text papers for records that are deemed eligible for full-text screening (see Figure 1.1). mx\_download() takes the list of included records and saves the PDF for each to a folder specified by the user. This functionality is similar to the “Find Full Text” feature offered by EndNote.

mx\_download(results, # Search results, less excluded records  
 "pdf/") # Directory to save PDFs to

## 1.6 Discussion

### 1.6.1 Reception and future plans

The tool has be well received by the community (as of March 2021, medrxivr has been downloaded more than 2400 times), and several use cases have been reported. It has been used to visualize the growing number of preprints related to the 2019 coronavirus outbreak,[[3]](#footnote-41) perform searches of preprints as part of systematic reviews,14 and examine how data-sharing behaviour is affected by different publication policies (preprint server versus journal).[CITE]

Following rigorous peer-review, it has been onboarded into the rOpenSci suite of packages, a collection of “carefully vetted, staff- and community-contributed R software tools that lower barriers to working with scientific data sources on the web”,[CITE] and an associated article published in the Journal of Open Source Software.15 The entire review discussion is publicly available and can be viewed online.[[4]](#footnote-43) The tool has also been well received by the open-source community, demonstrated by the engagement of other developers in contributing to important new functionality and suggesting bug-fixes.

[Add bit about the benefits of the rOpenSci review process]

Lobbying of the Cold Springs Harbor Laboratory to develop the API to allow for direct searching of the database has been ongoing. This would negate the current need to download a local copy of the relevant preprint database before searching it, which is currently the rate limiting step for performing searches. For example, as of January 2021, downloading a copy of the bioRxiv database takes approximately an hour.

### 1.6.2 Limitations of medrxivr

While searching of the medRxiv and bioRxiv databases was crucial for the systematic review element of this thesis presented in Chapter ??, there are some important limitations to note here. A key example is that the tool only searches the available metadata of preprint records (the title, abstract and keywords), rather than the full text of preprints, meaning some relevant records might be missed. However, this approach ethos other search platforms such as OvidSP, and while some relevant records may be missed (reduced sensitivity), limiting the search to the metadata fields prevents non-relevant records from being returned (high specificity). A key example of the reduced specificity when searching the full text, identified during development of medrxivr, is that a search for “dementia” would return a record where the only occurrence of this term is in the title of one of the references.16

There is also the potential that the cross-section of literature posted on medrxiv/bioRxiv would be substantially different from the true grey literature (studies or analyses that are not published for a range of reasons including results that are not deemed “novel” or are not statistically significant).17 This is because simply lowering the barriers to publication may well encourage authors to published “null” results, but due to the effort involved in writing up a distributable manuscript, it is unlikely to completely address the “file drawer” effect.4 It is likely too early (and likely too methodologically difficult) to tell whether the increased popularity and acceptance of preprint repositories will have any effect of the availability of research that was not considered “publishable” at other venues. (This disagress with the use cases listed in the intro!)

### 1.6.3 Role of open source tools in evidence synthesis

*Not sure where this should go, but keen to include it*

Part of the motivation for creating the medrxivr tool was a belief that the development and distribution of open source scripts and tools should be a fundamental part of evidence synthesis research.18,19 In the case of medrxivr, it is likely that several other evidence synthesists had written personal scripts that have a similar, or related, functionality - in fact, following development of the tool, I identified one other researcher that has done so (Nicholas Fraser, author of the rbiorxiv package, which allows for importing medRxiv metadata into R but does not provide search functionality).20 If these scripts continue to be developed in private and are never shared or publicised, this will inevitably lead to harm to the evidence synthesis community, not only in terms of duplication of time and effort but also due to lost opportunities for collaboration.19 Creating and sharing well-documented packages, the recognised standard for sharing code in R, represents one way to reduce this inefficiency.21

## 1.7 Summary

* In this Chapter, I have introduced a new tool, medrxivr, for performing complex systematic searches of the medRxiv and bioRxiv preprint repositories.
* I have outlined the motivation for developing this tool in relation to this thesis - more specifically, that it was used to perform systematic and reproducible searches of a key literature sources used in the comprehensive systematic review described in Chapter ??.
* I have contrasted medrxivr with other available interfaces to medRxiv/bioRxiv data to highlight the added functionality it offers. I have also discussed the tools reception to date, a roadmap for its future development and its place in the broader evidence synthesis in R ecosystem.

1. Committee on Publication Ethics (COPE). *Discussion Document: Preprints*. (2018).

2. Vale, R. D. & Hyman, A. A. Priority of discovery in the life sciences. *eLife* **5**, e16931 (2016).

3. Fraser, N. *et al.* Preprinting a pandemic: The role of preprints in the COVID-19 pandemic. *bioRxiv* 2020.05.22.111294 (2020). doi:[10.1101/2020.05.22.111294](https://doi.org/10.1101/2020.05.22.111294)

4. Rosenthal, R. The file drawer problem and tolerance for null results. *Psychological Bulletin* **86**, 638–641 (1979).

5. Sever, R. *et al.* *bioRxiv: The preprint server for biology*. (Scientific Communication and Education, 2019). doi:[10.1101/833400](https://doi.org/10.1101/833400)

6. Rawlinson, C. & Bloom, T. New preprint server for medical research. *BMJ* **365**, (2019).

7. Wateridge, J. IT projects: A basis for success. *International Journal of Project Management* **13**, 169–172 (1995).

8. Abdill, R. J. & Blekhman, R. Rxivist.org: Sorting biology preprints using social media and readership metrics. *PLOS Biology* **17**, e3000269 (2019).

9. Iwema, C. L., LaDue, J., Zack, A. & Chattopadhyay, A. Search.bioPreprint: A discovery tool for cutting edge, preprint biomedical research articles. *F1000Research* **5**, 1396 (2016).

10. Shaw, M. "Self-healing": Softening precision to avoid brittleness: Position paper for WOSS ’02: Workshop on self-healing systems. *Proceedings of the first workshop on Self-healing systems* 111–114 (2002). doi:[10.1145/582128.582152](https://doi.org/10.1145/582128.582152)

11. Laprie, J. C. Dependability: Basic Concepts and Terminology. in *Dependability: Basic Concepts and Terminology: In English, French, German, Italian and Japanese* (ed. Laprie, J. C.) 3–245 (Springer, 1992).

12. R Core Team. *R: A language and environment for statistical computing*. (R Foundation for Statistical Computing, 2019).

13. Bramer, W. M., de Jonge, G. B., Rethlefsen, M. L., Mast, F. & Kleijnen, J. A systematic approach to searching: An efficient and complete method to develop literature searches. *Journal of the Medical Library Association : JMLA* **106**, 531–541 (2018).

14. Noone, C. *et al.* Investigating and evaluating evidence of the behavioural determinants of adherence to social distancing measures A protocol for a scoping review of COVID-19 research. *HRB Open Research* **3**, 46 (2020).

15. McGuinness, L. & Schmidt, L. Medrxivr: Accessing and searching medRxiv and bioRxiv preprint data in R. *Journal of Open Source Software* **5**, 2651 (2020).

16. Bong, S. M., McKay, J. L., Factor, S. A. & Ting, L. H. *Perception of whole-body motion during balance perturbations is impaired in Parkinson’s disease and is associated with balance impairment*. (Neurology, 2019). doi:[10.1101/19000265](https://doi.org/10.1101/19000265)

17. Song, F. *et al.* Dissemination and publication of research findings: An updated review of related biases. *Health Technology Assessment* **14**, (2010).

18. Goldacre, B., Morton, C. E. & DeVito, N. J. Why researchers should share their analytic code. *BMJ* l6365 (2019). doi:[10.1136/bmj.l6365](https://doi.org/10.1136/bmj.l6365)

19. McKiernan, E. C. *et al.* How open science helps researchers succeed. *eLife* **5**, e16800 (2016).

20. Fraser, N. *Rbiorxiv: Client for the ’bioRxiv’ API*. (2020).

21. Vuorre, M. & Crump, M. J. C. Sharing and organizing research products as R packages. *Behavior Research Methods* (2020). doi:[10.3758/s13428-020-01436-x](https://doi.org/10.3758/s13428-020-01436-x)

1. Available at <https://github.com/ropensci/medrxivr> [↑](#footnote-ref-22)
2. Available at <https://rxivist.org/> [↑](#footnote-ref-27)
3. <https://twitter.com/L_Brierley/status/1233109086444695553> [↑](#footnote-ref-41)
4. <https://github.com/ropensci/software-review/issues/380> [↑](#footnote-ref-43)