Extracting data from vector figures in scholarly articles

CHJ Hartgerink (Tilburg University, [c.h.j.hartgerink@uvt.nl](mailto:c.h.j.hartgerink@uvt.nl))

14 July, 2017

# Introduction

Information from figures in scholarly articles can be harvested for the underlying data (see also the preceding "In Brief" report; Hartgerink, C. H. J. 2017). Figures are typically presented in order to communicate something about the underlying data, but in an inherently static way. As such, reshaping this communication is not readily possible, because the data are not available. Examples of reuse if the data are available could be as simple as joining data across figures, standardizing axes across figures for easy comparison, changing color codings to be more colorblind friendly, or using the data to compute relative numbers instead of absolute numbers. Moreover, considering the current low rates of data sharing (Wicherts et al. 2006; Vanpaemel et al. 2015; Krawczyk and Reuben 2012) and rapid decrease of the odds of successfully requesting those data (Vines et al. 2014), reusing data effectively becomes impossible in the long run because data simply are not available any more. Hence, we find it important to be able to have alternative ways of extracting data solely from results presented in a scholarly report.

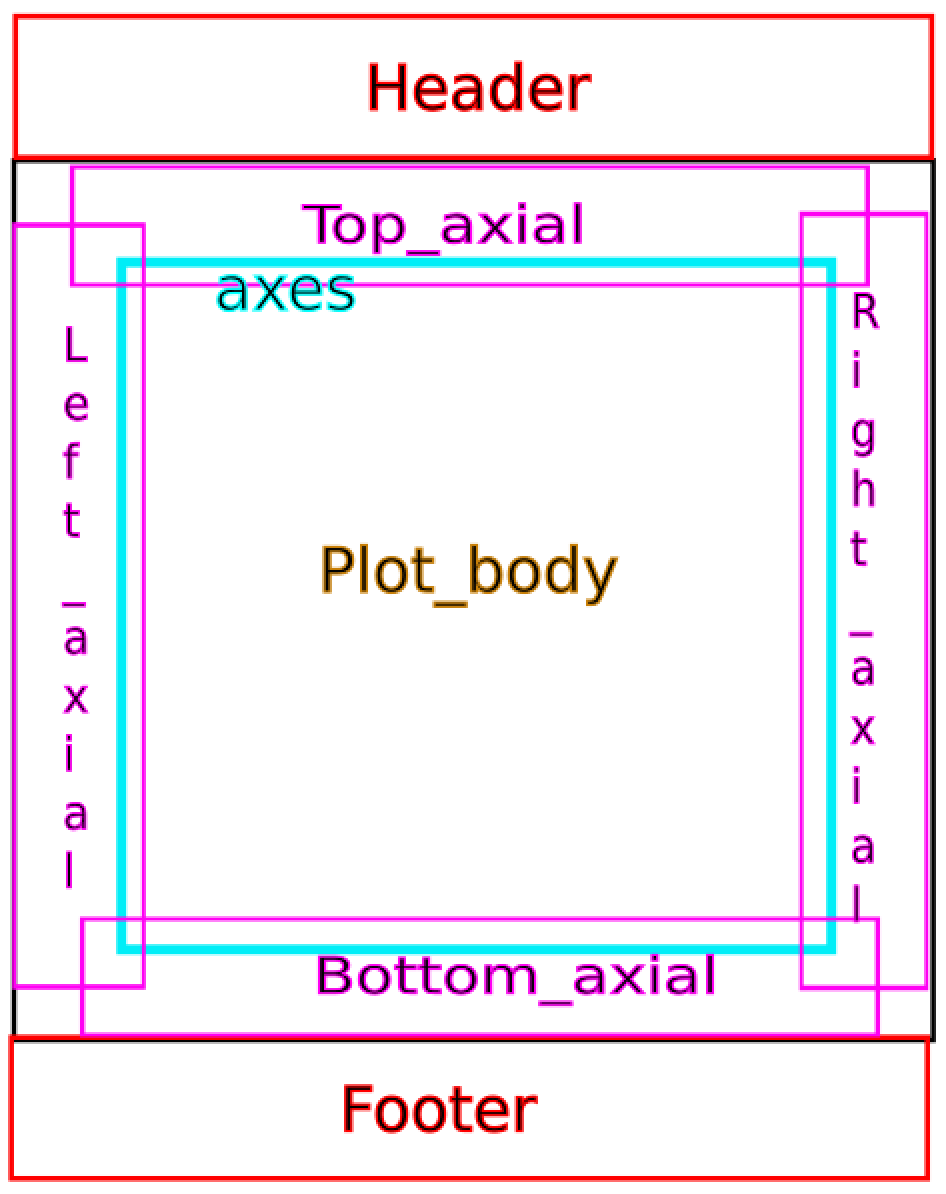
Some figures are stored in bitmap format whereas others are stored in vector format. In a bitmap format the image is stored by saving the color code for each pixel. This means that information about overlapping datapoints is lost, because a pixel in a bitmap does not differentiate between different layers. However, in a vector format, information is stored on the shape and its position on the canvas, which is unrestricted to a specific pixel size, and information can be saved. As such, these images can be enlarged without loss of image quality. Moreover, the position of those shapes can be retraced in order to reconstruct data points in a figure. This can even be done when data points overlap, because unlike in the pixel format, overlapping shapes are stored alongside each other in a vector image.

In the current report, we share the results of the alpha software norma ([github.com/contentmine/norma](https://github.com/contentmine/norma)) to extract raw data from vector based figures. More specifically, we report the method of data extraction, the effectiveness, and provide documentation to use the software pipeline. Finally, we review the potential of using vector based images to extract data from scholarly reports in light of the results.

# Method

## Extraction procedure

At the highest level, typical figure components are the body, header, footer, and axes. Figure 1 provides a visual depiction of these figure components. In order to extract data, recognition of some these components is mandatory, whereas recognition of others is optional. For example, the header and footer are irrelevant to data extraction, but are relevant to data comprehension; hence these are optional. Left- and bottom axes are mandatory, because these typically depict the scale of the presented plots. Right- and top axes are optional because they are rarely used as the main axes and mostly just to delimit the plotbox (as far as we know). Logically, the body of the plot, containing the depicted data, is mandatory for data extraction.



Visual representation of the typical components to a data based plot. This serves as the basis of the software to extract data from the plot body.

Based on the plot body, absolute locations of the individual data points are extracted. Not all vector images are created in a similar way, but in the simplest scenario with data points depicted as circles, the vector gives three parameters: the coordinate of the centre, the coordinate of the centre, and the radius . As such, for a simple circle the underlying vector code might look as follows:

<circle cx="103.71" cy="121.22" r="25.234" fill-opacity="0" stroke="#cf1d35" stroke-width=".26458"/>

This information can be readily extracted after isolating the vector figure from a PDF file. The current alpha software is primarily developed to operate on circles of similar size within one plot but can be extended for data depicted in other ways.

In order to make the absolute locations of the shapes represent the original data points as accurately as possible, they are mapped onto the identified x- and y-axis. Although absolute locations retain the relative relations between the individual data points, they are not representative of the original data. As such, the software extracts the locations of the axes and their respective scales. Subsequently, the location of the x-axis and y-axis, combined with the information about the scale, are combined to remap the absolute locations on the canvas into the original data points. The current alpha software assumes a linear scale, but logarithmic scales could be incorporated at a future stage.

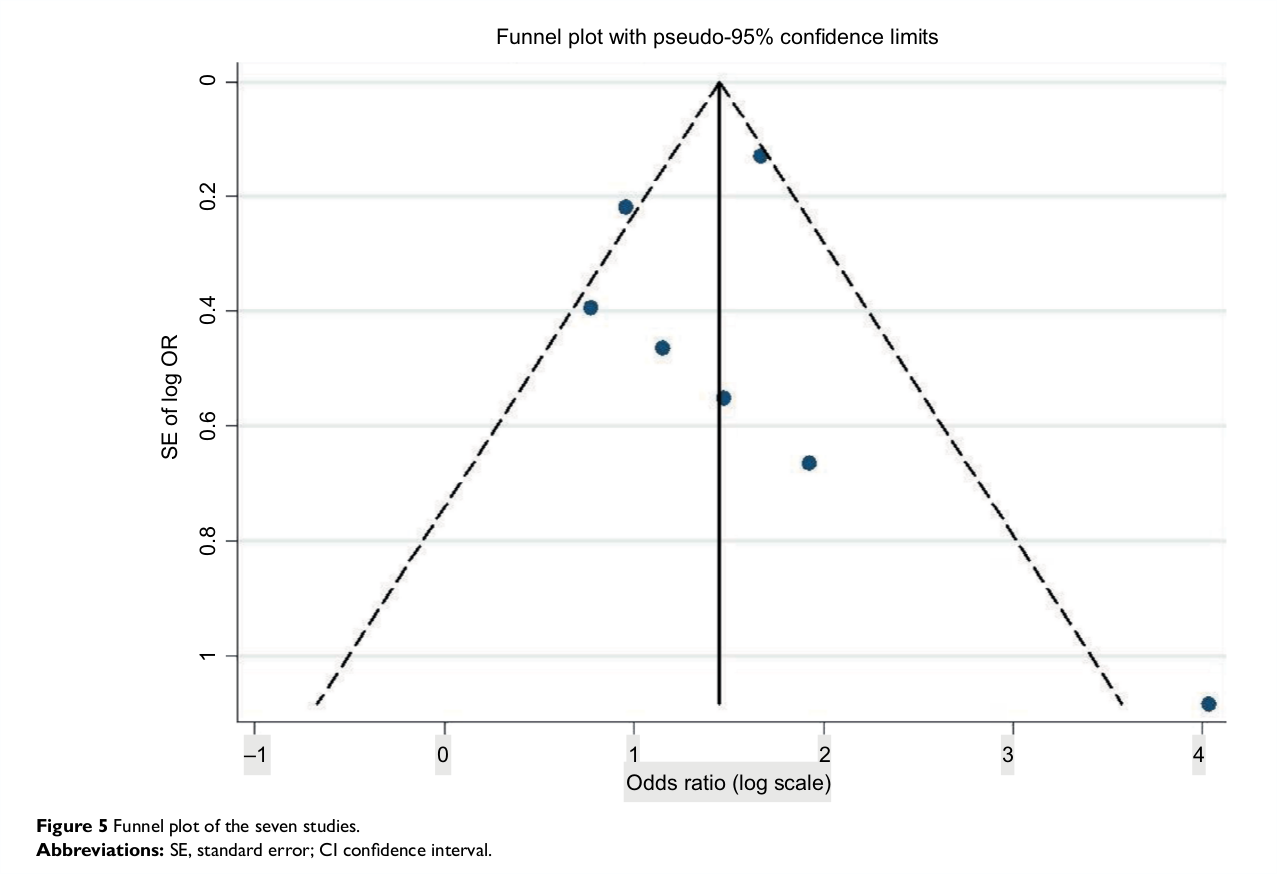
## Corpus

Using ScienceOpen, we searched for meta-analytic reports that mention "publication bias". For this project, we focused on funnel plot figures from meta-analyses. We restricted our search on ScienceOpen to Open Access reports, in order to legally redistribute those reports in the [Github project repository (https://github.com/chartgerink/2015ori-3)](https://github.com/chartgerink/2015ori-3), which facilitates reproducibility of our procedure. We searched the ScienceOpen database on March 30 2017; this search resulted in 422 reports (see also Figure 1), but the webpage presented only 368 reports.



Screenshot of the search criteria used to search ScienceOpen.

We manually searched through these 368 reports for vector based funnel plots. The first author (CHJH) checked each article for (1) whether a funnel plot was present; (2) if so, how many funnels were present, and (3) whether the funnel plots were vector based. In order to determine whether a funnel plot was vector based, a heuristic was used. This heuristic was to try and select the axes (either x- or y-axis) from the plot. If we could select the labels from the tickmarks, the plot was deemed to be vector based, otherwise it was dropped. We later found out this was a liberal heuristic, considering some publishers present vector axes but incorporate a bitmap plot body (see Figure 2 for an example).



Example of a funnel plot with selectable axes (Suphanchaimat et al. 2016), but including a bitmap body. This can be seen by the large difference in quality between the body and the axes, where the axes are crisp and the body is pixelated. The x-axis is selected.

## Documentation

The following documentation assumes that the Github repository for this project is cloned and that the working directory is the resulting folder. As such, in order to walk through these documentation steps, the following code is run from the shell command line

# Via SSH  
git clone git@github.com:chartgerink/2015ori-3  
# Via HTTPS (if you don't know what SSH is, use this)  
git clone https://github.com/chartgerink/2015ori-3  
# Change the working directory  
cd 2015ori-3

If git is not available from the commandline, a direct download of the project is available from Github (<https://github.com/chartgerink/2015ori-3/archive/master.zip>). All other dependencies to reproduce the results are included in the packaged command line tool norma and the user is only required to have Java installed on their system and available from the commandline.

In order to extract data from vector figures with the software norma, 5 steps are taken. First, the user needs to organize all original PDFs into one folder. Second, this folder needs to be converted to a cproject structure. The cproject structure normalizes the contents for each paper into a ctree, such that subsequent operations are trivial to standardize (and extensions can be applied relatively easily). For example, the root folder might contain ctree1.pdf, but after transforming the root folder into a cproject it contains a folder ctree1/ with fulltext.pdf. By running the command

java -jar bin/norma-0.5.0-SNAPSHOT-jar-with-dependencies.jar --project corpus-raw --fileFilter '.\*/(.\*).pdf' --makeProject '(\1)/fulltext.pdf'

the folder corpus-raw (--project corpus-raw) is restructured into a cproject structure, containing a folder for each PDF file (--fileFilter '.\*/(.\*).pdf' --makeProject '(\1)/fulltext.pdf'). This results in the following folder structure:

cproject/  
├── ctree1  
│   └── fulltext.pdf  
├── ctree2  
│   └── fulltext.pdf  
...  
└─── ctreeN  
    └── fulltext.pdf

After converting the folder into a cproject, the norma software is applied to convert the PDF files into separate SVGs per page. In order to convert each page of the PDF into a separate SVG file, we used the following command

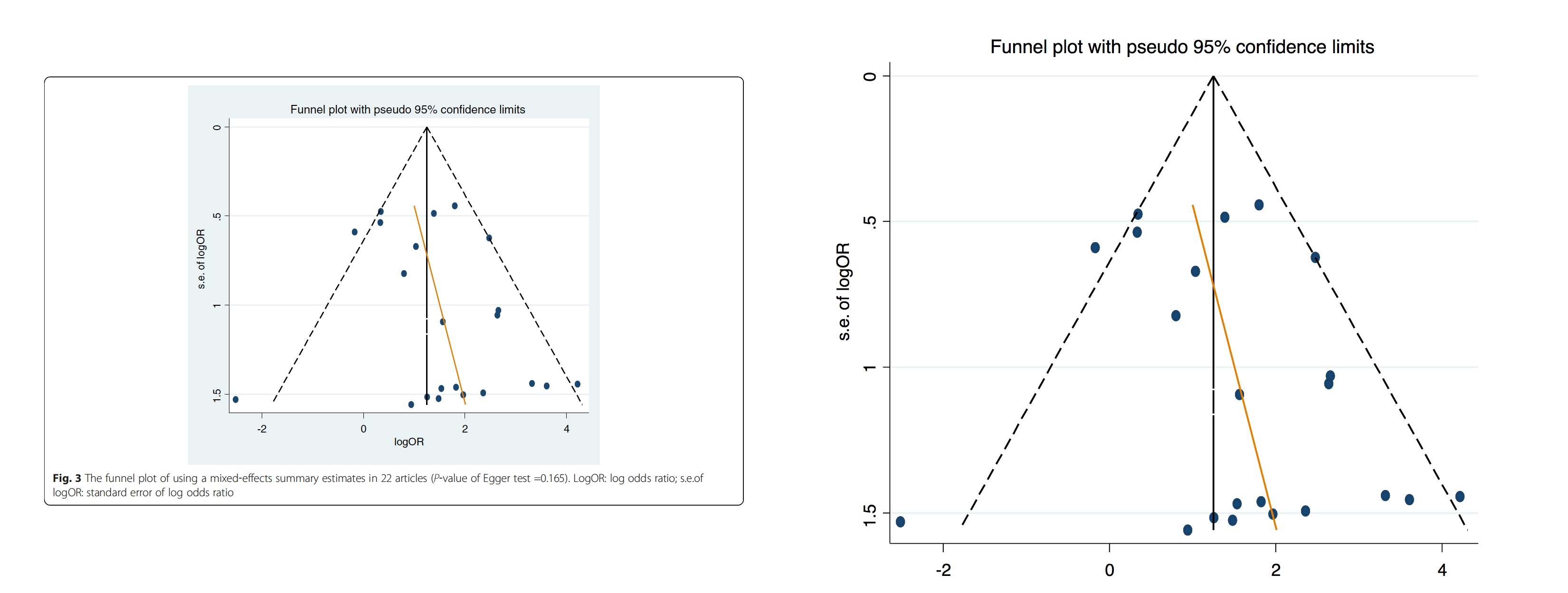
java -jar bin/norma-0.5.0-SNAPSHOT-jar-with-dependencies.jar --project corpus-raw -i fulltext.pdf --outputDir corpus-raw --transform pdf2svg

resulting in a svg/ folder for each ctree in the structure presented above. That is, each ctree now contains a folder with one vector file for each page in the fulltext PDF.

The following step, extracting the plots from the page and saving these, currently needs to be done manually. We recommend using the FOSS software [Inkscape](https://inkscape.org/) to do this. For each article, open the pages containing funnel plots, select the area of the plot, and press the keyboard shortcut SHIFT+1 (i.e., !) to inverse the selection; then press the delete key to retain only the plot. Subsequently save the file as Plain SVG (not Inkscape SVG) and structure the folders as follows:

cproject/  
└─── ctree1  
    └── fulltext.pdf  
    └── figures/  
    └── figure1/  
 └── figure.svg  
    └── figure2/  
 └── figure.svg

where figure1 contains the first funnel plot (not the figure number in the paper), figure2/ contains the second funnel plot, etc. If a figure is contained in a box, it is important to retain only the figure and exclude the box (see Figure 2 for an example). In the Github repository, we provide a project folder that already contains all the clipped images for the corpus under investigation in this paper.



The original depiction of a funnel plot (left, from Bae and Kim 2016) and the manually extracted part that is subsequently ingested into the norma software (right).

Finally, each figure is converted to a data file with norma. The following command produces an annoted SVG file showing the identified areas from Figure 1 and a CSV file containing the data based on the manually clipped figures available in the corpus-clipped/ folder.

java -jar bin/norma-0.5.0-SNAPSHOT-jar-with-dependencies.jar --project corpus-clipped --fileFilter "^.\*figures/figure(\\d+)/figure(\_\\d+)?\\.svg" --outputDir corpus-clipped --transform scatter2csv

We provide the fully extracted data in the folder corpus-extracted/ of the Github repository (<https://github.com/chartgerink/2015ori-3/archive/master.zip>).

# Results

By searching ScienceOpen, we identified 15 meta-analytic reports containing vector based funnel plots. Upon manual inspection of the 368 initially found meta-analytic reports, 136 (37%) contained funnel plots. Of those 136 meta-analytic reports with funnel plots, we identified 32 reports (24%) with vector based images, assuming the heuristic described in the methods section (i.e., selectable tick marks). Finally, of those 32 reports with selectable tick marks, 15 reports contained a vector based plot body (47%).

These 15 reports contained 27 vector funnel plots; we extracted data for 24 funnel plots (89%) using the software. Table 1 depicts the DOIs and the figure numbers for which we extracted or failed to extract data. For the 3 funnel plots without extracted data, the notes indicate potential reasons as to why we were unable to extract the data. This provides indications as to how the software can be developed further.

All meta-analytic reports with vector based funnel plots and the results of automated data extraction. The figure number depicts the funnel plot order for extraction, the paper figure number depicts the original figure number in the paper. 'Data extracted' indicates whether any datafile was generated by the software. '# extracted data points' indicates the number of rows in the datafile; '# manual data points' indicates the visually discernable data points in the plot. 'X-axis correct' and 'Y-axis correct' depicts whether the extracted data corresponded to the data points of the funnel plot upon manual inspection.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DOI | Fig. # | Paper fig. # | Data extracted | # extracted data points | # manual data points | X-axis correct | Y-axis correct |
| 10.1186/s12885-016-2685-3 | 1 | 4 | yes | 24 | 24 | yes | no |
| 10.1186/s12889-016-3083-0 | 1 | 3 | yes | 5 | 5 | no | no |
| 10.1186/s12891-016-1231-4 | 1 | 4 | yes | 6 | 6 | yes | no |
| 10.1186/s13027-016-0058-9 | 1 | 3 | yes | 24 | 22 | yes | yes |
| 10.1186/s13054-016-1298-1 | 1 | 4 | no | NA | NA | NA | NA |
| 10.1186/s40064-016-3064-x | 1 | 6a | no | NA | NA | NA | NA |
| 10.1186/s40064-016-3064-x | 2 | 6b | yes | 21 | 8 | no | no |
| 10.1186/s40064-016-3064-x | 3 | 6c | yes | 19 | 6 | no | no |
| 10.1515/med-2016-0052 | 1 | 2 | yes | 23 | 23 | yes | yes |
| 10.1515/med-2016-0052 | 2 | 4 | yes | 23 | 23 | yes | yes |
| 10.1515/med-2016-0052 | 3 | 6 | yes | 23 | 23 | yes | yes |
| 10.1515/med-2016-0099 | 1 | 7a | yes | 7 | 7 | yes | yes |
| 10.1515/med-2016-0099 | 2 | 7b | yes | 11 | 11 | yes | yes |
| 10.1515/med-2016-0099 | 3 | 7c | yes | 10 | 10 | yes | yes |
| 10.1515/med-2016-0099 | 4 | 7d | yes | 7 | 7 | yes | yes |
| 10.1590/S1518-8787.2016050006236 | 1 | 3 | no | NA | NA | NA | NA |
| 10.21053/ceo.2016.9.1.1 | 1 | 3 | yes | 18 | 18 | yes | no |
| 10.21053/ceo.2016.9.1.1 | 2 | 3 | yes | 13 | 13 | yes | no |
| 10.21053/ceo.2016.9.1.1 | 3 | 3 | yes | 14 | 14 | yes | yes |
| 10.21053/ceo.2016.9.1.1 | 4 | 3 | yes | 9 | 9 | yes | yes |
| 10.2147/BCTT.S94617 | 1 | 7 | yes | 7 | 7 | no | yes |
| 10.3349/ymj.2016.57.5.1260 | 1 | 7 | yes | 24 | 24 | yes | yes |
| 10.3349/ymj.2016.57.5.1260 | 2 | 8 | yes | 12 | 12 | yes | yes |
| 10.3390/ijerph13050458 | 1 | 24 | yes | 15 | 15 | yes | no |
| 10.5114/aoms.2016.61916 | 1 | 4 | yes | 5 | 5 | yes | no |
| 10.5114/aoms.2016.61916 | 2 | 4 | yes | 4 | 4 | yes | no |
| 10.5812/ircmj.40061 | 1 | 11 | yes | 30 | 30 | yes | yes |

Of the 24 funnel plots with extracted data, we correctly extracted data for 15 funnel plots (56%). That is, the data points were correctly mapped onto the x- and y-axis and the number of extracted data points corresponded to the number of visually discernable data points. For the remaining 9 funnel plots, there was 1 with correctly mapped x- and y-axes, but with an incorrect number of extracted data points. For the remaining 14 funnel plots, the software did not correctly map the axes or did not extract the correct number of data points.

# Discussion

As the results indicate, vector figures are a fruitful and feasible resource for data extraction. Based on initial alpha software of norma to extract data from vector figures, we correctly extracted data from 56% of funnel plots for which data was extracted. This is a very strict assessment, considering our manual investigation depicted that four extracted datasets only required a simple reversal of one axis to be 100% accurate (i.e., 1.23 should be -1.23, etc.), adding a constant to all data points on an axis to adjust an incorrect mapping, or by rescaling an axis to fit to the logarithmic scale of one axis. If these manual corrections for four funnel plots are made, 70% of all funnel plots for which data were correctly extracted.

Considering the alpha software to extract data from vectors was developed in the timespan of approximately one month, we are hopeful that future development can refine the data extraction and eliminate some flaws that exist in the alpha software, including those such as reversed axes, logarithmic axes, etc. The current software was developed specifically on funnel plots, but its use can be extended to include other types of plots, such as histograms, etc. Moreover, third-dimensions such as variable point size provide a fruitful avenue, considering the SVG also contains information on this (see [Extraction procedure](#extraction-procedure)).

The main bottleneck for data extraction from vector figures is the publication of vector figures. The results indicate that the availability of vector figures is relatively sparse; only 15 out of 136 papers with funnel plots contained vector figures in our sample. From our own anecdotal publishing experience, vector based figures are often converted into bitmaps in the editing stage, resulting in loss of information. For publishers themselves, it is also fruitful to use vector based figures where possible, considering figure quality is no longer an issue as a result. As such, we encourage both authors and publishers to produce figures in vector format (e.g., PDF, SVG, EPS) instead of bitmap format (e.g., JPEG, PNG, GIF) when it regards figures presenting results. Not only will it benefit the quality of the publications, it will also present a new way of data preservation.

# Author notes

All presented materials in this report are version-controlled and freely available a CC 0 Public Domain Dedication on Github (<https://github.com/chartgerink/2015ori-3>), and are preserved at Zenodo (**???**). This research was funded by the Office of Research Integrity under grant ORIIR160019; software developed by ContentMine.

# References

Bae, Jong-Myon, and Eun Hee Kim. 2016. “Human Papillomavirus Infection and Risk of Breast Cancer: A Meta-Analysis of Case-Control Studies.” *Infectious Agents and Cancer* 11 (1). Springer Nature. doi:[10.1186/s13027-016-0058-9](https://doi.org/10.1186/s13027-016-0058-9).

Hartgerink, C. H. J. 2017. “Developing the Potential of Data Extraction in Scholarly Articles for Research Verification.” *D-Lib Magazine*. [http://www.dlib.org/dlib/may17/05inbrief.html#HARTGERINK](http://www.dlib.org/dlib/may17/05inbrief.html" \l "HARTGERINK).

Krawczyk, Michal, and Ernesto Reuben. 2012. “(Un)Available Upon Request: Field Experiment on Researchers’ Willingness to Share Supplementary Materials.” *Accountability in Research* 19 (3): 175–86. doi:[10.1080/08989621.2012.678688](https://doi.org/10.1080/08989621.2012.678688).

Suphanchaimat, Rapeepong, Nisachol Cetthakrikul, Alexander Dalliston, and Weerasak Putthasri. 2016. “The Impact of Rural-Exposure Strategies on the Intention of Dental Students and Dental Graduates to Practice in Rural Areas: A Systematic Review and Meta-Analysis.” *Advances in Medical Education and Practice* Volume 7 (October). Dove Medical Press Ltd.: 623–33. doi:[10.2147/amep.s116699](https://doi.org/10.2147/amep.s116699).

Vanpaemel, Wolf, Maarten Vermorgen, Leen Deriemaecker, and Gert Storms. 2015. “Are We Wasting a Good Crisis? The Availability of Psychological Research Data After the Storm.” *Collabra* 1 (1). University of California Press. doi:[10.1525/collabra.13](https://doi.org/10.1525/collabra.13).

Vines, Timothy H., Arianne Y.K. Albert, Rose L. Andrew, Florence Débarre, Dan G. Bock, Michelle T. Franklin, Kimberly J. Gilbert, Jean-Sébastien Moore, Sébastien Renaut, and Diana J. Rennison. 2014. “The Availability of Research Data Declines Rapidly with Article Age.” *Current Biology* 24 (1). Elsevier BV: 94–97. doi:[10.1016/j.cub.2013.11.014](https://doi.org/10.1016/j.cub.2013.11.014).

Wicherts, Jelte M., Denny Borsboom, Judith Kats, and Dylan Molenaar. 2006. “The Poor Availability of Psychological Research Data for Reanalysis.” *American Psychologist* 61 (7). American Psychological Association (APA): 726–28. doi:[10.1037/0003-066x.61.7.726](https://doi.org/10.1037/0003-066x.61.7.726).