

# PRISMA2020: An R package and Shiny app for producing PRISMA 2020-compliant flow diagrams, with interactivity for optimised digital transparency and Open Synthesis

Neal R. Haddaway<sup>1,2</sup> | Matthew J. Page<sup>3</sup> | Chris C. Pritchard<sup>4</sup> | Luke A. McGuinness<sup>5</sup>

<sup>1</sup>Leibniz-Centre for Agricultural Landscape Research (ZALF), Müncheberg, Germany

<sup>2</sup>Africa Centre for Evidence, University of Johannesburg, Johannesburg, South Africa

<sup>3</sup>School of Public Health and Preventive Medicine, Monash University, Melbourne, Australia

<sup>4</sup>Institute of Health and Allied Professions, Nottingham Trent University, Nottingham, UK

<sup>5</sup>Department of Population Health Sciences, University of Bristol, Bristol, UK

## Correspondence

Neal R. Haddaway, Leibniz-Centre for Agricultural Landscape Research (ZALF), Müncheberg, Germany.  
Email: [neal\\_haddaway@hotmail.com](mailto:neal_haddaway@hotmail.com)

## Abstract

**Background:** Reporting standards, such as PRISMA aim to ensure that the methods and results of systematic reviews are described in sufficient detail to allow full transparency. Flow diagrams in evidence syntheses allow the reader to rapidly understand the core procedures used in a review and examine the attrition of irrelevant records throughout the review process. Recent research suggests that use of flow diagrams in systematic reviews is poor and of low quality and called for standardised templates to facilitate better reporting in flow diagrams. The increasing options for interactivity provided by the Internet gives us an opportunity to support easy-to-use evidence synthesis tools, and here we report on the development of a tool for the production of PRISMA 2020-compliant systematic review flow diagrams.

**Methods and Findings:** We developed a free-to-use, Open Source R package and web-based Shiny app to allow users to design PRISMA flow diagrams for their own systematic reviews. Our tool allows users to produce standardised visualisations that transparently document the methods and results of a systematic review process in a variety of formats. In addition, we provide the opportunity to produce interactive, web-based flow diagrams (exported as HTML files), that allow readers to click on boxes of the diagram and navigate to further details on methods, results or data files. We provide an interactive example here; <https://prisma-flowdiagram.github.io/>.

**Conclusions:** We have developed a user-friendly tool for producing PRISMA 2020-compliant flow diagrams for users with coding experience and, importantly, for users without prior experience in coding by making use of Shiny ([https://estech.shinyapps.io/prisma\\_flowdiagram/](https://estech.shinyapps.io/prisma_flowdiagram/)). This free-to-use tool will make it easier to produce clear and PRISMA 2020-compliant systematic review flow diagrams. Significantly, users can also produce interactive flow diagrams for the first time, allowing readers of their reviews to smoothly and swiftly explore and navigate to further details of the methods and results of a review. We believe this tool will increase use of PRISMA flow diagrams, improve the compliance and quality of flow diagrams, and facilitate strong science communication of the methods and results of systematic reviews by

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2022 The Authors. *Campbell Systematic Reviews* published by John Wiley & Sons Ltd on behalf of The Campbell Collaboration.

making use of interactivity. We encourage the systematic review community to make use of the tool, and provide feedback to streamline and improve their usability and efficiency.

## 1 | INTRODUCTION

### 1.1 | Evidence synthesis reporting standards

Evidence syntheses (e.g., systematic reviews and evidence maps) typically aim to reliably synthesise an evidence base, and are based on state-of-the-art methodologies designed to maximise comprehensiveness (or representativeness), procedural objectivity, and reproducibility, whilst minimising subjectivity and risk of bias (Collaboration for Environmental Evidence, 2018; Higgins et al., 2019). Reproducibility is made possible through a high degree of transparency when reporting the planned or final methods used in a review protocol or final report. Reporting standards, such as PRISMA (Preferred Reporting Items for Systematic reviews and Meta-Analyses [Moher et al., 2009]) and ROSES (RepOrting standards for Systematic Evidence Syntheses [Haddaway, Macura, et al., 2018]), aim to ensure that review methods and findings are described in sufficient detail.

In 2009, the PRISMA statement—a reporting guideline designed primarily for systematic reviews of health interventions—was released (Liberati et al., 2009; Moher et al., 2009). The guideline was developed by a consortium of systematic reviewers, methodologists and journal editors to address evidence of incomplete reporting in systematic reviews (Moher et al., 2007), with recommendations formed largely based on expert consensus obtained via Delphi surveys and consensus meetings. The PRISMA statement has been widely endorsed and adopted by journals, and evidence suggests use of the guideline is associated with more complete reporting of systematic reviews (Page et al., 2016). However, to address the many innovations in methods for systematic reviews, changes in terminology, and new options to disseminate research evidence that have occurred since 2009, an update to the guideline (referred to now as PRISMA 2020 [Page et al., 2021]) has recently occurred.

### 1.2 | Review flow diagrams

Flow diagrams in evidence syntheses allow the reader to rapidly understand the core procedures used in a review and examine the attrition of irrelevant records throughout the review process. The PRISMA flow diagram, published in 2009, is a static infographic that describes the sources, numbers and fates of all identified and screened records in a review (for more details, see the original flow diagram (Moher et al., 2009) and an update from 2014 [Stovold et al., 2014]). A recent assessment of the quality and use of flow diagrams in systematic reviews found that only 50% of identified reviews made use of flow diagrams, with their quality generally being

low and not significantly improving over time (quality defined by the presence of critical data on the flow of studies through a review [Vu-Ngoc et al., 2018]): as a result, the authors called for a standardised flow diagram template to improve reporting quality.

Several changes were made to the original PRISMA flow diagram in the 2020 update (Page et al., 2021). The 2020 template: (i) recommends authors specify how many records were excluded before screening (e.g., because they were duplicate records that were removed, or marked as ineligible by automation tools); (ii) recommends authors specify how many full text reports were sought for retrieval and how many were not retrieved; (iii) gives authors the option to specify how many studies and reports included in a previous version of the review were carried over into the latest iteration of the review (if an updated review); and (iv) gives authors the option to illustrate the flow of records through the review as separated by type of source (e.g., bibliographic databases, websites, organisation and citation searching). Also, the phrase ‘studies included qualitative synthesis’ has been replaced with ‘studies included in review’, given the former phrase has been incorrectly interpreted by some users as referring to syntheses of qualitative data. Furthermore, the recommendation to report in the flow diagram the number of studies included in quantitative synthesis (e.g., meta-analysis) has been removed, given a systematic review typically includes many quantitative syntheses, and the number of studies included in each varies (e.g., one meta-analysis might include 12 studies, another might include five).

### 1.3 | Transparency and Open Science in evidence syntheses

Broadly speaking, the Open Science movement aims to promote research integrity, experimental and analytical repeatability and full transparency, from project inception to publication and communication. Various definitions and frameworks for Open Science have been proposed (e.g., Open Data, Open Methods, Open Access, Open Source proposed by Kraker et al., 2011, and 44 components by Pontika & Knoth, 2015).

The application of Open Science principles to evidence synthesis has been explored by Haddaway (2018), defined as *Open Synthesis*: the concept has since been expanded to cover 10 proposed components (Open Synthesis Working Group 2020). Open Synthesis is important and beneficial for a number of key reasons (Haddaway, 2018): (1) there is a need to be able to access and verify methods used in reviews and allow interrogation of the fate of each record in the review process; (2) to reduce research waste, data

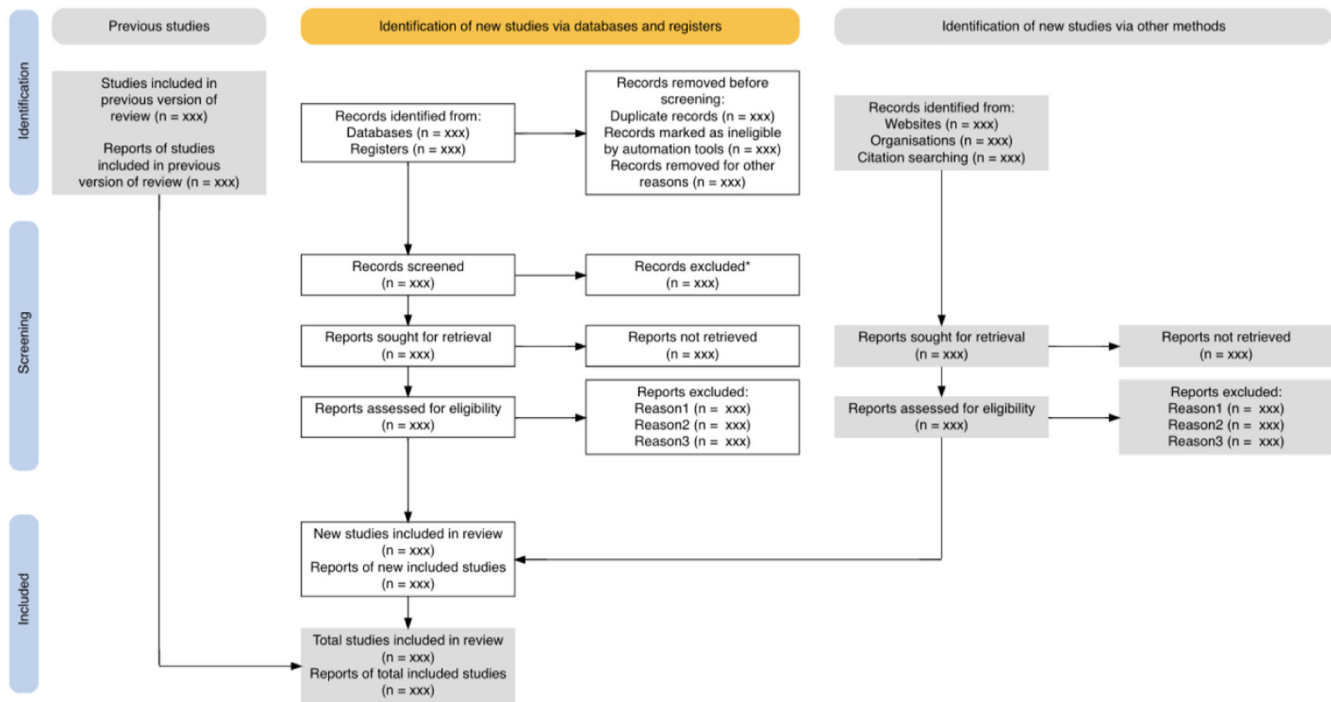
**TABLE 1** Contents of the template CSV file for data upload. Users should alter only the following columns: 'boxtext', 'tooltips', 'url', and 'n'

data	node	box	description	boxtext	tooltips	url	n
NA	node4	prevstud	Grey title box; Previous studies	Previous studies	Grey title box; Previous studies	prevstud.html	xxx
previous_studies	node5	box 1	Studies included in previous version of review	Studies included in previous version of review	Studies included in previous version of review	previous_studies.html	xxx
previous_reports	NA	box 1	Reports of studies included in previous version of review	Reports of studies included in previous version of review	NA	previous_reports.html	xxx
NA	node6	newstud	Yellow title box; Identification of new studies via databases and registers	Identification of new studies via databases and registers	Yellow title box; Identification of new studies via databases and registers	newstud.html	xxx
database_results	node7	box 2	Records identified from: Databases	Databases	Records identified from: Databases	database_results.html	xxx
register_results	NA	box 2	Records identified from: Registers	Registers	NA	NA	xxx
NA	node16	othstud	Grey title box; Identification of new studies via other methods	Identification of new studies via other methods	Grey title box; Identification of new studies via other methods	othstud.html	xxx
website_results	node17	box 11	Records identified from: Websites	Websites	Records identified from: Websites	website_results.html	xxx
organisation_results	box 11	box 11	Records identified from: Organisations	Organisations	NA	NA	xxx
citations_results	NA	box 11	Records identified from: Citation searching	Citation searching	NA	NA	xxx
duplicates	node8	box 3	Duplicate records	Duplicate records	Duplicate records	duplicates.html	xxx
excluded_automatic	NA	box 3	Records marked as ineligible by automation tools	Records marked as ineligible by automation tools	NA	NA	xxx
excluded_other	NA	box 3	Records removed for other reasons	Records removed for other reasons	NA	NA	xxx
records_screened	node9	box 4	Records screened (databases and registers)	Records screened	Records screened (databases and registers)	records_screened.html	xxx
records_excluded	node10	box 5	Records excluded (databases and registers)	Records excluded	Records excluded (databases and registers)	records_excluded.html	xxx

(Continues)

TABLE 1 (Continued)

data	node	box	description	boxtext	tooltips	url	n
dbr_sought_reports	node11	box 6	Reports sought for retrieval (databases and registers)	Reports sought for retrieval	Reports sought for retrieval (databases and registers)	dbr_sought_reports.html	xxx
dbr_notretrieved_reports	node12	box 7	Reports not retrieved (databases and registers)	Reports not retrieved	Reports not retrieved (databases and registers)	dbr_notretrieved_reports.html	xxx
other_sought_reports	node18	box 12	Reports sought for retrieval (other)	Reports sought for retrieval	Reports sought for retrieval (other)	other_sought_reports.html	xxx
other_notretrieved_reports	node19	box 13	Reports not retrieved (other)	Reports not retrieved	Reports not retrieved (other)	other_notretrieved_reports.html	xxx
dbr_assessed	node13	box 8	Reports assessed for eligibility (databases and registers)	Reports assessed for eligibility	Reports assessed for eligibility (databases and registers)	dbr_assessed.html	xxx
dbr_excluded	node14	box 9	Reports excluded (databases and registers): [separate reasons and numbers using: e.g., Reason1, xxx; Reason2, xxx; Reason3, xxx]	Reports excluded:	Reports excluded (databases and registers): [separate reasons and numbers using: e.g., Reason1, xxx; Reason2, xxx; Reason3, xxx]	dbrexcludedrecords.html	Reason1, xxx; Reason2, xxx; Reason3, xxx
other_assessed	node20	box 14	Reports assessed for eligibility (other)	Reports assessed for eligibility	Reports assessed for eligibility (other)	other_assessed.html	xxx
other_excluded	node21	box 15	Reports excluded (other): [separate reasons and numbers using: e.g., Reason1, xxx; Reason2, xxx; Reason3, xxx]	Reports excluded:	Reports excluded (other): [separate reasons and numbers using: e.g., Reason1, xxx; Reason2, xxx; Reason3, xxx]	other_excluded.html	Reason1, xxx; Reason2, xxx; Reason3, xxx
new_studies	node15	box 10	New studies included in review	New studies included in review	New studies included in review	new_studies.html	xxx
new_reports	NA	box 10	Reports of new included studies	Reports of new included studies	NA	NA	xxx
total_studies	node22	box 16	Total studies included in review	Total studies included in review	Total studies included in review	total_studies.html	xxx
total_reports	NA	box 16	Reports of total included studies	Reports of total included studies	NA	NA	xxx
identification	node1	identification	Blue identification box	Identification	Blue identification box	identification.html	xxx
screening	node2	screening	Blue screening box	Screening	Blue screening box	screening.html	xxx
included	node3	included	Blue included box	Included	Blue included box	included.html	xxx



**FIGURE 1** The full output plot from the `PRISMA_flowdiagram()` function

collected within a review should be made publicly accessible and readily reusable in replications, updates and overlapping reviews; (3) capacity building via learning-by-doing is facilitated by having access to machine readable data and code from a review.

## 1.4 | Interactivity and Web 2.0

Systematic review flow diagrams undoubtedly facilitate rapid comprehension of basic review methodology. However, they have far greater potential as a tool for communication and transparency when used not only as static graphics, but also as interactive 'site maps' for reviews. This is the essence of the concept 'Web 2.0'; a rethinking of the Internet as a tool for interactivity, rather than simply passive communication (Blank & Reisdorf, 2012). Flow diagrams in their crudest sense consist of inputs, processes and outputs, with the 'nodes' (i.e., boxes) in a systematic review flow diagram containing summaries of the numbers of records included or excluded at each stage, and 'edges' (i.e., arrows) indicating the 'flow' or movement of records from information sources, through the screening stages of the review, to the final set of included studies. For each node, there is a rich set of information relating both to the methods used and the respective associated records: for example, the number of records excluded at full text eligibility screening are presented alongside a summary of the reasons for exclusion.

In a static review document, it may require substantial effort to determine the methods used to process records or the underlying records themselves. Indeed, the difficulty in locating the relevant information (particularly if stored in supplementary data) often

hampers peer-review and editorial assessment. This is one of the key reasons that reporting standards require authors to specify the location of relevant information in review protocols or reports (e.g., see the PRISMA checklist; <http://www.prisma-statement.org/PRISMAStatement/Checklist>). However, if we repurpose the flow diagram from a static element to an interactive 'site map' of the review, readers may immediately navigate to relevant information regarding review methods, inputs and outputs. Cross-linking between different elements of a review (e.g., the methods or the results for any one step) may help to facilitate the validation and assessment of systematic reviews and make it far easier to access and reuse their methods, data and code. Such interactivity could be achieved through hyperlinking within static digital files, such as PDF (portable document format) files, or through web-based visualisations that would facilitate updating or 'living reviews' (Blank & Reisdorf, 2012).

Furthermore, by embedding and nesting relevant information behind an interactive visualisation such as a flow diagram, review authors could make use of a key concept in science communication: that of *simplification*. Simplification is a key principle in audio-visual science communication (Elliott et al., 2017) and relies on prioritisation of information rather than 'dumbing down' (Finkler & Leon, 2019). Extensive detail on the methods employed and on the reporting of information sources, data inputs and outputs could be accessed via hyperlinks, with core information placed front-and-centre. This layered or nested approach to science communication would allow the reader to choose how much and what type of information to view, rather than the linear format currently used across science publishing.

## 2 | METHODS

### 2.1 | Objectives

This project had the following aims:

- 1) to develop a novel package for the R programming environment (R Core Team, 2018) for producing systematic review flow diagrams that conform to the latest update of the PRISMA statement (Page et al., 2021);

- 2) to adapt this code and publish a free-to-use, web-based tool (a *Shiny app*) for producing publication-quality flow diagram figures without any necessary prior coding experience;
- 3) to allow users to produce interactive versions of the flow diagrams that include hyperlinks to specific web pages, files or document sections.

The project was produced collaboratively as part of the Evidence Synthesis Hackathon (<https://www.eshackathon.org>) using a combination of languages (R, DOT, HTML and JavaScript) with the aim of

#### (a) PRISMA flow diagram Shiny app landing page

PRISMA Flow Diagram Home Create flow diagram

Systematic reviews should be described in a high degree of methodological detail. The PRISMA Statement calls for a high level of reporting detail in systematic reviews and meta-analyses. An integral part of the methodological description of a review is a flow diagram.

This tool allows you to produce a flow diagram for your own review that conforms to the PRISMA2020 Statement. You can provide the numbers and texts for the boxes in the data entry section of the 'Create flow diagram' tab. Alternatively, you can add your own values using the template file below.

At present, this version of the tool doesn't support embedding tooltips and hyperlinks in the plot. For this functionality, please use the PRISMA2020 flow diagram R package on GitHub.

Please let us know if you have any feedback or if you encounter an error by sending an email to [neal.haddaway@sei.org](mailto:neal.haddaway@sei.org)

[Download the template CSV file](#)

Upload your edited file here:  
Choose CSV File  
Browse... No file selected

Please cite as:  
Neal R Haddaway, Chris C Pritchard, Luke A McGuinness. (2020). PRISMA2020: R package and ShinyApp for producing PRISMA 2020 compliant flow diagrams (Version 0.0.2). Zenodo. <http://doi.org/10.5281/zenodo.4267835>  
[Download citation \(.ris\)](#)

Credits:  
Neal R Haddaway (creator, author)  
Luke A McGuinness (coder, author)  
Chris C Pritchard (coder, author)  
Matthew J Page (advisor)  
Jack Walsey (advisor)

Created November 2020, Updated June 2021

#### (b) Data entry and flow diagram page

Main options

Previous studies Other searches for studies  
Not included Included

Click to reset

Identification

Databases Registers  
0 0

Websites Organisations  
0 0

Citations  
0

Duplicates removed  
0

Automatically excluded Other exclusions  
0 0

Screening

Records screened Records excluded  
0 0

Reports sought Reports not retrieved  
0 0

Identification of new studies via databases and registers

Records identified from:  
Databases (n = 0)  
Registers (n = 0)

Records removed before screening:  
Duplicate records (n = 0)  
Records marked as ineligible by automation tools (n = 0)  
Records removed for other reasons (n = 0)

Records screened (n = 0)

Records excluded (n = 0)

Reports sought for retrieval (n = 0)

Reports not retrieved (n = 0)

Reports assessed for eligibility (n = 0)

Reports excluded:  
Reason1 (n = xxx)  
Reason2 (n = xxx)  
Reason3 (n = xxx)

New studies included in review (n = 0)  
Reports of new included studies (n = 0)

Identification of new studies via other methods

Records identified from:  
Websites (n = 0)  
Organisations (n = 0)  
Citation searching (n = 0)

Reports sought for retrieval (n = 0)

Reports not retrieved (n = 0)

Reports assessed for eligibility (n = 0)

Reports excluded:  
Reason1 (n = xxx)  
Reason2 (n = xxx)  
Reason3 (n = xxx)

FIGURE 2 Screenshot of the PRISMA2020 Shiny app (a) landing page and (b) data entry and diagram visualisation page



being provided to the public as a free and open source R package and Shiny app. The project code was published and managed on GitHub ([Haddaway et al., 2021]; <https://github.com/nehaddaway/PRISMA2020>) and the Shiny app is hosted on a subscription-based Shiny server ([https://estech.shinyapps.io/prisma\\_flowdiagram](https://estech.shinyapps.io/prisma_flowdiagram)). Code has been annotated and documented in line with coding best practices and to facilitate understanding and reuse. The PRISMA2020 package has also been published on CRAN (the Comprehensive R Archive Network) in their archive of R packages (<https://cran.r-project.org/web/packages/PRISMA2020/index.html>).

### 3 | RESULTS

In the following pages, we summarise the functionality of the R package (requiring knowledge of coding in R) and Shiny app (a point-and-click web app with no need for coding), providing a summary in lay terms, along with a more detailed description for the code-savvy ('Code detail' boxes). Functions are indicated by courier font, whilst packages are indicated by italics.

#### 3.1 | The PRISMA2020 R package—For users with coding experience

##### 3.1.1 | Data import and cleaning

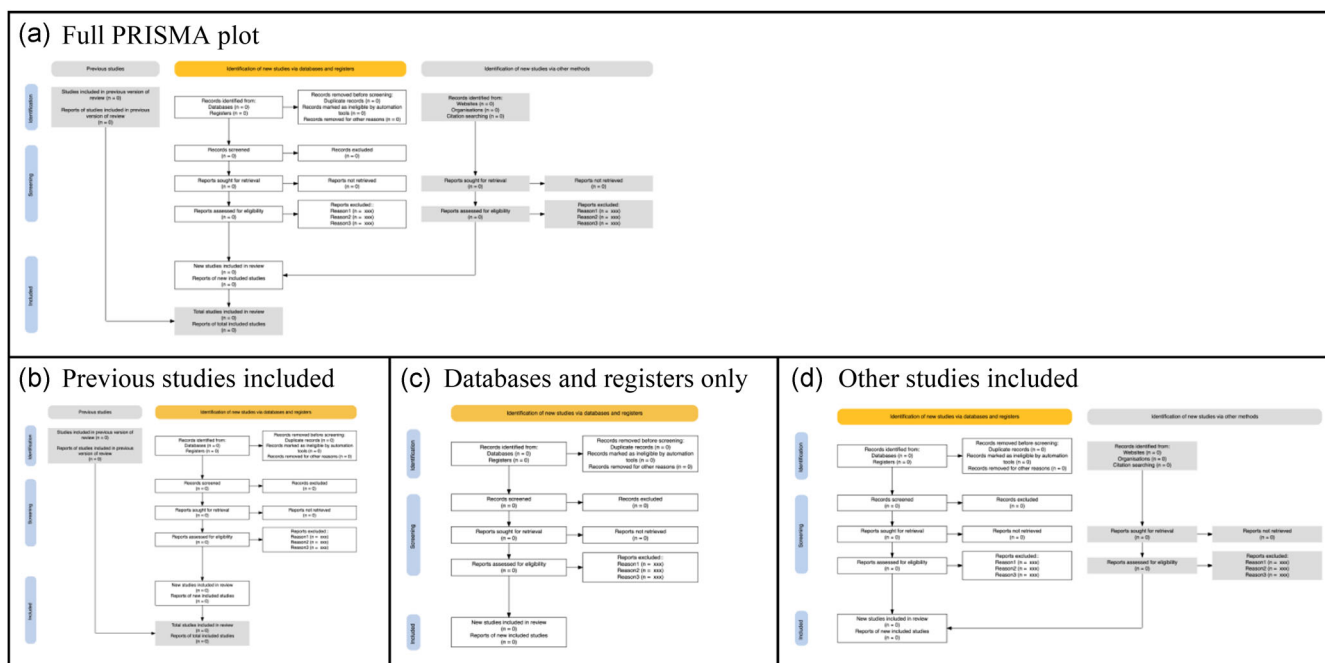
The data needed for the `PRISMA_flowdiagram()` function can be entered either directly as a set of numbers or R objects, but data

upload can be facilitated by using a template comma separated value (CSV) file (see Table 1). We recommend the use of a CSV file as opposed to manually inputting numbers, as this allows for better reproducibility/transparency, as the underlying CSV can be shared. This file can be edited to a large extent and the edits incorporated into the text, numbers, hyperlinks and tooltips used to make the plot. Users can alter the following columns: 'boxtext' (the text that appears in the flow diagram), 'tooltips' (the text that appears as the mouse hovers over the box), 'url' (the url to which the hyperlink leads on clicking), and 'n' (the number appearing in the flow diagram).

The function `PRISMA_read()` reads in a template CSV file containing data to display in the flow diagram, including text contents, quantitative data (i.e., the number of records in each box), tooltips (i.e., the text that appears when the mouse hovers over a box), and hyperlinks for 'on click' functionality. The output is a list of named objects that can be read directly into `PRISMA_flowdiagram()`.

```
Code detail: The PRISMA_read() function uses text matching against a set of node (or box) names to assign the uploaded data to the appropriate box in the figure, for example:

previous_studies <- data[grep('previous_studies', data[,1]),]$n
```



**FIGURE 3** The possible layouts that can be obtained via the 'previous studies' and 'other studies' arms checkboxes. (a) The full plot, (b) previous studies included, (c) databases and registers only, and (d) other studies included

### 3.1.2 | Creating a static flow diagram

The function `PRISMA_flowdiagram()` produces a PRISMA 2020-style flow diagram for systematic reviews. In summary, boxes are placed at specific locations across the graph, and they are automatically connected with arrows according to a specified set of connections.

**Code detail:** `PRISMA_flowdiagram()` uses the `grViz()` function from the *DiagrammeR* package (Iannone & Iannone, 2020) to plot a DOT graphic using `layout = neato` to explicitly place 'nodes' (boxes) at a particular location and `splines='ortho'` to specify axis-aligned edges are drawn between nodes. The label (including data) and tooltip for each node is read in within the main text of the function by using `paste()` to combine DOT strings and R objects.

Along with the text, data, tooltips and hyperlinks, users can specify whether to plot the 'previous studies' arm or the 'other studies' arm of the flow diagram by specifying these options within the `PRISMA_flowdiagram()` function.

In addition, the font, box fill, box line colour and line arrow head/tail can be altered as desired.

**Code detail:** Since text rotation is not supported in DOT or *DiagrammeR*, the vertical labels for the left-hand blue bars are added via JavaScript appending using the `appendContent()` and `onStaticRenderComplete()` functions from the *htmlwidgets* package (Vaidyanathan et al., 2018) to append a block of JavaScript to the HTML output.

First, within the R code, a placeholder label is created consisting of a single whitespace. The JavaScript code uses the `document.getElementById()` to locate each of the blue bar nodes and replace the whitespace with the appropriate label. A CSS transform is applied to rotate the label by 90 degrees and the correct x and y coordinates for the new label are calculated based on their previous values. This means that the label location is adjusted based on the presence or absence of the 'previous' and 'other' arms and is able to withstand changes to the diagram format moving forward.

The function also includes the ability to plot an interactive version if the function parameter is set to 'TRUE', as described in Point 4, below.

The final plot output (see Figure 1) can be saved in a range of file formats (HTML, PDF, PNG, SVG, PS or WEBP).

### 3.1.3 | Creating an interactive flow diagram

Flow diagrams can be made interactive by specifying the additional parameter `interactive = TRUE` (this defaults to `FALSE`) in the `PRISMA_flowdiagram()` function. The resulting HTML output plot includes hyperlinks (links to other webpages) on click for each box, along with the tooltips (temporarily visible boxes containing text that appear when the mouse hovers over content) specified in the main `PRISMA_flowdiagram()` function (see above).

**Code detail:** The internal function `PRISMA_interactive_()` uses the `prependContent()` and `onStaticRenderComplete()` functions from the *htmlwidgets* package (Vaidyanathan et al., 2018) to prepend a block of JavaScript to the HTML output. This JavaScript identifies each node in turn using `getElementById(id)` and inserts an HTML anchor element carrying the relevant hyperlink for each node using the internal function `PRISMA_add_hyperlink_()`.

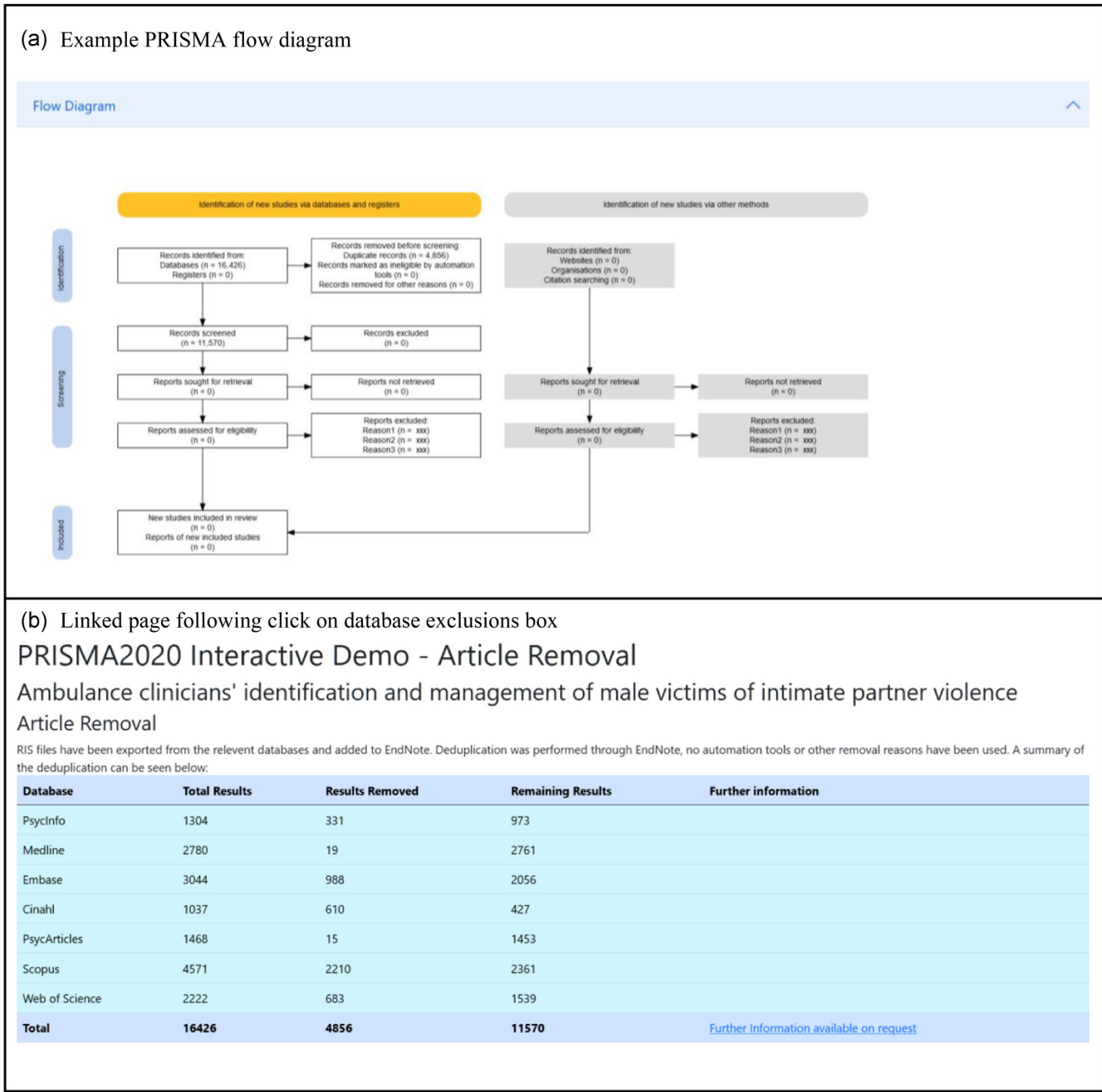
### 3.1.4 | Saving the output as a file

The `PRISMA_save()` function allows for the flow diagram to be saved as an HTML file (with interactivity preserved, either as a standalone file with scripts and styles embedded, or as a ZIP archive in which these are stored separately), or as a PDF, PNG, SVG, PS or WEBP file (without interactivity). This function takes the plot produced by `PRISMA_flowdiagram()` and saves the file. A default option for filename is provided, but this can be overridden along with the filetype which is calculated from the file extension by default.

**Code detail:** When saving as HTML, the `PRISMA_save()` function uses the `savewidget()` function from the *htmlwidgets* package (Vaidyanathan et al., 2018). When saving as other formats, the internal function `PRISMA_gen_tmp_svg()` is used, this first uses `savewidget()` to create an HTML file in a temporary directory and then uses the various XML manipulation functions from the *xml2* package (Wickham et al., 2018) to step through the HTML, using `xpath` (Dyck et al., 2014) to find the SVG embedded within the HTML.

As JavaScript is not supported in SVG files, the *xml2* package is again used to add a rotate transformation and programmatically alter the x and y coordinates to create the blue vertical labels. Following this, the temporary SVG is either copied to its final destination, or the *rsvg* (Ooms, 2017) package is used to convert it into the desired output format.





**FIGURE 4** Screenshot of the case study website, showing the PRISMA flow diagram (a) and the resulting page linked to by clicking on the database exclusions box (b)

### 3.2 | The Shiny app—A web-based application for all

Shiny is a package within the R environment that allows users to construct standalone web-based applications based on R functions (Beeley, 2013). The ‘app’ can be interacted with by entering data, running functions with user-specified settings to plot figures, and downloading the resultant figures in a variety of formats.

The PRISMA2020 Shiny app is available free-of-charge ([https://estech.shinyapps.io/prisma\\_flowdiagram/](https://estech.shinyapps.io/prisma_flowdiagram/)) and can also be found through the PRISMA website (<http://prisma-statement.org/prismastatement/flowdiagram.aspx>). The app landing page (the ‘Home’ tab) describes the app and its background, linking to the PRISMA website and PRISMA 2020 statement (Page et al., 2021) (see Figure 2). Users can enter their data either by uploading an edited template CSV file, or by manually entering data in the ‘Create flow

diagram' tab. Once uploaded, users proceed to this tab to see the resultant figure.

On the 'Create flow diagram' tab users can specify whether to include the 'previous studies' and 'other studies' arms of the flow diagram using the check boxes. The resulting flow diagram (see Figure 3). The 'Previous studies' and 'Other studies' arms can be toggled on and off via the 'Data upload' tab and the plot responds reactively.

Whilst it is currently not possible to add interactivity using the manual data entry in the Shiny app, users can customise hyperlinks and tooltips by uploading data in a CSV file and downloading of an interactive HTML plot.

**Code detail:** Shiny does not support HTML appending or prepending (adding code before or after a given element) via *DiagrammeR*. Instead, a different method is used to label the blue nodes. The javascript file `labels.js` is included via a script tag inserted in the `<head>` area of the shiny HTML pages. This contains several functions, the `renderLabel()` function adds a label to the given node, in the same way as the JS appends in *DiagrammeR*.

The `createLabels()` function is called just before the plot is re-rendered, this registers a `MutationObserver` that waits for the nodes to be created. Once the nodes are visible to the DOM (Document Object Model, a programming interface for HTML), the `renderLabel()` function is called, once for each node, to add the labels.

### 3.3 | Interactivity

The interactivity here represents an additional step to cross link and host the interactive PRISMA flow diagram with the relevant texts and data. As described above, such interactivity allows for radically improved transparency and may be of particular value for highly complex review methods. Including interactivity obviously corresponds to additional effort on the behalf of review authors, but has clear benefits for transparency and communication. Interactivity is provided by both the R package and the Shiny app.

Interactivity in the *PRISMA2020* flow diagrams is provided in two ways. First, mouse-over tooltips appear as the user's mouse is moved over a particular box. These popup boxes can contain user-specified text providing more information as desired. For example, a short elaboration of the numbers of text in each box to clarify meanings. Alternatively, tooltips can provide an explanation of the information that will be hyperlinked to on clicking. Second, the boxes can be given hyperlinks so that the user can follow a predetermined link. These links can be anchors within a document or webpage, or datafiles or web pages stored on external or local

repositories (e.g., supplementary files on a data repository such as figshare or Zenodo).

As described above, this interactivity conforms to the principle of science communication simplicity by prioritising information provision hierarchically (i.e., showing critical information first, with further details accessible on interrogation). Tooltips provide a semi-passive means of providing information: the user is exposed to further details as they move their mouse over the plot. Hyperlinks are an active means of requesting further information. This nested, hierarchical provision of information may be particularly useful for complex systematic review methodology.

It is worth noting that users should ensure they do not breach bibliographic database (and other) Terms of Use or inadvertently infringe copyright by linking to directly exported search results or including copyrighted data such as abstracts. Many providers would not count this form of transparency as outside acceptable use, but we encourage users to be certain for the resources they have used. One way to avoid this would be to provide a digitised (e.g., comma-separated text file) list of digital object identifiers for all records linked to in an interactive version of the flow diagram: this would contravene neither copyright nor Terms of Use and could be transformed into a full set of citations using freely accessible resources like CrossRef.

### 3.4 | Case study

We have prepared a case study that demonstrates possible interactivity that can be employed in a web-based *PRISMA2020* flow diagram (see Figure 4). The example website is available at <https://prisma-flowdiagram.github.io/>.

The website is based on data from an ongoing systematic review into ambulance clinician responses to adult male victims of intimate partner violence (Mackay et al., 2021). The site uses a flowchart generated from this software, alongside bootstrap (<https://getbootstrap.com>) to make a fully interactive experience, enabling users to interrogate various aspects of the review. As the review is currently underway, the site will be updated as the review progresses.

## 4 | DISCUSSION

The *PRISMA 2020* update represents a significant development of the *PRISMA* statement, increasing the usability and level of detail needed in systematic reviews. The *PRISMA2020* flow diagram similarly provides a clearer and more detailed template. To our knowledge, our tool is the only one catering for the *PRISMA 2020* update. Previous tools offer far less flexibility and usability, for example, those provided by the theta Collaborative (<http://prisma.thetacollaborative.ca/>), Jack Wasey's R package *PRISMAstatement* (<https://rdr.io/cran/PRISMAstatement/>), and 'plot\_PRISMA()' from the *metagear* R package (<https://rdr.io/cran/metagear/>).

We have developed a user-friendly tool for producing PRISMA 2020-compliant flow diagrams for users with coding experience and, importantly, for users without prior experience in coding by making use of Shiny ([https://estech.shinyapps.io/prisma\\_flowdiagram/](https://estech.shinyapps.io/prisma_flowdiagram/)). This free-to-use tool will make it easier to produce clear and PRISMA 2020-compliant systematic review flow diagrams. Significantly, users can also produce interactive flow diagrams for the first time, allowing readers of their reviews to smoothly and swiftly explore and navigate to further details of the methods and results of a review.

In addition, the ability to produce flow diagrams using code in a data-driven approach carries with it a number of benefits, including: facilitating Open Science (specifically Open Code); reducing the risk of transcription errors; and, opening up possibilities for reproducible documents such as executable research articles (Tsang & Maciocci, 2020) and communicating the results of living systematic reviews (Elliott et al., 2017).

We believe this tool will increase use of PRISMA flow diagrams, improve the compliance and quality of flow diagrams, and facilitate strong science communication of the methods and results of systematic reviews by making use of interactivity. We encourage the systematic review community to make use of the tool, and provide feedback to streamline and improve usability and efficiency.

## CONFLICTS OF INTEREST

Matthew Page co-led the development of the PRISMA 2020 statement and Luke McGuinness is a coauthor of the PRISMA 2020 statement, but they have no commercial interest in the use of this reporting guideline.

## ACKNOWLEDGEMENTS

We thank Jack Wasey for his work on an R package for PRISMA 2009-compliant flow diagrams. We also thank the many users of the beta version of the R package and Shiny App for feedback instrumental in improving the user experience. This study is a project of the Evidence Synthesis Hackathon (<https://www.eshackathon.org/software/PRISMA2020.html>).

## AUTHOR CONTRIBUTIONS

This manuscript, the R package and the Shiny App were conceived and drafted by Neal Haddaway. Substantial code improvements were made by Luke McGuinness and Chris Pritchard. Matthew Page provided conceptual feedback. All authors contributed to the editing and revision of the manuscript, and have read and agreed to publication in its final form.

## REFERENCES

Beeley, C. (2013). *Web application development with R using Shiny*. Packt Publishing Ltd.

Blank, G., & Reisdorf, B. C. (2012). The participatory web: A user perspective on Web 2.0. *Information, Communication & Society*, 15(4), 537–554.

Collaboration for Environmental Evidence. (2018). *Guidelines and Standards for Evidence Synthesis in Environmental Management: Version 5.0. Project Report*.

Dyck, M., Robie, J., Snelson, J., & Chamberlin, D. (2014). XML path language (XPath) 3.1. W3C Working Draft.

Elliott, J. H., Synnot, A., Turner, T., Simmonds, M., Akl, E. A., McDonald, S., Salanti, G., Meerpohl, J., MacLehose, H., Hilton, J., Tovey, D., Shemilt, I., Thomas, J., & Living Systematic Review, N. (2017). Living systematic review: 1. Introduction—The why, what, when, and how. *Journal of Clinical Epidemiology*, 91, 23–30.

Finkler, W., & Leon, B. (2019). The power of storytelling and video: A visual rhetoric for science communication. *Journal of science communication*, 18(5), A02.

Haddaway, N., McGuinness, L., & Pritchard, C. (2021). *PRISMA2020: R package and ShinyApp for producing PRISMA 2020 compliant flow diagrams (Version 0.0.1)*. Zenodo.

Haddaway, N. R. (2018). Open Synthesis: on the need for evidence synthesis to embrace Open Science. *Environmental Evidence*, 7(1), 1–5.

Haddaway, N. R., Macura, B., Whaley, P., & Pullin, A. S. (2018). ROSES RepOrting standards for Systematic Evidence Syntheses: pro forma, flow-diagram and descriptive summary of the plan and conduct of environmental systematic reviews and systematic maps. *Environmental Evidence*, 7(1), 1–8.

Higgins, J. P., Thomas, J., Chandler, J., Cumpston, M., Li, T., Page, M. J., & Welch, V. (2019). *Cochrane handbook for systematic reviews of interventions*. John Wiley & Sons.

Iannone, R., & Iannone, M. R. (2020). *Package 'DiagrammeR'*.

Kraker, P., Leony, D., Reinhardt, W., & Beham, G. (2011). The case for an open science in technology enhanced learning. *International Journal of Technology Enhanced Learning*, 3(6), 643–654.

Liberati, A., Altman, D. G., Tetzlaff, J., Mulrow, C., Gøtzsche, P., & Ioannidis, J. (2009). The PRISMA statement for reporting systematic and meta-analyses of studies that evaluate interventions: explanation and elaboration. *PLoS Medicine*, 6(7), 1–28.

Mackay, J., Pritchard, C., Jieru Yan, R., Kuhn, I., & Barker, E. (2021). *Ambulance clinicians' identification and management of male victims of intimate partner violence: A systematic review protocol PROSPE RO*. CRD42021253584.

Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., & Group, P. (2009). Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *PLoS Medicine*, 6(7), e1000097.

Moher, D., Tetzlaff, J., Tricco, A. C., Sampson, M., & Altman, D. G. (2007). Epidemiology and reporting characteristics of systematic reviews. *PLoS Medicine*, 4(3), e78.

Ooms, J. (2017). *rsvg: Render SVG Images into PDF, PNG, PostScript, or Bitmap Arrays R package version 1.1*. R Foundation for Statistical Computing.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hrobjartsson, A., Lalu, M. M., Li, T., Loder, L. W., Mayo-Wilson, E., McDonald, S.... Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372.

Page, M. J., Shamseer, L., Altman, D. G., Tetzlaff, J., Sampson, M., Tricco, A. C., Catalá-López, F., Li, L., Reid, E. K., Sarkis-Onofre, R., & Moher, D. (2016). Epidemiology and reporting characteristics of systematic reviews of biomedical research: A cross-sectional study. *PLoS Medicine*, 13(5), e1002028.

Pontika, N., & Knoth, P. (2015). Open science taxonomy.

Stovold, E., Beecher, D., Foxlee, R., & Noel-Storr, A. (2014). Study flow diagrams in Cochrane systematic review updates: An adapted PRISMA flow diagram. *Systematic Reviews*, 3(1), 1–5.

R Core Team. (2018). *R: A language and environment*. R (version 3.5.3).

- Tsang, E., & Maciocci, G. (2020). *Welcome to a new ERA of reproducible publishing*. eLIFE Labs.
- Vaidyanathan, R., Xie, Y., Allaire, J., Cheng, J., & Russell, K. (2018). *htmlwidgets: HTML Widgets for R*. R package version 06.
- Vu-Ngoc, H., Elawady, S. S., Mehyar, G. M., Abdelhamid, A. H., Mattar, O. M., Halhouli, O., Vuong, N. L., Ali, C., Hassan, U. H., Kien, N. D., Hirayama, K., & Huy, N. T. (2018). Quality of flow diagram in systematic review and/or meta-analysis. *PLoS One*, *13*(6), e0195955.
- Wickham, H., Hester, J., & Ooms, J. (2018). *xml2: Parse XML*. R package version 1.2. 0.

**How to cite this article:** Haddaway, N. R., Page, M. J., Pritchard, C. C., & McGuinness, L. A. (2022). *PRISMA2020: An R package and Shiny app for producing PRISMA 2020-compliant flow diagrams, with interactivity for optimised digital transparency and Open Synthesis*. *Campbell Systematic Reviews*, *18*, e1230.  
<https://doi.org/10.1002/cl2.1230>