PDF Data Extractor (PDE) - A Free Web Application and R Package Allowing the Extraction of Tables from Portable Document Format (PDF) Files and High-Throughput Keyword Searches of Full-Text Articles

Erik Stricker[1§*] **and Michael E. Scheurer**[1§*]

[1]Baylor College of Medicine

***For correspondence:**
stricker@bcm.edu (ES);
scheurer@bcm.edu (MES)

Present address: [§]Department of Pediatrics, Department of Molecular Virology and Microbiology
Baylor College of Medicine, Houston, TX 77030, USA

Abstract The PDF Data Extractor (`PDE`) R package is designed to perform comprehensive literature reviews for scientists at any stage in a user-friendly way. The `PDE_analyzer_i()` function permits the user to filter and search thousands of scientific articles using a simple user interface, requiring no bioinformatics skills. In the additional `PDE_reader_i()` interface, the user can then quickly browse the sentences with detected keywords, open the full-text article, when required, and convert tables conveniently from PDF files to Excel sheets (pdf2table). Specific features of the literature analysis include the adaptability of analysis parameters and the detection of abbreviations of search words in articles.

In this article, we demonstrate and exemplify how the `PDE` package allows the user-friendly, efficient, and automated extraction of meta-data from full-text articles, which can aid in summarizing the existing literature on any topic of interest. As such, we recommend the use of the `PDE` package as the first step in conducting an extensive review of the scientific literature. The `PDE` package is available from the Comprehensive R Archive Network at https://CRAN.R-project.org/package=PDE.

Introduction

In an age of exponentially increasing numbers of published scientific articles, it is surprising that most systematic literature reviews and extraction of information from tables are still conducted by manually processing articles individually. Systematic literature reviews aim to find and collect relevant information concerning a specific research question and are an essential step in virtually every area of research, e.g., for the preparation of review articles, project proposals, and experimental designs. While machine learning tools are available for literature searches and screens (***Marshall and Wallace, 2019***), they: 1) require a large number of manually evaluated articles for the training of the tool, 2) are often restricted to filtering articles by study design or choosing topics from a limited set of terms, and 3) are generally limited to the evaluation of article titles and abstracts. The

PDF Data Extractor (PDE) R package easily extracts information and tables from full-text articles in Portable Document Format (PDF) based on user-defined keywords and does not require a training set.

In addition to the high-througput evaluation and categorization of scientific articles, the conversion of tables from PDF files into processable file formats such as comma- or tab-separated values files, i.e., *.csv or *.tsv, is often a tedious but integral part of literature reviews. While many tools allow the fast and fairly accurate conversion of PDF files into Microsoft Excel files, they often require a paid subscription for the processing of more than a few files (e.g., https://smallpdf.com/, https://www.adobe.com/acrobat/online/pdf-to-excel.html, and https://pdftables.com/), are limited in input file numbers and therefore do not allow high-throughput table extraction (e.g., https://pdftoxls.com/, https://docs.zone/pdf-to-excel, and https://www.pdftoexcel.com/), lose table headings and footnotes (e.g., `tabulizer` R package, and https://pdftoxls.com/) or require manual selection of tables in a file and adjustment of the table format information (e.g., `pdf_text` and `pdf_data` from `pdftools` R package which only extract the PDF file text or positional information of words (e.g., Microsoft Excel). The PDE was developed to overcome these hurdles by providing a free tool with a simple user interface allowing the extraction of tables with their headings and footnotes from hundreds of PDF files in minutes. The detection of tables is search word based and therefore falsely extracted paragraphs are greatly reduced. On comparison to the above-mentioned tools, the PDE allows search and filter word input before analyses, resulting in fewer and more specific output files. To enhance the advantage provided by filter and search words, the PDE utilzes machine learning to identify abbreviations of given filter and search words and extract additional relevant data accordingly.

In this article, we will list all possible parameters and give a detailed description on their use for both the `PDE_analyzer_i()` and `PDE_reader_i()`, outline the installation and use of the PDE package by means of a small reproducible example, compare the accuracy of the PDE package against a published Cochrane Systematic Review by ***Kraal et al.*** (***2017***) and discusses the limitations of the PDE package and expand on features included in future updates. The PDE package is available from the Comprehensive R Archive Network at https://CRAN.R-project.org/package=PDE and includes the small sample set of 5 PDF files presented in this paper to exemplify the use of the package.

Results

PDE R package

The PDF Data Extractor (PDE) R package is designed to perform comprehensive literature reviews for scientists at any stage in a user-friendly way. The `PDE_analyzer_i()` function permits the user to filter and search thousands of scientific articles using a simple user interface, requiring no bioinformatics skills. In the additional `PDE_reader_i()` interface, the user can then quickly browse the sentences with detected keywords, open the full-text article, when required, and convert tables conveniently from PDF files to Excel sheets (pdf2table).

To overcome the potential drawback of non-existing graphical user interfaces (GUI) in R (***R Core Team, 2019***) and the requirement of coding skills, we decided to use Tcl/Tk for the creation of a user-friendly interface. The Tcl/Tk implementation with the `tcltk` (***R Core Team, 2019***) and `tcltk2` (***Grosjean, 2019***) packages is part of Microsoft Windows and Linux versions of R by default and easy to install on a Mac. Furthermore, the `tcltk` package offers the display of interactive tables allowing the user to quickly assess the results of the keyword searches. Additionally, we are using the XpdfReader (***Noonburg, 1996–2020***) command line tools `pdftotext` and `pdftohtml` to allow the searchability of PDF files and the `pdftopng` tool to export ambiguous tables. The user has the choice to either install the XpdfReader using the `PDE_install_XpdfReader4.02()` function or download

and install the software from the developer's website https://www.xpdfreader.com/download.html. The latest versions 4.02 of the `pdftotext` and `pdftohtml` tools perform especially well in the reassembly of words, e.g., unequally spaced letters or hyphenation at the end of a line.

The `PDE_analyzer()` performs the sentence and table extraction on the complete full-text articles. The highlights of the `PDE_analyzer()` include:

1. Filter words: The user can provide a list of filter words specific for certain study types or topics. The `PDE_analyzer()` only processes articles which carry words from this list when detected a minimum number of times as defined by the user (filter word threshold).
2. Sentence extraction based on search words: A list of user-defined search words are used to extract sentences relevant for the evaluation of the suitability of a scientific article. All sentences carrying at least one of the search words are output into a comma- or tab-separated values file, i.e., *.csv or *.tsv.
3. Abbreviation auto-detection: The `PDE_analyzer()` also has the ability to recognize abbreviations of search words used in articles, without the predefinition of any abbreviations being required.
4. Context extraction: The user can choose to export a preferred number of sentences before and after the sentences containing search words and their abbreviations.
5. `PDE_pdf2table()`: The tables of a PDF document can be exported into a Microsoft Excel readable file format with and without the use of search words. The `PDE_analyzer()` detects the beginning of a table based on the standard annotation of tables in scientific literature, i.e., "Table" [Table index] [Table heading], or user-defined table headings, allowing the exclusion of non-table content from the PDF file.
6. GUI: Facilitated by `tcltk` and `tclk2` R packages, the user can enter analysis parameters using the `PDE_analyzer_i()` interactive version. It allows the generation of jobs, execution of analysis and monitoring of active analyses in a visual interface.

The included `PDE_reader_i()` allows the user-friendly visualization and quick processing of the obtained results. In the additional interface, the user can:

1. quickly browse the sentences with detected keywords
2. open the full-text article, when required
3. obtain all tables from the current PDF file in a Microsoft Excel readable file format
4. flag or mark articles by adding a prefix, i.e., "!_" or "x_", to the file name. (e.g.,

Detailed guide and simple example

The objective of this guide was to demonstrate the automated evaluation of articles and extraction of relevant sentences as well as tables by means of a simple example. The goal was to identify articles having case-control data on Methotrexate administration from a small set of 5 PDF files included in the installation of the package. All PDF files can be found in the `PDE` installation subfolder `examples/Methotrexate/`. The output files of the example can be found in the `PDE` installation folder for reference as well. The folder can be located by running the following code in the R console after installation:

```
R> system.file(package = "PDE")
```

A description on how to install the `PDE` R package can be found in the Materials and Methods (see **Section - Installation**). Below follow the steps for the execution of the example:

1. Run

```
R> library("PDE")
R> PDE_analyzer_i()
```

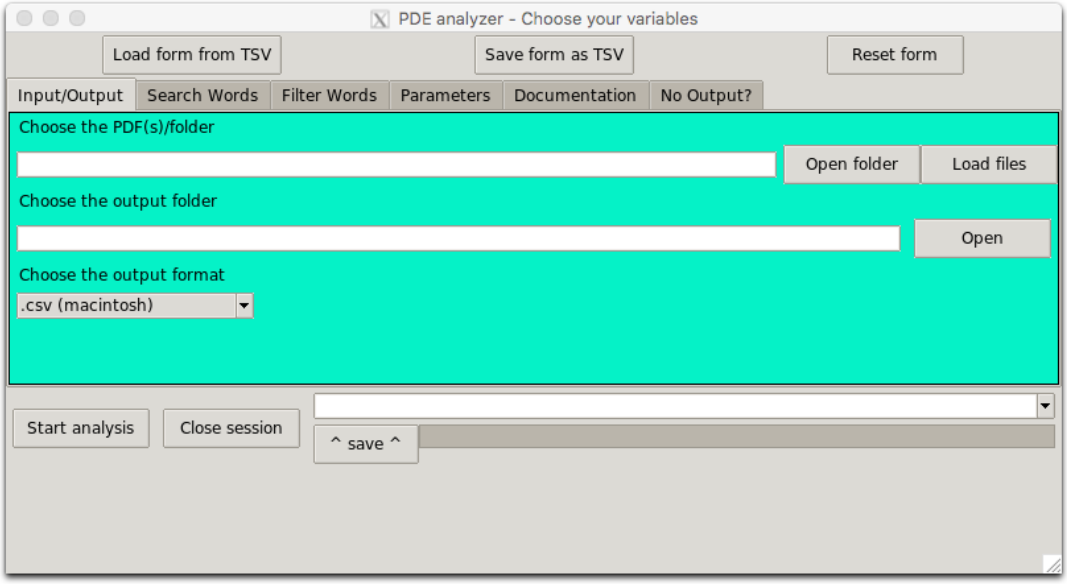


Figure 1. User interface generated by the `PDE_analyzer_i()` on Mac

This should open a user interface (see Fig. 1).

Step-by-step selection of the parameters

Below follows a detailed description of each element of the `PDE_analyzer_i()`. Essential elements required for each analysis are marked by an asterisk. The corresponding parameters saved in the TSV file are indicated below each element and can either be used by directly indicating the TSV file for `PDE_analyzer()` or calling `PDE_extr_data_from_pdfs()` with the parameters.

Alternative to step-by-step selection: Load form from TSV

2. **Load form from TSV/Save form as .tsv:** The filled form can and should be saved as a TSV file at any time, accordingly the saved parameters can be loaded from saved TSV files. Alternatively to the step-by-step selection, all example parameters can be loaded from the file with the name `PDE_parameters_v1.0_all_files+-0.tsv` found in the subfolder `examples/tsvs/` (Fig. 2). Then continue with **Start the PDE_analyzer()**.
3. **Reset form:** This will clear all fields and variables.

Input/Output Tab:

This tab is the only section requiring user input for a simple table extraction (Fig. 2.A).

2. **Open PDF folder*:** Open a folder with PDF files for analysis. All PDF files in the chosen folder and subfolders will be analyzed. For this example, 3 PDF files downloaded from PubMed using `(methotrexate) NOT Review[Publication Type]` as well as 1 erroneous (`99999999_x.pdf`) and 1 empty file (`00000000_x.pdf`) are in the following folder:

   `examples/Methotrexate/`

   The file names indicate the PMIDs. In addition, negative controls are marked with an _x and the files which include tables with the search words are marked with an _! (this naming system is specifically chosen for the example, but generally analyses files are not restricted to any particular naming system other than no two files should have the same name).
   or

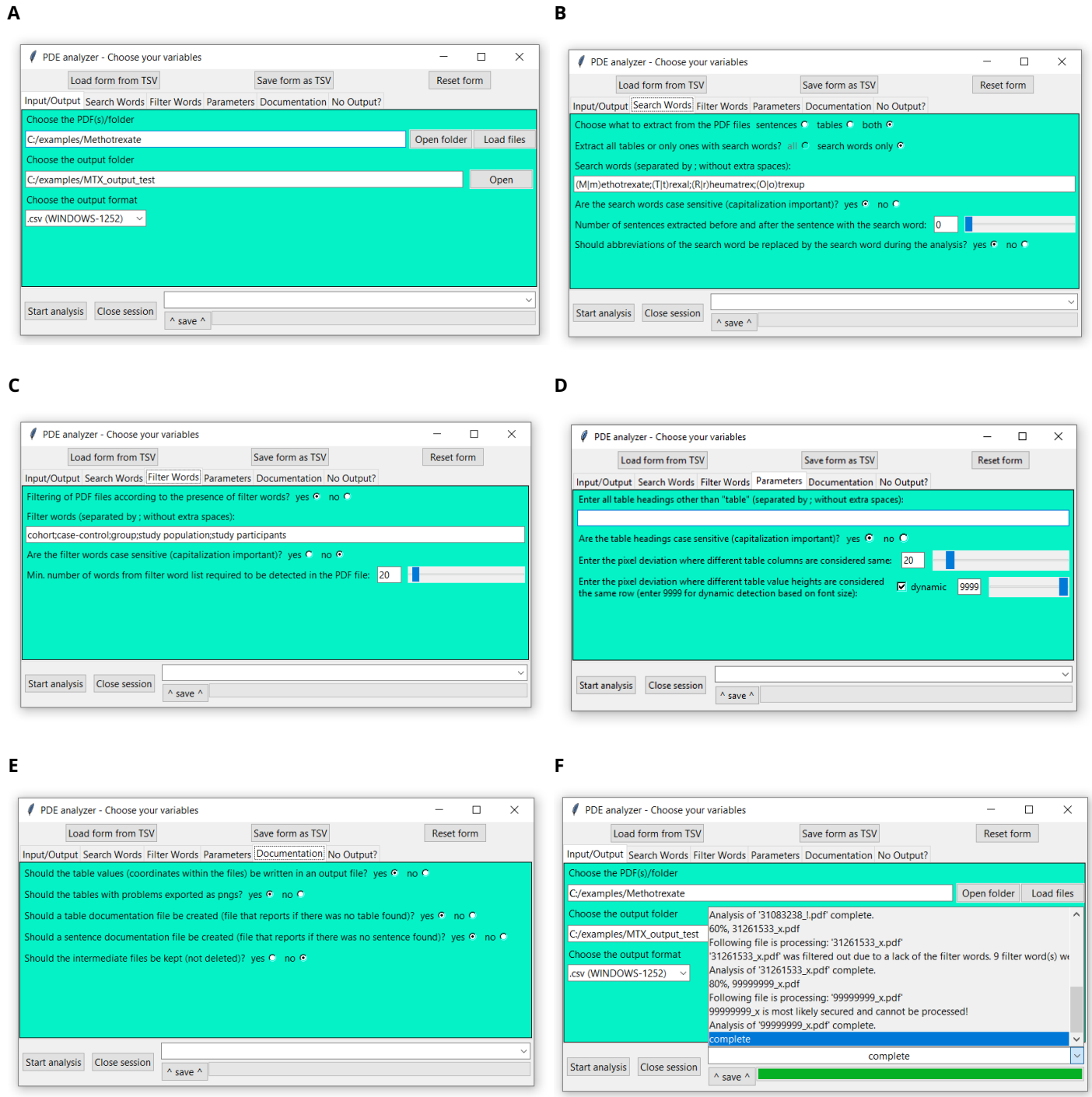


Figure 2. User interface generated by the `PDE_analyzer_i()` on Windows with example data. The objective of this simple example was to demonstrate the detection and extraction of articles having case-control data on Methotrexate administration from a small set of 5 PDF files. All PDF files are located in the `PDE` installation subfolder `examples/Methotrexate/`. For the complete example, (A) the Input/Output, (B) Search Word, (C) Filter Word, (D) Parameter, and (E) Documentation Tabs were modified. Upon initiation of the analyses, progress updates are displayed at the botton right corner of the interface (F).

Load PDF files*: Select one or more PDF files for analysis (use Ctrl and/or Shift to select). Multiple PDF files will be separated by ";" without a space. For the example, select the 5 PDF files in the `examples/Methotrexate/` folder (use Ctrl and/or Shift to select multiple).
Following parameter is saved in the TSV file: `pdfs`

```
examples/Methotrexate/29973177_!.pdf
examples/Methotrexate/31083238_!.pdf
examples/Methotrexate/31261533_x.pdf
examples/Methotrexate/00000000_x.pdf
examples/Methotrexate/99999999_x.pdf
```

3. **Open output folder:** All analysis files will be created inside of this folder; therefore, choose an empty folder or create a new one as output directory, since analyses create at least a number of files equal to the amount of PDF files analyzed. If no output folder is chosen, the results will be saved in the R working directory.
   The files created by the `PDE_analyzer()` should be identical to the files found in `examples/MTX_output_linux`, `examples/MTX_output_mac`, or `examples/MTX_output_win`. Any output folder can be chosen for the example analysis but the folder indicated below is recommended for direct comparison.
   Following parameter is saved in the TSV file: `out`

```
## out (MTX example):
examples/MTX_output_test
```

4. **Choose output format:** The resulting analyses files can either be generated as comma-separated values files (.csv) or tab-separated values files (.tsv), with the former being easier to open and save in Microsoft Excel, while the later leads to less errors when opening in Microsoft Excel (as tabs are rare in texts). Depending on the operating system the output files are opened in, it is recommended to choose the Microsoft Windows (WINDOWS-1252), Mac (macintosh) or Linux (UTF-8) encoding.
   For the example analysis, the comma-separated values table format for Windows was chosen.
   Following parameter is saved in the TSV file: `out.table.format`

```
## out.table.format (MTX example):
.csv (WINDOWS-1252)
```

Search Words Tab:

Search words can help in selecting data of interest for later manual evaluation (Fig. 2.B).

5. **Choose what to extract:** The `PDE_analyzer()` has 2 main functions
   (a) PDF2TXT (extract sentences from PDF files)
   (b) PDF2TABLE (table of PDF to Microsoft Excel file)

   which can be combined or executed separately. Each function can be combined with filters and search words. A file with the sentences carrying the search words will have the name format: [search words]txt+-[context][PDF file name] in the corresponding subfolder. Tables will be named: [PDF file name][number of table][table heading].
   For the example, the analyzer will extract sentences and tables with the keywords. Accordingly, the option below should be chosen.
   Following parameter is saved in the TSV file: `whattoextr`

```
## whattoextr (MTX example):
both
```

6. **Search words?:** The algorithm can either extract **sentences**, **tables**, or **sentences** and **tables** with one of the search words present. If the `tables` only analysis was chosen, the algorithm can also extract all tables detected in the paper (choose this option here). In the latter case, the search words field should remain empty.
   The search words were used to extract all Methotrexate relevant information.

   ```
   ## Search words? (MTX example):
   yes
   ```

7. **Search words:** Type in the list of search words separated by ";" without spaces in between. The list of search words includes all aliases. Parentheses and a vertical line were used in this example to demonstrate how to indicate alternative capitalization or letters. Alternatively, case-sensitivity could be disabled which then required no alternative capitalization. The search words are only separated by semicolons (no spaces for separation).
   Following parameter is saved in the TSV file: `search.words`

   ```
   ## search.words (MTX example):
   (M|m)ethotrexate;(T|t)rexal;(R|r)heumatrex;(O|o)trexup
   ```

8. **Search words case sensitive:** As explained above, the search words were selected to be case-sensitive for this example.
   Following parameter is saved in the TSV file: `ignore.case.sw`

   ```
   ## ignore.case.sw (MTX example):
   yes
   ```

9. **Number of sentences before and after:** When 0 is chosen, only the sentence with the search word is extracted. If any number n is chosen, n number of sentences before and n number of sentences after the sentence with the search word will be extracted. A sentence is currently defined by starting and ending with a ". " (period with a subsequent space).
   For simplicity, only the sentences with the search words were extracted for the example.
   Following parameter is saved in the TSV file: `context`

   ```
   ## context (MTX example):
   0
   ```

10. **Evaluate abbreviations?:** If **yes** was chosen, all abbreviations that were used in the PDF documents for the search words will be saved and then replaced by the abbreviation (search word)$*, e.g., MTX will be replaced by MTX (Methotrexate)$*. In addition plural versions of the abbreviations, i.e., the abbreviation with an "s" at the end will be replaced accordingly as well.
    Abbreviations of Methotrexate such as MTX should also be detected in the document.
    Following parameter is saved in the TSV file: `eval.abbrevs`

    ```
    ## eval.abbrevs (MTX example):
    yes
    ```

Filter Words Tab:

The selection of filter words can help specifying relevant articles and reduce the computational time (Fig. 2.C).

11. **Filter words?:** In some cases, only articles of a certain topic should be analyzed. Filter words provide a way to analyze only articles which carry words from a list at least n times.
    For this analysis, filter words were used to only analyze articles with case-control data.

    ```
    ## Filter words? (MTX example):
    yes
    ```

12. **Filter words:** Type in the list of filter words separated by ";" without spaces in between. A hit will be counted every time a word from the list is detected in the article.
    The words below should be found at a high frequency in case-control papers and are therefore used for the example. The filter words are only separated by semicolons (no spaces for separation).
    Following parameter is saved in the TSV file: `filter.words`

    ```
    ## filter.words (MTX example):
    cohort;case-control;group;study population;study participants
    ```

13. **Filter words case sensitive:** E.g., for "Word", if "no" was chosen then "word", "WORD", "Word", etc., will be detected, if "yes" was chosen only "Word" will be detected.
    Since it does not matter for the example if a word is found capitalized at the beginning of a sentence, in a heading or within a sentence, the search is not case-sensitive.
    Following parameter is saved in the TSV file: `ignore.case.fw`

    ```
    ## ignore.case.fw (MTX example):
    no
    ```

14. **Filter word times:** This represents the minimum number of hits described above which has to be detected for a paper to be further analyzed. If the threshold is not met, a documentation file can be exported if selected in the documentation section.
    For the example, we kept the parameter at the default value of 20. Negative control files included an average of 2.4 times the filter words (despite showing a higher number of filter words, `31261533_x.pdf` did not include controls classifying it as a case-control study). The case-control papers displayed on average 55 times the filter words.
    Following parameter is saved in the TSV file: `filter.word.times`

    ```
    ## filter.word.times (MTX example):
    20
    ```

Parameters:

In this tab, specifications for the table extraction can be tweaked to ensure optimal output files (Fig. 2.D).

15. **Enter table headings:** Standard scientific articles have their tables labeled with "TABLE", "TAB", "Table" or "table" plus number and are detected accordingly. If a table is expected to have a different heading, it should be typed in this field. For multiple different heading use ";" without extra spaces.
    For most scientific papers, this option is not necessary to be populated as it is of greater use in extracting tables from non-journal articles. Accordingly, for the example the field was left empty.
    Following parameter is saved in the TSV file: `table.heading.words`

    ```
    ## table.heading.words (MTX example):
    [blank]
    ```

16. **Table heading case sensitive:** E.g., for "HEADING", if **no** was chosen then "HEADING", "heading", "Heading", etc., will be detected, if **yes** was chosen only "HEADING" will be detected.
    Irrelevant for the example, as table heading was left blank.
    Following parameter is saved in the TSV file: `ignore.case.th`

17. **Column pixel deviation:** For some tables the heading is slightly indented which would make the algorithm assume it was a separated column. With the column pixel deviation the size of indention which would be considered the same column can be adjusted. Following parameter is saved in the TSV file: `dev_x`

```
## dev_x (MTX example):
20
```

18. **Row pixel deviation:** For some tables elements even though in the same row can have slightly different vertical coordiates. With the row pixel deviation the variation of vertical coordinates which would be considered the same row can be adjusted. It can be either a number or set to dynamic detection [9999], in which case the font size is used to detect which words are in the same row. Following parameter is saved in the TSV file: `dev_y`

```
## dev_y (MTX example):
9999 ## [dynamic]
```

Documentation/Debugging:

Beyond the standard sentence and table output files, further documentation files can be generated (Fig. 2.E). Additionally, suspending the deletion of intermediate files can narrow down data extraction problems.

19. **Table values in file:** When **tables** detection/export is chosen, this option will be relevant. For **yes**, a separate file with the headings of all tables, their relative location in the generated HTML and TXT files, as well as information if search words were found will be generated. The files will start with "htmltablelines", "txttablelines", "keeplayouttablelines" followed by the PDF file name and can be found in html.docu, txt.docu, keeptxt.docu subfolders.
    This option is commonly not necessary to be selected. Nonetheless, it helps to identify if the PDE detects the tables and, if yes, if they are exported. When comparing the files starting with the PDF file name followed by "htmltablelines", "txttablelines", "keeplayouttablelines", it can be observed that all detected tables contained at least one of the search words.
    Following parameter is saved in the TSV file: `write.table.locations`

```
## write.table.locations (MTX example):
yes
```

20. **Export tables with problems:** For **yes**, if a table was detected in a PDF file but is an image or cannot be read, the page with the table will be exported as a portable network graphics (PNG) file. The documentation file will have the name format: [PDF file name]page[page number]w.table-[page number].png
    This is recommended to capture all tables, even if the program cannot detect the table content. This applies especially, for older articles with scanned tables.
    Following parameter is saved in the TSV file: `exp.nondetc.tabs`

```
## exp.nondetc.tabs (MTX example):
yes
```

21. **Table documentation files?:** For **yes**, if search words are used for table detection and no search words were found in the tables of a PDF file, a file will be created with the PDF file name followed by "no.table.w.search.words" in the folder with the name no_tab_w_sw.
    For completeness of the example, "`yes`" was chosen. Generally, it is safe to assume that papers without a file being created were sorted out due to a lack of search words or filter words.
    Following parameter is saved in the TSV file: `write.tab.doc.file`

```
## write.tab.doc.file (MTX example):
yes
```

22. **Sentence documentation file?:** For **yes**, if no search words were found in the sentences of a PDF file, a file will be created with the PDF file name followed by "no.txt.w.search.words" in

the no_txt_w_sw folder. If the PDF file is empty, a file will be created with the PDF file name followed by "non-readable" in the nr folder. Files that were filtered out using the filterwords will lead to the creation of a file with the PDF file name followed by "no.txt.w.filter.words" in the excl_by_fw folder.
Again, for completeness of the example, "`yes`" was chosen. This option does not influence the creation of the [id]_is_secured.txt file in the secured folder.
Following parameter is saved in the TSV file: `write.txt.doc.file`

```
## write.txt.doc.file (MTX example):
yes
```

23. **Do not delete intermediate files:** The program generates a txt, keeplayouttxt and HTML copy of the PDF file, which will be deleted if intermediate files deletion is chosen. In case, this option was chosen accidentally, the user has two options to delete the .txt and .html file. 1) Slow & easy option: Rerun the analysis with this option being yes. 2) Quick and slightly more complicated option: Open the file explorer and search for *.txt and *.html in the PDF folder. Then select all files and folders of the search result and press delete.
This option is primarily for debugging. Having access to the .txt and .html files will allow the identification of undetected tables/sentences or conversion issues.
Following parameter is saved in the TSV file: `delete`

```
## delete (MTX example):
no
```

No Output? Tab:

There are some common reasons why tables might not be extracted from a PDF file with the PDE_analyzer. The "No Output?" tab highlights the five suggestions to troubleshoot any problem. The recommendations include: (1) making sure that the tables in the PDF file have a standardized header or that the custom heading field is used under the Parameter tab, (2) ensuring that the PDF file is not secured or image file only, (3) double checking the filterwords and their threshold, (4) verifying the presence of the search words in the table, and (5) contacting the package maintainer for persistent issues.

Start the PDE_analyzer()

During the analysis, the progress bar indicates the number of files analyzed, while the drop down menu and the R console display status updates (Fig. 2.F). The messages printed in the console can be suppressed by loading `PDE_analyzer_i(verbose=FALSE)`:

```
Following file is processing: '00000000_x.pdf'
XpdfReader installed.
00000000_x.pdf has no readable content
Analysis of '00000000_x.pdf' complete.
Following file is processing: '29973177_!.pdf'
XpdfReader installed.
58 filter word(s) were detected in 29973177_!.pdf.
4 table(s) with search words found in '29973177_!.pdf'.
43 sentences with search words were found in '29973177_!.pdf'.
Analysis of '29973177_!.pdf' complete.
Following file is processing: '31261533_x.pdf'
XpdfReader installed.
'31261533_x.pdf' was filtered out due to a lack of the filter
 words. 9 filter word(s) were detected
Analysis of '31261533_x.pdf' complete.
```

Figure 3. The results of the example analysis opened in the `PDE_reader_i()` on Windows.

```
Following file is processing: '99999999_x.pdf'
XpdfReader installed.
99999999_x is most likely secured and cannot be processed!"
Analysis of '99999999_x.pdf' complete.
Analyses are complete.
```

As mentioned above, the resulting files should be identical to the files found in `examples/MTX_output_linux`, `examples/MTX_output_mac`, or `examples/MTX_output_win`.

View the results with the PDE_reader_i()

1. To open the `PDE_reader_i()` run:

   ```
   R> library("PDE")
   R> PDE_reader_i()
   ```

   This should open a user interface, i.e., a window with feather icon in task bar (Fig. 3).

Load and open:

2. Load either a sentence analysis file, such as `31083238_!_txt+-0__M_m_ethotrexate,_T_t_rexal,_R....csv` or a whole folder folder, e.g., `examples/MTX_output_linux`, `examples/MTX_output_mac`, or `examples/MTX_output_win`. The table shown in the center of the application is writable, editable and copyable, but changes will not be saved in the original file.
   NOTE: Analysis files refer to the files created by the `PDE_analyzer()` which contain "txt+-" in their file name.

- **Open analysis file:** Loading a single file is the quickest option and will open the selected file in the reader. The file will also be added into the memory of the program until the program is closed. All other analysis files in the folder will be shown under **13. Jump to file**.
- **Load analysis folder:** This will load all analysis files into the memory. For larger number of files, the progress bar on the top right will indicate the progress and indicate when the program can be accessed again. All files will be shown under **Jump to file** and are quickly accessible, since they are in the memory.

3. **Save memory to file:** The table and all tables that are currently displayed by the `PDE_reader_i()` during a session are saved into the memory of the program, enabling quick browsing through the tables with minimal loading time. Since the memory is reset anytime the program closes, the memory can be saved into a .RData file to prevent long loading times during later sessions.
4. **Load memory from file:** Tables that were saved into the memory during earlier sessions can be loaded into the program from a corresponding .RData file.
5. **Reset form:** The form and memory can be emptied with this button.
6. **On/off:** Search word highlighting can be turned off and on using this button in case an appropriate TSV file is loaded (see 8. Load TSV file). The button shows the current state of highlighting. The search word is found between ▶ and ◀ .
7. **Load all:** The options **Load TSV file** and **Open analysis file** will only save the already displayed tables into the memory. To shorten loading times, all analysis files in the current folder, as well as their search word highlighted tables, can be loaded into the memory with this button. This will load the tables with and without highlighting to allow rapid switching between the two. The green bar on the top right will display the progress.
   NOTE: In the case of high numbers of search words or analysis files, this step can take a long time (e.g., 1500 analysis files + 400 search words -> 1.5 h). For this reason, saving the memory to a file once the files are loaded is recommended.
8. **Load TSV file:** Open the TSV file, such as `PDE_parameters_v1.0_all_files+-0.tsv` in the folder `examples/tsvs/` to highlight the search words in following way: -[search word]-.
9. **Load PDF folder:** To enable the **Open current PDF** as well as **Extract tables** button load the PDF folder into the reader, i.e., `examples/Methotrexate/`.
10. The name of the current PDF file will show to the left of the **Open current PDF** button below the load PDF folder row.
11. **Open current PDF:** If a PDF file analyzed is detected in the PDF folder, pressing the button will open the PDF file in the system default PDF viewer.
12. **Extract tables:** This button allows the user to extract all tables from the current PDF file converting them into an Excel-compatible format. Extraction parameters such as pixel deviation between columns (see Detailed guide - `PDE_analyzer()` user interface: 17. Column and row pixel deviation) are derived from the TSV file (see 8. Load TSV file) chosen for search word highlighting. The extraction of the tables usually takes a few seconds, and, after extraction, the destination folder (same as analysis file folder) of the extracted tables is opened.
    For the example, we extracted all tables from the detected PDF file (since each table had either the word Methotrexate or MTX in it). The button can still be pressed though to watch the program extract all tables into a new subfolder named `extracted_tables` which can be found in the PDF folder. Figure 4 represents an exemplary extraction of a table from the PDF file.
    NOTE: The table extraction only works when PDF file and TSV files are available.
13. **Jump to file:** Instead of going from one file to the next, the user can also quickly jump to a file through the drop-down menu.

Table display settings:

14. **Font size:** The font size of all buttons, the labels and the table can be increased (+), decreased (-) or reset (o) located above the table.
15. **Hotkey mode:** There are 4 different hotkey modes (see Table 1), which allow the use of the buttons of a keyboard to quickly navigate through files. The hotkeys for each mode are as follows and can be changed by clicking on the botton on the right of the hotkey mode label:

Table 1. Assignments of key board buttons to the hot key modes.

	standard	**one hand**	**one hand & standard (oh+std)**	**no hotkey**
previous	n or left arrow	j or a	j or a or n or left arrow	none
next	p or right arrow	l or d	l or d or p or right arrow	none
scroll table down	down arrow	k or s	k or s or down arrow	none
scroll table up	up arrow	i or s	i or s or up arrow	none
flag file	f	h or f	h or f	none
x mark file	x	space	space or x	none
unmark file	u	u or e	u or e	none
open PDF	o	o or q	o or q	none

16. **Wrap:** When choosing this option, located on the right above the table, the text in the central table will have line breaks to be fully visible. This will prevent in some occasions the resizing of the window. To prevent this issue, choose **don't wrap** while resizing and activate resizing afterwards, again. In case the table does not fit vertically inside the window, the scroll bar can be used to show different rows of the table.
17. **Sentence number:** If sentences surrounding the sentence containing the search word were extracted by the `PDE_analyzer()` (i.e., `context` > 0), the number of sentences displayed can be decreased (-), increased (+), or reset (o). When changing this setting, the sentences with the search word will always be displayed.
18. **Show txtcontent only:** Generally, the analysis file includes information about the page and paragraphs from where the sentences were extracted. When selecting **Show txtcontent only**, only the sentences without the positional information are displayed.
19. **Show original text (abbreviations collapsed):** Choosing this setting will restore the original sentences by replacing ABBREV (search word)$* with ABBREV, e.g., MTX (Methotrexate)$* with MTX. This setting will also lead to the disappearance of some search words, as only the abbreviations remain.
20. The table shown in the center of the application is writable, editable and copyable, but changes will not be saved in the original file.

Browse and mark:

21. **Prev and Next:** Using these buttons, the user can quickly browse through the different tables in the analysis folder.
22. **Flag file:** Using this button, the user can either **mark analysis file only**, **mark PDF file only** or **mark analysis file & PDF**. The reader will rename the corresponding file adding a "!_" to the beginning of its name.
    NOTE: Make sure to have selected the file type (analysis file +- PDF file) that should be marked.
23. **X mark file:** Using this button, the user can either **mark analysis file only**, **mark PDF file**

A

Table 2 Human chorionic gonadotropin levels (mIU/ml) in patients not treated with MTX (Group 1), treated with MTX by intramuscular injection (Group 2) or treated with MTX by intravenous injection (Group 3)

	Mean ± SD/Rate			*P* value
	Group 1(*n* = 47)	Group 2(*n* = 46)	Group 3(*n* = 14)	
A:Before MTX treatment	–	44,603.28 ± 38,615.27	25,648.86 ± 16,125.92	0.080
B:After MTX treatment or before curettage	31,163.48 ± 28,626.86	37,712.91 ± 23,880.85	31,005.21 ± 23,029.37	0.196
C:Three days after curettage	2838.68 ± 3.10	4434.98 ± 4.90	3263.46 ± 2.93	0.077
B-A	–	9664.21 ± 8993.10	22,506.29 ± 36,916.63	0.219
(B-A)/A	–	0.27 ± 0.213	1.33 ± 3.451	0.273
C/B	0.085 ± 0.04	0.084 ± 0.05	0.79 ± 0.04	0.674

B

	A	B	C	D	E
1	Table 2 Human chorionic gonadotropin levels (mIU/ml) in patients not treated with MTX (Group 1), treated with MTX by				
2	intramuscular injection (Group 2) or treated with MTX by intravenous injection (Group 3)				
3		Mean ± SD/Rate			P value
4		Group 1(n = 47)	Group 2(n = 46)	Group 3(n = 14)	
5	A:Before MTX treatment	–	44,603.28 ± 38,615.27	25,648.86 ± 16,125.92	0.08
6	B:After MTX treatment or before curettage	31,163.48 ± 28,626.86	37,712.91 ± 23,880.85	31,005.21 ± 23,029.37	0.196
7	C:Three days after curettage	2838.68 ± 3.10	4434.98 ± 4.90	3263.46 ± 2.93	0.077
8	B-A	–	9664.21 ± 8993.10	22,506.29 ± 36,916.63	0.219
9	(B-A)/A	–	0.27 ± 0.213	1.33 ± 3.451	0.273
10	C/B	0.085 ± 0.04	0.084 ± 0.05	0.79 ± 0.04	0.674

Figure 4. Exemplary results of the pdf2table function included in the `PDE_analyzer()`. The original table (A) was obtained from ***Wang et al.*** (***2018***) and extracted using the `PDE_analyzer_i()` with standard parameters. Merely the column width for the `PDE`-extracted table (B) displayed in Microsoft Excel was adjusted with no modifications to the table content.

only or **mark analysis file & PDF**. The reader will rename the corresponding file adding a "x_" to the beginning of its name.

24. **Unmark file:** Using this button, the user can either **unmark analysis file only**, **unmark PDF file only** or **unmark analysis file & PDF**. The reader will remove and existing "!_" or "x_" at the beginning of the file name.
    NOTE: Flagging and marking changes file names but can be reversed in the program at any time.

Efficacy comparison to a Cochrane systematic review

To assess the sensitivity and specificity of the `PDE_analyzer()`, we reproduced the results of a published Cochrane systematic review by ***Kraal et al.*** (***2017***). The objective of ***Kraal et al.*** (***2017***) was to assess the efficacy and adverse effects of ^{131}I-meta-iodobenzylguanidine (^{131}I-MIBG) therapy in patients with newly diagnosed high-risk (HR) neuroblastoma (NBL) through a systematic literature review. Therefore, the group searched the Cochrane Central Register of Controlled Trials (CENTRAL; the Cochrane Library 2016, Issue 3), PubMed (1945 to 25 April 2016) and Embase (Ovid) (1980 to 25 April 2016) for articles on ^{131}I-MIBG and HR NBL. All article titles and abstracts were manually evaluated by ***Kraal et al.*** (***2017***) and eligible articles were assessed in the full-text.

For the partial reproduction of the work by ***Kraal et al.*** (***2017***), we used the MeSH headings and text words described by the group in a PubMed search of relevant articles. We downloaded open access papers using the Pubmed Batch Downloader by ***Greenwald*** (***2019***) and obtained articles with restricted access through the Texas Medical Center Library using the OpenAthens plugin in EndNote. Using the `PDE_analyzer_i()` interactive interface, we applied the variable parameters

described in Table 2 to the analysis. To exhaust the full potential of the `PDE_analyzer()`, we decided to extract sentences as well as tables with the search words. The XPDF (***Noonburg, 1996–2020***) command line tools were obtained from https://www.xpdfreader.com/download.html. Efficacy and adverse effects described under the "Types of outcome measures" heading in ***Kraal et al.*** (***2017***) were translated into specific filter/search words. Different variants of a word were indicated by the regular expression for OR (|). To detect the filter and search words at the beginning of sentences (captilized), within sentences (lower case) and in headings (all-caps), `ignore.case.fw]` and `ignore.case.sw]` were set to `TRUE`, i.e., the case should be ignored. For the exemplary literature review, the standard filter word threshold of `20` was chosen. This is an empirically determined number obtained from the analyses of several thousand articles. Search words for text and table extraction were chosen to be identical to the filter words, since the sentences and tables with the search words provide the most information on the relevance of an article for a systematic review. Two sentences before and after the sentence containing the search word were extracted. This allowed, to evaluate the context as well as focus on the relevant sentence with the `PDE_reader_i()`. Specifically, words like `bone marrow` or `blood pressure` had the potential to be abbreviated. Accordingly, any type of detected abbreviation of the words should be counted as an incidence of the word. The standard pixel deviation for indentations in table columns is `20` and was chosen for this representative analysis accordingly. All tables were exported as Microsoft Excel-compatible comma-separated values (`.csv`) files. Additional information on table locations was not required. Specifically, 90 degrees-rotated tables not extracted by the `PDE_analyzer()` were selected to be exported as PNG files, i.e., `exp.nondetc.tabs -> TRUE`. To evaluate if all files were correctly processed, additional table and text detection documentation files were exported, i.e., `write.tab.doc.file -> TRUE` & codewrite.txt.doc.file -> TRUE. Intermediate files were of no relevance and were therefore deleted after analysis.

While the PubMed search yielded 2291 results out of the 3366 articles of which the titles and abstracts were screened by ***Kraal et al.*** (***2017***), 1262 full-text articles in PDF format were obtained and used for analysis by the `PDE_analyzer()` (see Fig. 5). The analysis took approximately 1.5 hours (an average of 4 seconds per articles) and yielded 762 articles containing the filter/search words detected at least 20-times and 26 articles without containing text that could be processed, i.e., secured files or scans of full articles. Accordingly, over one third of the articles (39.6%, 500/1262) were excluded without the requirement of a manual review. The 762 articles not excluded encompassed 38/39 (97.4%) of the PubMed-derived articles that ***Kraal et al.*** (***2017***) assessed for eligibility. The 39 evaluated articles contained on average, 89 detected instances of filter words, with a range of 11 to 253, and an average of 53 sentences with search words extracted, ranging from 13 to 133. The two articles by ***de Kraker et al.*** (***2008***), and ***Kraal et al.*** (***2015***), finally included in the Cochrane systematic review displayed the filter words 72 and 89 times, with 50 and 31 sentences extracted, respectively.

Interestingly, the article by ***Mastrangelo*** (***1987***) was further evaluated by ***Kraal et al.*** (***2017***) but then excluded from the review due to a lack of primary data. The `PDE_analyzer()`, on the other hand, excluded the article during initial analysis based on the number of filter word number (fw=11 was below the threshold of 20) (see Supplemental Data Table 1). Furthermore, direct comparison of abstract- and title-only review and the `PDE` package showed that 97.2% (70/72) of the search words were located within the body article, only two search words in the abstract and none in the title (Fig. 6). The distribution of search words was even throughout the article with 4/5 figures/tables including search words. While 59 search words were located in the body of the article, 11 search words were in the references. The search term "surviv" was most common with 10 incidences (8 survival, 1 surviving, 1 survivors) although absent in title and abstract.

In conclusion, the `PDE` package notably reduced the number of articles required to be manu-

Table 2. Parameters used for the reproduction of ***Kraal et al.*** (***2017***) with the `PDE_analyzer()`.

variable	value
whattoextr	txtandtab
pdftotextfile, pdftohtmlfile, pdftopngfile	xpdf-tools-win-4.02 was used on a windows machine in a 64-bit environment
filter.words	outcome;efficacy;effective;clinical sign;adverse effect;side effect;surviv;toxicity;secondary malignancy;cardio;hepat; sialoadenitis;myelo;neutrophil;anaemi;leukemi(a|c);stem cell;haemato;thrombo; haemoglobin;platelet;thyroid;bone marrow;bonemarrow;blood pressure;hypertension; biodistribution;liver;organ uptake;renal uptake
ignore.case.fw	TRUE
filter.word.times	20
search.words	outcome;efficacy;effective;clinical sign;adverse effect;side effect;surviv;toxicity;secondary malignancy;cardio;hepat; sialoadenitis;myelo;neutrophil;anaemi;leukemi(a|c);stem cell;haemato;thrombo; haemoglobin;platelet;thyroid;bone marrow;bonemarrow;blood pressure;hypertension; biodistribution;liver;organ uptake;renal uptake
ignore.case.sw	TRUE
context	2
eval.abbrevs	TRUE
dev_x	20
dev_y	9999
out.table.format	.csv (WINDOWS-1252)
write.table.locations	FALSE
exp.nondetc.tabs	TRUE
write.tab.doc.file	TRUE
write.txt.doc.file	TRUE
delete	TRUE

ally reviewed with high accuracy, provided additional preprocessed content for the assessment of articles, i.e., extracted sentences with search words, and evaluated articles in a reproducible manner easily applied to additional literature. An evaluation of sensitivity and specificity of the `PDE_analyzer()` is challenging since we observed that with higher thresholds for filter words, the selectivity is increased, whereas the number of articles assessed for eligibility in this pool is reduced although both included articles are still detected, e.g., a filter word threshold of 50 resulted in 468 articles assessed for eligibility with and overlap of 19/39 full-text articles assessed by ***Kraal et al.*** (***2017***) and both included articles still being detected (Fig. 7).

Discussion

The `PDE` R package accelerates the systematic review of literature and simplifies the extraction of tables from PDF files. The `PDE_analyzer()` allows presorting of articles according to filter words, auto-replacement of abbreviations, and easy extraction of sentences and tables with user defined search words. We decided to use XPDF (***Noonburg, 1996–2020***) command line tools `pdftotext` and `pdftohtml` for obtaining content and positional information in order to maintain cross-compatibility of the `PDE` package for Microsoft Windows, Mac and Linux systems. Additionally, `pdftotext` and `pdftohtml` perform especially well in the detection and correction of hyphenation in words at the

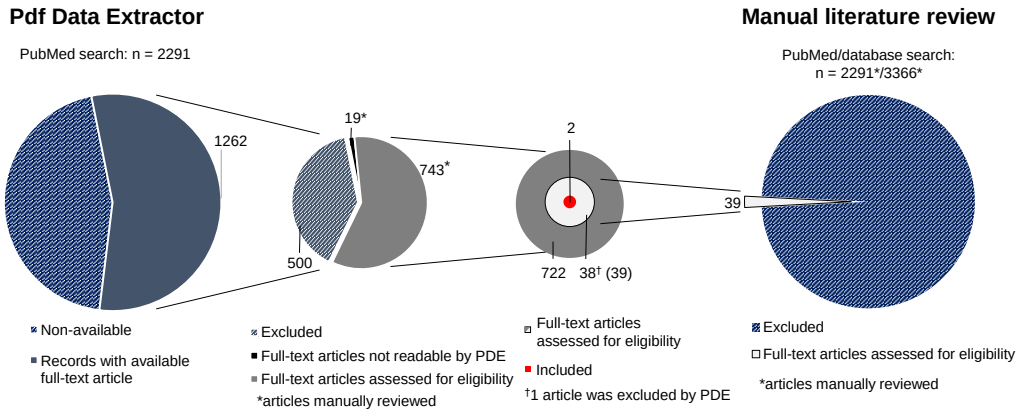


Figure 5. Schema representing the article numbers resulting from each step of either the analysis using the `PDE_analyzer()` (left) or the manual literature review applied by ***Kraal et al.*** (***2017***) (right). Both approaches yielded the same number of true positives.

end of the line and unequally spaced words.

Compared to other R packages capable of table extraction such as `pdftools` (***Ooms, 2019***) or `tabulizer` (***Leeper, 2018***), the `PDE_analyzer()` allows the reliable automatic detection of tables including headers in scientific articles. The output tables are saved by the `PDE_analyzer()` in universal comma-separated values (.csv) or tab-separated values (.tsv) files, in contrast to a list of word positions (`pdftools`). The `PDE` package can enhance the literature search for review articles, gene or disease curation, risk factor analysis, and general literature reviews. Additionally, data analysis can be increased by easy extraction of tables through the `PDE_pdf2table()` function.
Through the implementation of the `tcltk` (***R Core Team, 2019***) and `tcltk2` (***Grosjean, 2019***) packages, the user is provided with the `PDE_analyzer_i()` interactive GUI. The user interface not only allows the rapid selection of parameters and monitoring of active analyses but also the storage of the parameters used for an analysis in a tab-separated values (TSV) file. The file stores filter and search words alongside file locations, analysis thresholds, and versions of the dependent software tools. In contrast to machine learning algorithms, all parameters have a functional connection to the analysis and are easily comprehensible in their impact on the results.

The `PDE_reader_i()` is an additional GUI which allows quick browsing through the extracted sentences, opening of the full-text PDF files and obtaining tables from the current PDF file in a Microsoft Excel readable file format. The user interface also allows the flagging or marking of articles by adding a prefix, i.e., "!_" or "x_", to the file name. The prefix can also be easily removed at any time. This is especially useful as it does not require a separate documentation file and allows the interruption of evaluations at any time.

Demonstrated in **Section - Efficacy comparison to a Cochrane systematic review** through the partial reproduction of a Cochrane Systematic Review (***Kraal et al., 2017***), the `PDE_analyzer()` and `PDE_reader_i()` proved to provide tools for quick literature review. The `PDE_analyzer()` displayed an average of 4 seconds processing time per PDF file using 30 filter and search words. For the `PDE_reader_i()`, virtually no delay was detected when browsing through the preloaded analysis result files. Nonetheless, the systematic review of full-text PDF files in contrast to abstract- and title-only reviews requires generally a significant amount of time gaining access and downloading full-text PDF files. Even though time intensive, full-text analyses display clear advantagesover an abstract- and title-only review as conducted by ***Kraal et al.*** (***2017***), such as more sensitive detection of negative results, exclusion criteria, and data in multiaspect studies. We are aware that the

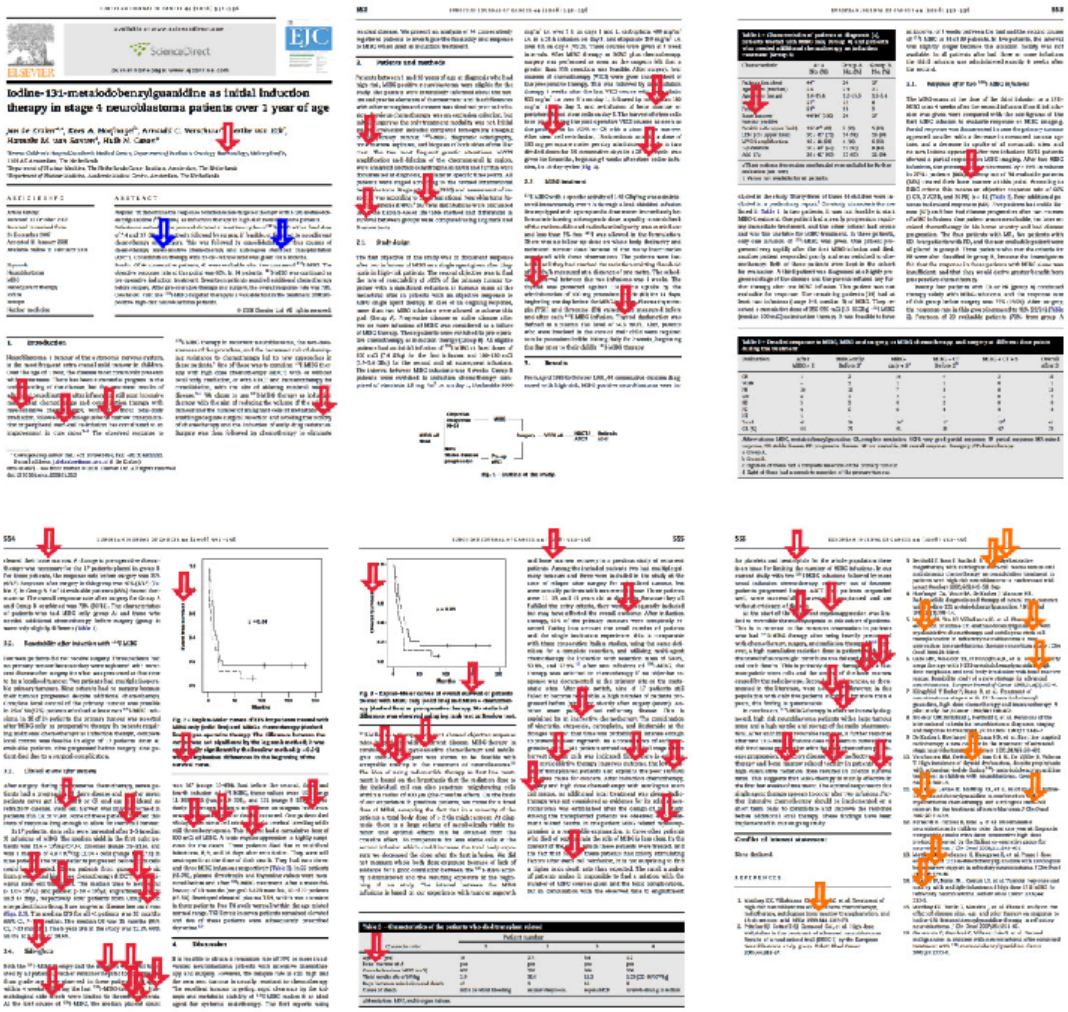


Figure 6. Overview of the full-text article by ***de Kraker et al.*** (***2008***) with highlighted search words. Blue arrows indicate the 2 search words identified in the abstract; red arrows point to the 59 search words found in the body of the article; orange arrows highlight the 11 search words detected in the references. For the list of search words see Table 2.

review of full-text articles can only be applied to a subset of the literature as some articles are not readily available. However, with standard parameters, the `PDE_analyzer()` was able to exclude 522 (35.4%) articles from the required manual review with high sensitivity (Fig. 7) resulting in a significant reduction of review time and reducing the possibility of human errors. Full-text searches are especially recommended for side notes or methodologies generally not mentioned in the abstract and have the potential of adding another layer of sensitivity to the literature review. Hence, the `PDE_analyzer()` is not meant to replace the manual review of all articles but rather reduce the number of manually assessed articles at an early stage and distill potentially relevant paragraphs. Since the `PDE` package does not change any PDF files, articles deemed relevant remain compatible with any downstream review softwares such as the Review Manager (***The Nordic Cochrane Centre, 2020***) or the EpiTools epidemiological calculator (***Sergeant, 2018***).

We recognize that the `PDE_analyzer()` depends highly on the integrity of the PDF file, cannot process secured or image-only PDF files, and cannot convert tables rotated by 90 degrees into .csv or .tsv files. Nevertheless, documentation files are created for secured, non-readable or image-only PDF files or tables. When choosing the filter word threshold for the `PDE_analyzer()`, the user has

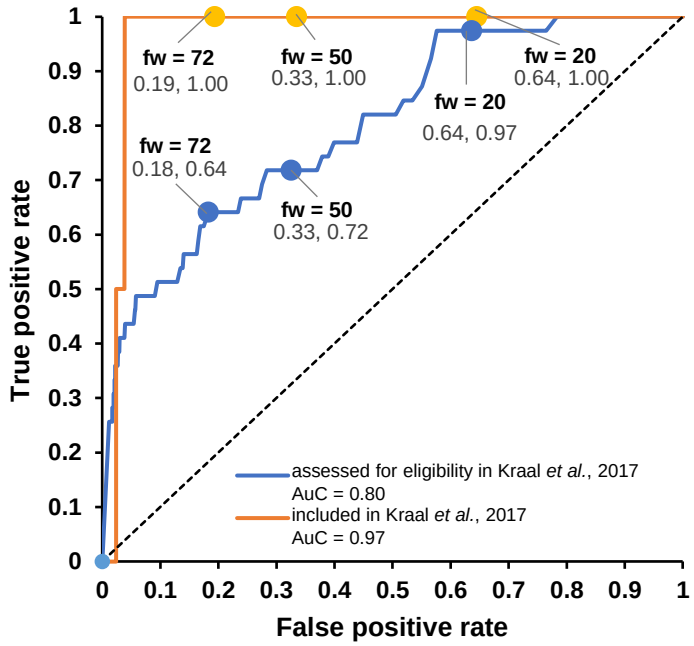


Figure 7. Receiver operating characteristic (ROC) curves of the article detection by the `PDE_analyzer()`. ROC curves for the detection of the articles assessed for eligibility in ***Kraal et al.*** (***2017***) (blue) and the articles included in the review by ***Kraal et al.*** (***2017***) (orange) were compared to the reference line (dashed line). Variable parameter was the filter word threshold (fw) with different thresholds indicated by solid circles. The area under the curve (AuC) for each evaluation is indicated in the legend.

Figure 7–source data 1. The zip archive contains the `PDE_analyzer()` output and quatification used to create figure 7. Filter word numbers were extracted from the output. Specificity and sensitivity at each filter word threshold were determined from extracted data. The corresponding Excel sheet is named "Figure 7-Source Data 1.xlsx".

to be aware that threshold signifies an absolute number rather than a fraction of the total words in the article, therefore being sensitive the the article length. While our analyses continously showed that relevant articles contained a filter word number significantly above the average independant of length, there remains a risk of underdetection of short articles, such as letters, editorials, and mini-reviews. Hence the user is encourage to deactivate filter words and assess articles with the `PDE_reader_i()`, when detection small details or brief descriptions is the goal. Lastly, the exemplary comparison of the abstract- and title-only review and the `PDE` package revealed that a notable fraction of the filter words (15.2% for ***de Kraker et al.*** (***2008***)) were located in the references section (see Fig. 6). While this still indicates the general topic of the article, the risk of false positives arising from reference inclusion is evident and consequently, features intended to be implemented in future updates of the `PDE` package include the voluntary exclusion of the reference section for search word detection and the creation of a `PDE_quantifier()`. The `PDE_quantifier()` would allow the user to extract metadata from the analyses such as number of filter/search words per article, relative abundance of each filter/search word, separation of filter/search words by section, and categorization based on multiple analyses.

Methods and Materials

Installation

Install the dependent packages

```
R> install.packages("tcltk2")
```

The package requires the XpdfReader software by Glyph & Cog, LLC. Please download and install the XpdfReader from the following website onto your local disk: https://www.xpdfreader.com/download.html. Alternatively, the following command can be used to install the correct XpdfReader:

```
R> PDE_install_XpdfReader4.02()    # Download and install the XpdfReader
R> PDE_check_Xpdf_install()        # Check if all required XPDF tools are installed
```

Install the package through CRAN

```
R> install.packages("PDE", dependencies = TRUE)
```

or choose the download location of the latest PDE_*.*.*.tar.gz and install it from a local path.

```
R> filename <- file.choose()
R> install.packages(filename, type="source", repos=NULL)
```

NOTE: The PDE package was tested on Microsoft Windows, Mac and Linux machines. Major differences include the visual appearance of the interfaces and the directory structures, but all functions are preserved.

Computational details

The results in this paper were obtained using R 3.6.0 with the PDE 1.3.0 package. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/package=PDE. The xpdf 4.02 command line tools were obtained from https://www.xpdfreader.com/download.html.

Acknowledgments

The authors want to thank Isabel F. Escapa, Tommy H. Tran, Andrea I. Lee, Katherine P. Lee, Erin C. Gregory-Peckham, and Jeremy Schraw for their help in testing the PDE package, troubleshooting problems and providing valuable feedback for the features.

Conflict of interest

Erik Stricker and Michael E. Scheurer declare that they have no conflict of interest.

References

Greenwald B. Pubmed-Batch-Download; 2019, https://github.com/billgreenwald/Pubmed-Batch-Download.git.

Grosjean P. SciViews-R: A GUI API for R. UMONS, MONS, Belgium; 2019, https://www.sciviews.org/SciViews-R.

Kraal KC, van Dalen EC, Tytgat GA, Van Eck-Smit BL. Iodine-131-meta-iodobenzylguanidine therapy for patients with newly diagnosed high-risk neuroblastoma. Cochrane Database Syst Rev. 2017; 4. https://www.ncbi.nlm.nih.gov/pubmed/28429876, doi: 10.1002/14651858.CD010349.pub2.

Kraal KC, Tytgat GA, van Eck-Smit BL, Kam B, Caron HN, van Noesel M. Upfront treatment of high-risk neuroblastoma with a combination of 131I-MIBG and topotecan. Pediatr Blood Cancer. 2015; 62(11):1886–91. https://www.ncbi.nlm.nih.gov/pubmed/25981988, doi: 10.1002/pbc.25580.

de Kraker J, Hoefnagel KA, Verschuur AC, van Eck B, van Santen HM, Caron HN. Iodine-131-metaiodobenzylguanidine as initial induction therapy in stage 4 neuroblastoma patients over 1 year of age. Eur J Cancer. 2008 June; 44(4):551–6. https://www.ncbi.nlm.nih.gov/pubmed/18267358, doi: 10.1016/j.ejca.2008.01.010.

Leeper TJ. tabulizer: Bindings for Tabula PDF Table Extractor Library; 2018, r package version 0.2.2.

Marshall IJ, Wallace BC. Toward systematic review automation: a practical guide to using machine learning tools in research synthesis. Syst Rev. 2019; 8(1):163. https://www.ncbi.nlm.nih.gov/pubmed/31296265, doi: 10.1186/s13643-019-1074-9.

Mastrangelo R. The treatment of neuroblastoma with 131I-MIBG. Med Pediatr Oncol. 1987; 15(4):157–158. https://www.ncbi.nlm.nih.gov/pubmed/3657703, doi: 10.1002/mpo.2950150403.

Noonburg DB. Xpdf: an open source PDF viewer and command line tools. Glyph & Cog, LLC; 1996–2020, https://www.xpdfreader.com/.

Ooms J. pdftools: Text Extraction, Rendering and Converting of PDF Documents; 2019, https://CRAN.R-project.org/package=pdftools, r package version 2.3.

R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria; 2019, https://www.R-project.org/.

Sergeant E. Epitools Epidemiological Calculators; 2018, https://epitools.ausvet.com.au.

The Nordic Cochrane Centre. Review Manager (RevMan). The Nordic Cochrane Centre, The Cochrane Collaboration, Copenhagen, Denmark; 2020, https://training.cochrane.org/online-learning/core-software-cochrane-reviews/revman.

Wang S, Beejadhursing R, Ma X, Li Y. Management of Caesarean scar pregnancy with or without methotrexate before curettage: human chorionic gonadotropin trends and patient outcomes. BMC Pregnancy Childbirth. 2018; 18(1):289. https://www.ncbi.nlm.nih.gov/pubmed/29973177, doi: 10.1186/s12884-018-1923-x.

appendix

Appendix 1

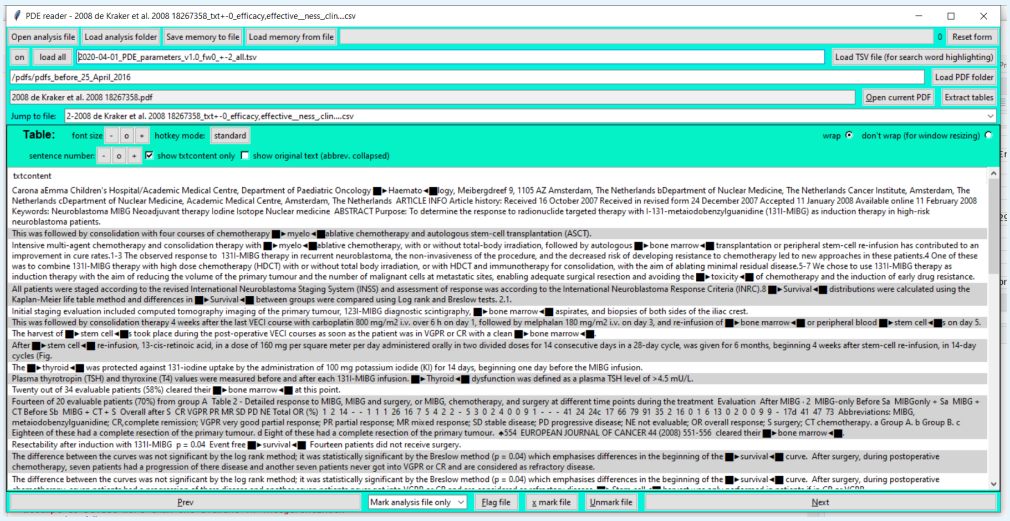


Figure 1. User interface of the `PDE_reader_i()` showing `PDE_analyzer_i()` results generated from the article by ***de Kraker et al.*** (***2008***) as outlined in the Section - Efficacy comparison to a Cochrane systematic review. Search words detected are marked through the `PDE_reader_i()` with solid boxes combined with arrow heads pointing towards the search word. The display of positional information was deactivated. The screenshot shows the first 20 out of 72 detected filter words.

PDE analyzer - Choose your variables

Load form from TSV | Save form as TSV | Reset form

Input/Output | Search Words | Filter Words | Parameters | Documentation | No Output?

Choose the PDF(s)/folder

Open folder | Load files

Choose the output folder

Open

Choose the output format

.csv (macintosh)

Start analysis | Close session | ^ save ^

PDE analyzer - Choose your variables

Load form from TSV | Save form as TSV | Reset form

Input/Output | Search Words | Filter Words | Parameters | Documentation | No Output?

Choose the PDF(s)/folder

C:/examples/Methotrexate | Open folder | Load files

Choose the output folder

C:/examples/MTX_output_test | Open

Choose the output format

.csv (WINDOWS-1252)

Start analysis | Close session | ^ save ^

PDE analyzer - Choose your variables

Load form from TSV | Save form as TSV | Reset form

Input/Output | Search Words | Filter Words | Parameters | Documentation | No Output?

Choose what to extract from the PDF files sentences ○ tables ○ both ◉

Extract all tables or only ones with search words? all ○ search words only ◉

Search words (separated by ; without extra spaces):

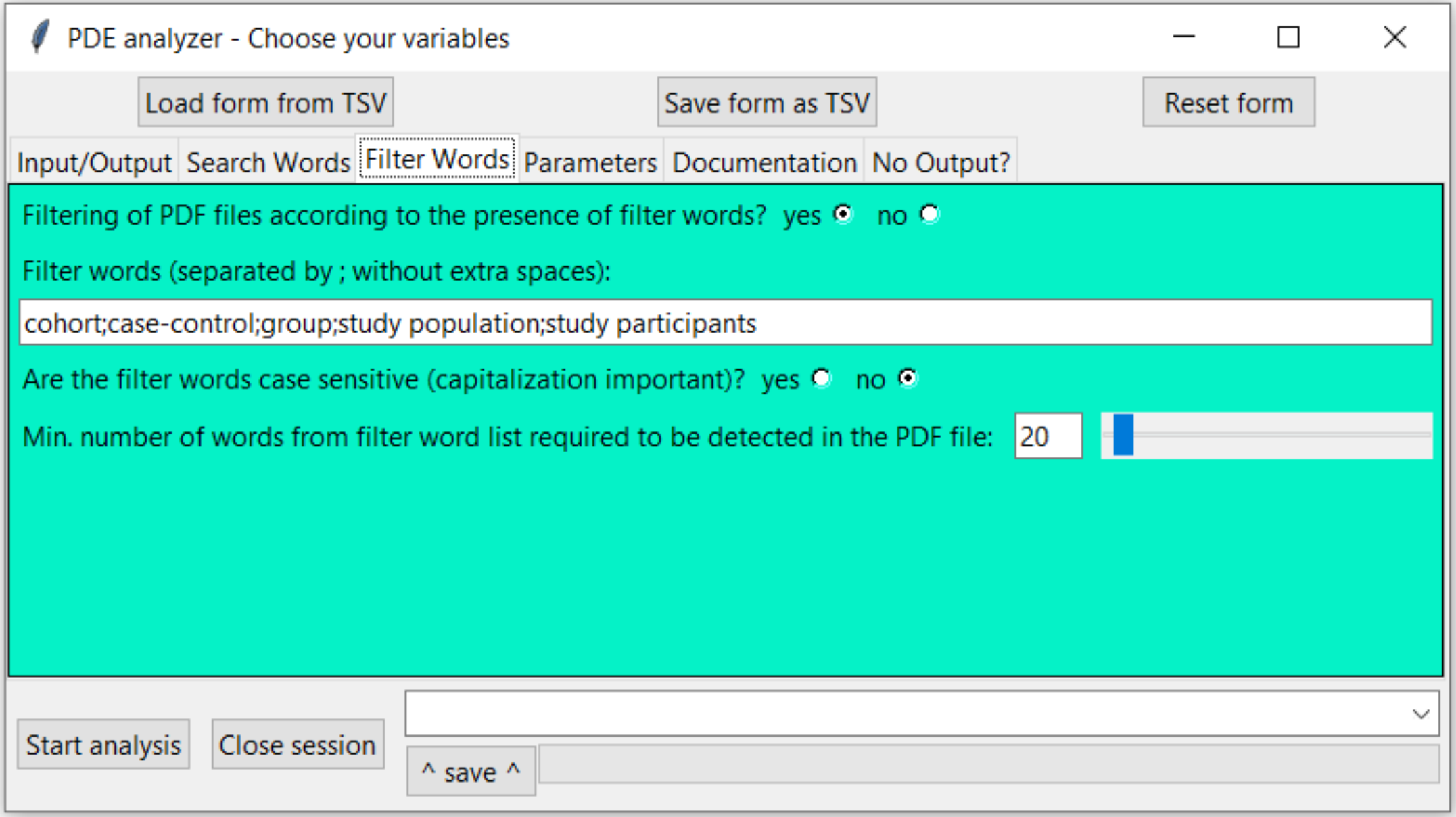
(M|m)ethotrexate;(T|t)rexal;(R|r)heumatrex;(O|o)trexup

Are the search words case sensitive (capitalization important)? yes ◉ no ○

Number of sentences extracted before and after the sentence with the search word: 0

Should abbreviations of the search word be replaced by the search word during the analysis? yes ◉ no ○

Start analysis | Close session | ^ save ^


PDE analyzer - Choose your variables
Load form from TSV
Save form as TSV
Reset form
Input/Output
Search Words
Filter Words
Parameters
Documentation
No Output?
Filtering of PDF files according to the presence of filter words? yes no
Filter words (separated by ; without extra spaces):
cohort;case-control;group;study population;study participants
Are the filter words case sensitive (capitalization important)? yes no
Min. number of words from filter word list required to be detected in the PDF file:
20
Start analysis
Close session
^ save ^

PDE analyzer - Choose your variables

Load form from TSV | Save form as TSV | Reset form

Input/Output | Search Words | Filter Words | Parameters | Documentation | No Output?

Enter all table headings other than "table" (separated by ; without extra spaces):

Are the table headings case sensitive (capitalization important)? yes no

Enter the pixel deviation where different table columns are considered same: 20

Enter the pixel deviation where different table value heights are considered the same row (enter 9999 for dynamic detection based on font size): dynamic 9999

Start analysis | Close session | ^ save ^

PDE analyzer - Choose your variables

Load form from TSV | Save form as TSV | Reset form

Input/Output | Search Words | Filter Words | Parameters | Documentation | No Output?

Should the table values (coordinates within the files) be written in an output file? yes ◉ no ○

Should the tables with problems exported as pngs? yes ◉ no ○

Should a table documentation file be created (file that reports if there was no table found)? yes ◉ no ○

Should a sentence documentation file be created (file that reports if there was no sentence found)? yes ◉ no ○

Should the intermediate files be kept (not deleted)? yes ○ no ◉

Start analysis | Close session | ^ save ^

PDE analyzer - Choose your variables

Load form from TSV | Save form as TSV | Reset form

Input/Output | Search Words | Filter Words | Parameters | Documentation | No Output?

Choose the PDF(s)/folder

C:/examples/Methotrexate

Open folder | Load files

Choose the output folder

C:/examples/MTX_output_test

Choose the output format

.csv (WINDOWS-1252)

Analysis of '31083238_!.pdf' complete.
60%, 31261533_x.pdf
Following file is processing: '31261533_x.pdf'
'31261533_x.pdf' was filtered out due to a lack of the filter words. 9 filter word(s) we
Analysis of '31261533_x.pdf' complete.
80%, 99999999_x.pdf
Following file is processing: '99999999_x.pdf'
99999999_x is most likely secured and cannot be processed!
Analysis of '99999999_x.pdf' complete.
complete

complete

Start analysis | Close session | ^ save ^

PDE analyzer - Choose your variables

Load form from TSV | Save form as TSV | Reset form

Input/Output | Search Words | Filter Words | Parameters | Documentation | No Output?

Choose the PDF(s)/folder

C:/examples/Methotrexate

Open folder | Load files

Choose the output folder

C:/examples/MTX_output_test

Choose the output format

.csv (WINDOWS-1252)

Analysis of '31083238_!.pdf' complete.
60%, 31261533_x.pdf
Following file is processing: '31261533_x.pdf'
'31261533_x.pdf' was filtered out due to a lack of the filter words. 9 filter word(s) we
Analysis of '31261533_x.pdf' complete.
80%, 99999999_x.pdf
Following file is processing: '99999999_x.pdf'
99999999_x is most likely secured and cannot be processed!
Analysis of '99999999_x.pdf' complete.
complete

complete

Start analysis | Close session | ^ save ^

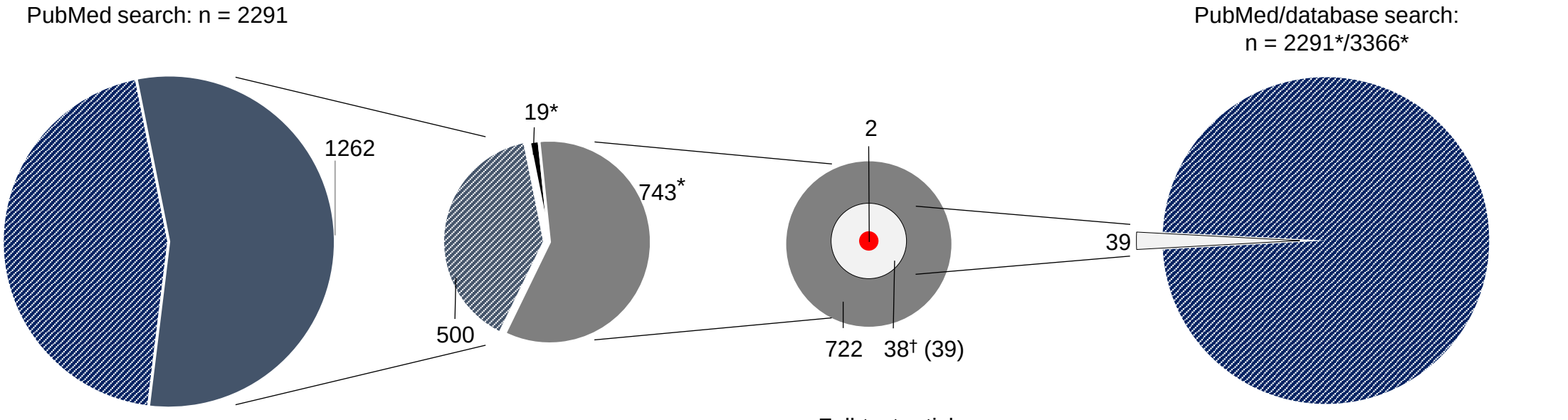
Table 2 Human chorionic gonadotropin levels (mIU/ml) in patients not treated with MTX (Group 1), treated with MTX by intramuscular injection (Group 2) or treated with MTX by intravenous injection (Group 3)

	Mean ± SD/Rate			*P* value
	Group 1(*n* = 47)	Group 2(*n* = 46)	Group 3(*n* = 14)	
A:Before MTX treatment	–	44,603.28 ± 38,615.27	25,648.86 ± 16,125.92	0.080
B:After MTX treatment or before curettage	31,163.48 ± 28,626.86	37,712.91 ± 23,880.85	31,005.21 ± 23,029.37	0.196
C:Three days after curettage	2838.68 ± 3.10	4434.98 ± 4.90	3263.46 ± 2.93	0.077
B-A	–	9664.21 ± 8993.10	22,506.29 ± 36,916.63	0.219
(B-A)/A	–	0.27 ± 0.213	1.33 ± 3.451	0.273
C/B	0.085 ± 0.04	0.084 ± 0.05	0.79 ± 0.04	0.674

	A	B	C	D	E
1	Table 2 Human chorionic gonadotropin levels (mIU/ml) in patients not treated with MTX (Group 1), treated with MTX by				
2	intramuscular injection (Group 2) or treated with MTX by intravenous injection (Group 3)				
3		Mean ± SD/Rate			P value
4		Group 1(n = 47)	Group 2(n = 46)	Group 3(n = 14)	
5	A:Before MTX treatment	–	44,603.28 ± 38,615.27	25,648.86 ± 16,125.92	0.08
6	B:After MTX treatment or before curettage	31,163.48 ± 28,626.86	37,712.91 ± 23,880.85	31,005.21 ± 23,029.37	0.196
7	C:Three days after curettage	2838.68 ± 3.10	4434.98 ± 4.90	3263.46 ± 2.93	0.077
8	B-A	–	9664.21 ± 8993.10	22,506.29 ± 36,916.63	0.219
9	(B-A)/A	–	0.27 ± 0.213	1.33 ± 3.451	0.273
10	C/B	0.085 ± 0.04	0.084 ± 0.05	0.79 ± 0.04	0.674

Pdf Data Extractor

Manual literature review



- Non-available
- Records with available full-text article

- Excluded
- Full-text articles not readable by PDE
- Full-text articles assessed for eligibility

*articles manually reviewed

- Full-text articles assessed for eligibility
- Included

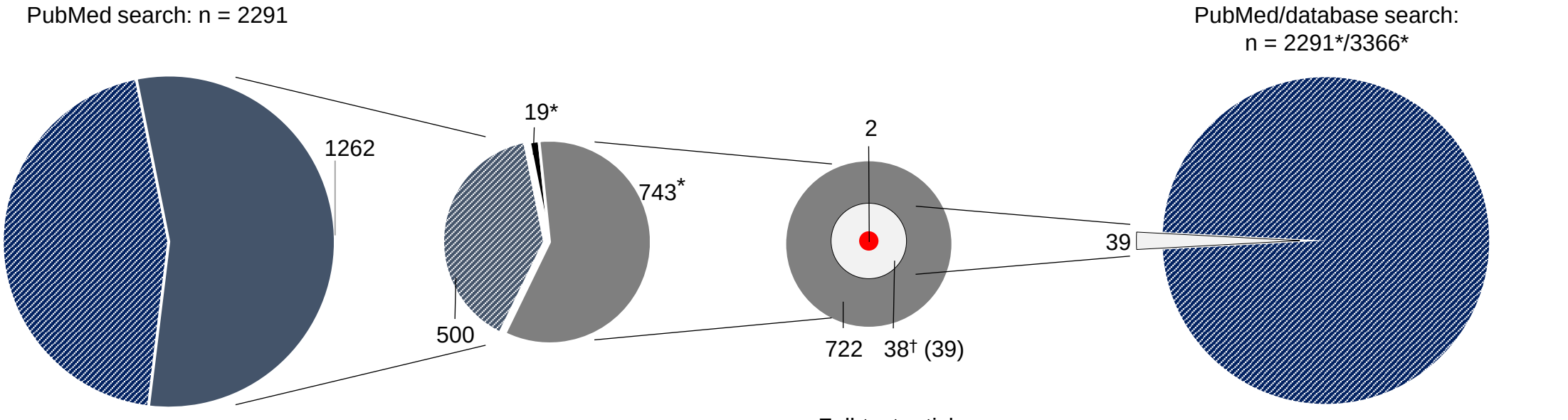
†1 article was excluded by PDE

- Excluded
- Full-text articles assessed for eligibility

*articles manually reviewed

Pdf Data Extractor

Manual literature review



- Non-available
- Records with available full-text article

- Excluded
- Full-text articles not readable by PDE
- Full-text articles assessed for eligibility

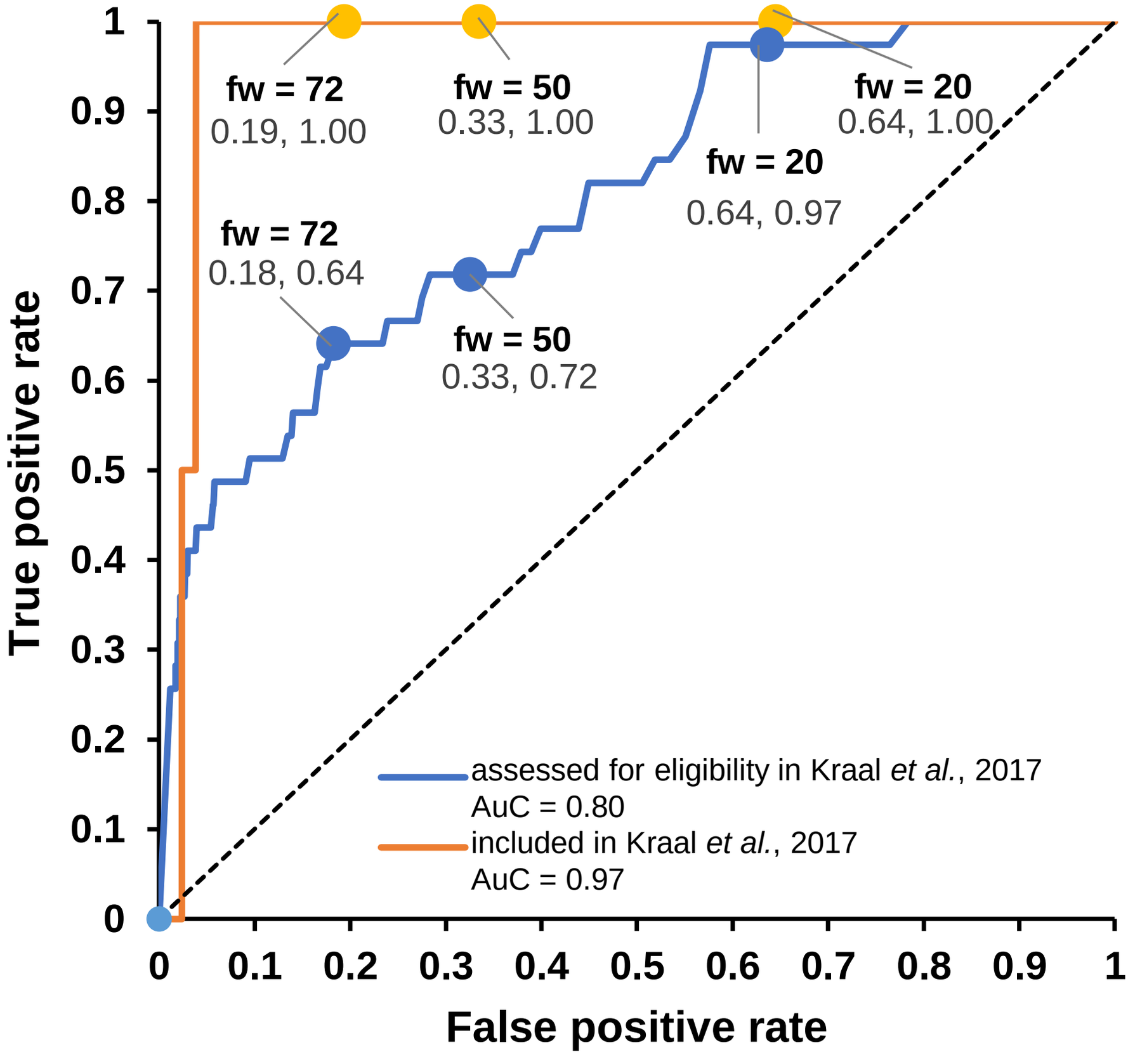
*articles manually reviewed

- Full-text articles assessed for eligibility
- Included

†1 article was excluded by PDE

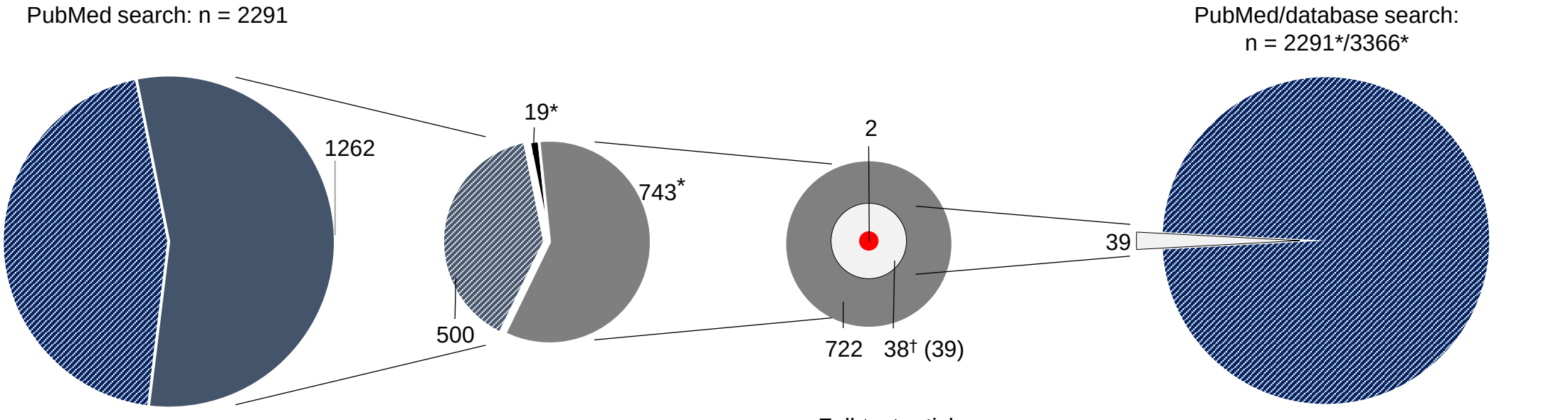
- Excluded
- Full-text articles assessed for eligibility

*articles manually reviewed


True positive rate
False positive rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
fw = 72
0.19, 1.00
fw = 50
0.33, 1.00
fw = 20
0.64, 1.00
fw = 20
0.64, 0.97
fw = 72
0.18, 0.64
fw = 50
0.33, 0.72
assessed for eligibility in Kraal et al., 2017
AuC = 0.80
included in Kraal et al., 2017
AuC = 0.97

Pdf Data Extractor

Manual literature review



- Non-available
- Records with available full-text article

- Excluded
- Full-text articles not readable by PDE
- Full-text articles assessed for eligibility

*articles manually reviewed

- Full-text articles assessed for eligibility
- Included

†1 article was excluded by PDE

- Excluded
- Full-text articles assessed for eligibility

*articles manually reviewed