Why are open source statistical programming languages the best?

Because they R.

1

medrxivr: an R package for searching medRxiv and bioRxiv preprint data

1.1 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 as part of this thesis to allow researchers to find preprints related to their research in a transparent and reproducible way. Development of this tool was required as part of this thesis, as preprints represent a key source of information needed for the research reported in future chapters.

1.2 Introduction

Preprints represent an increasingly important source of scientific information. Authors can use preprint servers for several purposes - to establish primacy when submitting to a journal where a the peer-review process may take several months; to post complete manuscripts that will never be formally published,[CITE] such as those reporting updated case reports during the COVID-19 pandemic;[CITE]

and to make available publications that may not have been accepted elsewhere in an attempt to combat publication bias, or the "file-drawer" effect.

As a result, preprint repositories should be considered an 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,[2] and medRxiv, which evolved to replace the "Epidemiology" and "Clinical Trial" categories of bioRxiv, which launched in 2019.[3]

Preprints represent a particularly important to this thesis, ias one of the primary types of evidence that I planned to use in the triangulation exercise (Mendelian randomization (MR)) is still developing and many MR studies that look at the clinical question that this project aims to investigate are posted as preprints on the bioRxiv, and more recently medRxiv, preprint servers. Being able to systematically search these records for the purpose of the systematic review described in Chapter **??** was a necessity. In light of the above, the `medrxivr` R package was created in order to facilitate the searching of this data source.

More generally, developing new tools for facilitating evidence synthesis will benefit patients by speeding up the speed with which systematic reviews are produced. A recent study put the mean duration of a systematic review project, from date of registration to date of publication is 67.3 weeks.[4] A large proportion of this time will be spent on manual tasks such as running searches and extracting records, title and abstract screening, and data extraction - therefore, efforts to increase the efficiency of the process should focus on process that can be automated.

In addition, we need to focus on developing tools that allow systematic, reproducible access to more informal sources of scientific material. One key emerging source

Searching pre-print repositories is becoming an increasingly important part of a systematic review. Preprints - unpublished versions of manuscripts, frequently uploaded to a repository at the same time they are submitted to a journal for peer review - represent an important source of grey literature. As the barriers and time to publication are both lower, they also represent an important source of information that may be either currently tied up in the peer-review process

At present, medRxiv allows only simple search queries, as opposed to the often complex Boolean logic that information specialists use to query other major databases. Additionally, record metadata (titles/abstracts/author lists) must be accessed individually, rather than in batches, meaning that downloading relevant records for title and abstract screening a time consuming task.

This chapter outlines the development and key functionality of `medrxivr (version 0.2)`. The motivating factors that necessitated the development of this tool as part of this thesis are discussed. 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. The GitHub repository for the `medrxivr` contains a complete record of the development of this tool, including discussion with other members of the systematic review community.

Current extraction mechanisms for extracting the results of searches from medRxiv are to go through each record, one-by-one, downloading individual citations. As the scale of the medRxiv database increases, particularly in light of the massive expansion as a result of COVID, this alreay time-consuming and error-prone method is not longer feasible.

1.3 Development

Work on this element began in Summer 2019, and initially existed as a collection of scripts built to allow for searching medRxiv and bioRxiv as part of the systematic search outline in Chapter **??**. Following interest from other researchers in using the ad-hoc web-scraping scripts, the initial version of the `medrxivr` package was released in February 2020. Additional development work took place, allowing for improved searching and exporting function

The updates to the systematic search to capture new literature published since the last search was performed with the fully developed package.

Early versions of the tool had a reliance on web-scraping data directly from the medRxiv website. Web-scraping is a fragile, or brittle, way to extract data, as it is

entirely dependent on consistent website design and underlying code structure remaining unchanged.[5,6].

However, an Application Programming Interface (API) for the medRxiv and bioRxiv repositories was released in early 2020, allowing for a newer version of the medrxivr package to engage in "fault prevention" and work towards a more robust interface with the available data.

Finally, while initial versions of the tool focused primarily on the medRxiv repository, discovery of some eligible required expansion to the bioRxiv repository, so that both can now be searched from a single tool. However, the introductory examples given below apply are taken from searches performed in the medRxiv repository.

Developed to meet three criteria:

1. reproducible and transparent search functionality, with Boolean operator logic;

2. support for bulk export of references returned by the search;

3. automated access full-text records of relevant records.

1.4 Installation

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

To install the stable `medrxivr` from CRAN:

```
install.packages("medrxivr")
```

Alternatively, the install the development version from GitHub:

```
# install.packages("devtools")
devtools::install_github("ropensci/medrxivr")
```

The `medrxivr` R package is split into two component parts:

- An interface to the Cold Springs Harbor Laboratory API, which allows the wholesale import of the medRxiv and bioRxiv metadata
- 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 the preprint repository metadata, and then run your searchers locally. This is a workaround as the Cold Springs Harbor Labratory API does not provide any functionality to search the database.

1.5 Importing preprint meta data

`medrixvr` provides two ways to access medRxiv data.

`mx_api_content(server = "medrxiv")` 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
preprint_data <- mx_api_content()
```

As an alternative to downloading a copy fo the database from the API in realtime, the `mx_snapshot()` function provides access to a maintained static snapshot of the database. The snapshot is created each morning at 6am. This method, which doet not rely on the API, was created as during development e unavailable during peak usage times) and has the additional advantage of being faster, as it reads data from a comma-seperated file rather than having to re-extract it from the API.

The relationship between the two methods for accessing the data conatined in the medRxiv database is summarized in the figure below:

1.6 Creating a search

Once a local copy of the database has been created, the functions in the `medrxivr` package then facilitate users in working with this dataset. There are two main functions and a helper function:

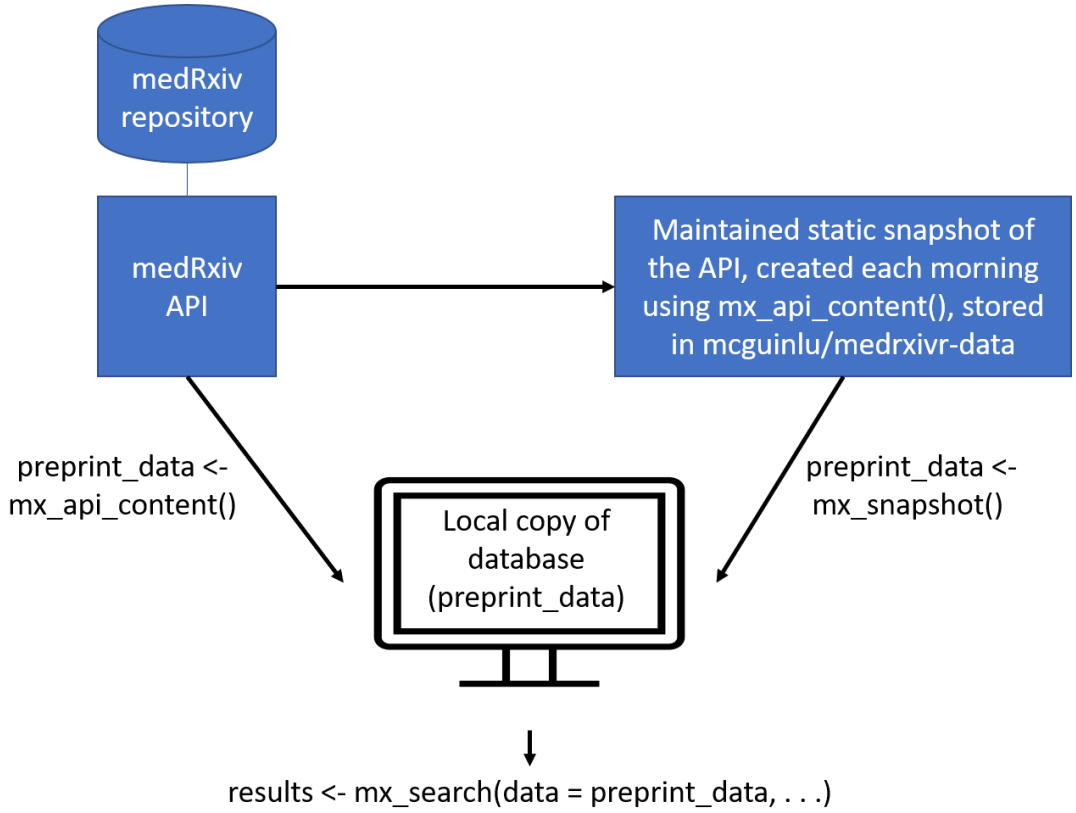

Figure 1.1: Overview of `medrxivr` data sources: Users can either access the API directly via `mx-api_content()`, or can import a maintained snapshot of the database, taken each morning at 6am, via the `mx_snapshot()` function. Note: due to the size of bioRxiv, a maintained snapshot of medrxiv is available via `mx_snapshot()`.

- `mx_search()`: Enables users to search the preprint data, using regular expressions and Boolean logic.

```
topic1  <- c("dementia","alzheimer's")  # Combined with OR
topic2  <- c("lipids","statins")        # Combined with OR

myquery <- list(topic1, topic2)         # Combined with AND

results <- mx_search(myquery)
```

Additional functionality allowing common syntax used by systematic reviewers and health librarians, including the use of NEAR statements (which allows for),

1.7 Further functionality

1.7.1 Exporting to bibliography

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` - as it was built with systematic reviews specifically in mind, it is

For example, the results of our simple search above can be exported using the following code:

```
mx_export(results,
          file = "medrxiv_export.bib")
```

1.7.2 PDF download

A second key use of the medrxivr package:

- `mx_download()` Takes the output from `mx_search()` and retrieves the full text PDF for each record, saving it to a folder specified by the user.

```
mx_download(results,        # Object returned by mx_search()
            "pdf/")         # Directory to save PDFs to
```

1.7.3 Search reporter

One additional function creates a formatted output table with each search strategy presented on an individual line and the number of records associated with this strategy.

```
mx_report(results)
```

This allows users to discover which aspects of their

1.7.4 Data visualisation

1.7.5 Shiny app

Part of the key theme of accessibility meant creating a web-application that allowed those without any knowledge of the R programming environment to benefit from it's functionality. The app is readily available and has

1.8 Reception and future plans

The tool has be well received by the community (as of December 2020, `medrxivr` has been downloaded more than 1000 times, and the medRxivr app has been visited more than ????), and several use cases have been reported. It has been used to visualize the growing number of preprints related to the 2019 coronavirus outbreak,[1] perform systematic searches in a number of other systematic reviews,[7] in addition to forming a platform for an analysis of researcher's differing data-sharing tendencies under two different journal models (preprint server vs formal publication).[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", and an associated article published in the Journal of Open Source Software.[8] The entire review discussion is publicly available and can be viewed online.[2]

Lobbying of the cold springs harbour labratory has been ongoing. This would negate the current need to download a full copy of the relevant preprint database before searching it locally, which is currently the rate limiting step for.

1.9 Comparison with other search options

During development of the package, there were several alternative considered as part of a audit of existing tools. However, development of a new custom tool was preferred as none met the three criteria required: 1) reproducible and transparent search

[1] `https://twitter.com/L_Brierley/status/1233109086444695553`
[2] `https://github.com/ropensci/software-review/issues/380`

functionality, with Boolean operator logic; 2) support for bulk export of references returned by the search; 3) automated access full-text records of relevant records.
Some previous tools exist while allowed for robust searching of preprint repositories, such as `search.bioPreprint`[9]- however, these tools were aimed more at those looking to keep up to date with recent developments rather than systematically assess the entirety . As such, they did not support needed functionality such as bulk export of records matching a search term, unlimited return of records (c.f. `search.bioPreprint` which is limited to the most recent 1000 records matching a user search).

1.10 Package infrastructure

The `medrxvir` package was written in R using RStudio, and followed development best practices, including complete and information documentation, a robust unit testing framework (99% of all code lines within the package are formally tested under this framework) across multiple platforms including Windows, MacOS, and Linux, and in-depth code review by two experienced reviewers.
The medRxiv snapshot is still taken every morning using GitHub Actions, an automated system for repetitive tasks.
An automatically generated documentation website is also generated on each update to the code-base[3]

1.11 Discussion

Packaging and sharing R scripts should be a fundamental part of evidence synthesis process. []
While searching of the medRxiv database was crucical for the systematic review element of this thesis presented in Chapter **??**
Need to also touch on the biases involved in search preprint literature, in that the authors of those in the

[3] `https://docs.ropensci.org/medrxivr/`

it is too early to see if preprint reps

There is the potential that the cross-section of literature posted on medrxiv would be substantially different from the true grey literature - simply lowering the barriers to publication may encourage authors to published "null" results,[CITE] but due to the effort involved in writing up a distributable manuscript, it is unlikely to completely address the "file drawer" effect.[CITE]

As

By implementing the tool described above as both as an R package and a `Shiny` web app, the functionality is available to evidence synthesists with varying levels of ability in R. These tools serve as an example of the advantages of "packaging" the R scripts that evidence synthesists often create for personal use.[10] In the case of `medrxivr`, it is likely that several other evidence synthesists had written scripts that have a similar functionality - in fact, in the course of its development, one other researcher that has done so was identified (xxxx xxxx, author of the `rbiorxiv` pacakge).[???] This duplication of time and effort is inefficient, and creating and sharing well-documented R packages represents one way to reduce this inefficiency. Taking this approach one step further, `Shiny` apps represent a straightforward way to provide a user-friendly GUI for a newly created R package within a very short time-frame, expanding the potential pool of users of the package to anyone with an internet connection.

Creating a package using R has a number of advantages unique to the R programming environment. R provides access to a range of powerful tools including the `ggplot` infrastructure for creating publication-quality plots, and RMarkdown, which enables creation of documents that can be rendered in a range of formats such as PDF, HTML, or Word.[11] Furthermore, and focusing specifically on evidence synthesis, building new tools as packages in R allows for easy integration with the range of existing evidence synthesis packages. Recently, the `metaverse` project,[12] of `medrivr` is a part, has begun to curate a collection of R packages that cover different aspects of the systematic review and meta-analysis process which, when taken together, form a coherent end-to-end open-source alternative to commercial offerings such

as Covidence or Review Manager. Key offerings in this suite of packages include `litsearcher`, which facilitates systematic search strategy development, `revtools`, a package for managing the review process and performing title and abstract screening, `metaDigitise`, a package for automatic extraction of data from figures in research papers, and `metafor`, a package for conducting meta-analyses in R.[13–16]

1.12 Summary

- In this Chapter, I have introduced a new tool, `medrxivr`, for performing complex searches in 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 source used in the comprehensive systematic review described in Chapter **??**.
- The impact of this tool to date, its place in the broader evidence synthesis in R ecosystem, and a roadmap for its future development has been discussed.

test

1. Bealy, C. R Programming Humour. *Stack Exchange* (2013).

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

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

4. Borah, R., Brown, A. W., Capers, P. L. & Kaiser, K. A. Analysis of the time and workers needed to conduct systematic reviews of medical interventions using data from the PROSPERO registry. *BMJ Open* **7**, e012545 (2017).

5. Shaw, M. "Self-healing": Softening precision to avoid brittleness: Position paper for WOSS '02: Workshop on self-healing systems. in *Proceedings of the first workshop on Self-healing systems* 111–114 (Association for Computing Machinery, 2002). doi:10.1145/582128.582152

6. 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). doi:10.1007/978-3-7091-9170-5_1

7. 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).

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

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. Wickham, H. *R packages: Organize, test, document, and share your code.* (O'Reilly Media, Inc., 2015).

11. Xie, Y., Allaire, J. J. & Grolemund, G. *R markdown: The definitive guide.* (Chapman and Hall/CRC, 2018).

12. Various Authors. Metaverse: An R ecosystem for meta-research. (2020).

13. Grames, E. M., Stillman, A. N., Tingley, M. W. & Elphick, C. S. An automated approach to identifying search terms for systematic reviews using keyword co-occurrence networks. *Methods in Ecology and Evolution* **10**, 1645–1654 (2019).

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

15. Pick, J. L., Nakagawa, S. & Noble, D. W. A. Reproducible, flexible and high-throughput data extraction from primary literature: The metaDigitise R package. *Methods in Ecology and Evolution* **10**, 426–431 (2018).

16. Westgate, M. J. Revtools: An R package to support article screening for evidence synthesis. *Research Synthesis Methods* **10**, 606–614 (2019).