

1 **Reproducible, flexible and high throughput data extraction from**
2 **primary literature: The metaDigitise R package**

3 Joel L. Pick^{1,*}, Shinichi Nakagawa¹, Daniel W.A. Noble¹

4 ¹ Ecology and Evolution Research Centre, School of Biological, Earth and
5 Environmental Sciences, University of New South Wales, Kensington, NSW 2052,
6 Sydney, AUSTRALIA

7 *Corresponding Author: joel.l.pick@gmail.com

8 **Abstract**

9 Research synthesis, especially in the form of meta-analysis, requires data
10 extraction from primary studies. Meta-analysis synthesizes effect sizes, often
11 calculated from summary statistics of studies. However, exact values of such
12 statistics are commonly hidden in figures. The R package **metaDigitise** extracts
13 descriptive statistics such as means, standard deviations and, if applicable,
14 correlations from the four types of plots: 1) mean and error plots (e.g. bar
15 graphs with standard errors), 2) box plots, 3) scatter plots and 4) histograms.
16 The package interactively guides the user through data extraction process.
17 Notably, it enables a large-scale extraction using image files, letting the user stop
18 processing, edit and add to the resulting data fame at any point. Further, it
19 facilitates reproducible data extraction from plots with little inter-observer bias,
20 thus, allowing a group of people to participate the extraction of data
21 collaboratively.

22 Keywords: meta-analysis, comparative analysis, data extraction, R,
23 reproducibility, figures, images, summary statistics

24 **Introduction**

25 In many different contexts, researchers need to make use of data presented in
26 primary literature. Most notably, this includes meta-analysis, which is becoming
27 increasingly common in many research fields. Meta-analysis uses effect size
28 estimates and their sampling variance, taken from many studies, to understand
29 whether particular effects are common across studies and to explain variation
30 among these effects (Glass, 1976; Borenstein et al., 2009; Koricheva, Gurevitch &
31 Mengersen, 2013; Nakagawa et al., 2017). Meta-analysis therefore relies foremost
32 on data extracted from primary literature, and more specifically, descriptive
33 statistics (e.g., means, standard deviations, correlation coefficients) that have
34 been reported in the text or tables of research papers. Descriptive statistics are
35 also, however, frequently presented in figures and so need to be manually
36 extracted using digitising programs. While inferential statistics (e.g., t - and
37 F -statistics) are often presented along side descriptive statistics and can be used
38 to derive effect sizes, descriptive statistics are much more appropriate to use
39 because sources of non-independence in experimental designs can be dealt with
40 more easily (Noble et al., 2017). Although there are several existing tools to
41 perform tasks like this (e.g. DataThief (Tummers, 2006), GraphClick
42 (Arizona-Software, 2008), WebPlotDigitizer (Rohatgi, 2017)), these tools are not
43 designed specifically for meta-analysis for three main reasons.

44 First, they typically only provide the user with calibrated x,y coordinates from
45 imported figures, and do not differentiate between common plot types that are
46 used to present data. This means that a large amount of downstream data
47 manipulation is subsequently required, that is different across plots types. For
48 example, data are frequently presented in mean and error plots (Figure 1A), for
49 which the user wants a mean and error estimate for each group presented in the
50 figure. With existing programs, x,y coordinates of means and errors are returned,
51 to which the user must manually discern between mean and error coordinates
52 and assign points to groups. The error then needs to be calculated as the
53 deviation from the mean, and then transformed to a standard deviation,
54 depending on the type of error presented.

55 Second, digitising programs do not easily allow the integration of metadata at
56 the time of data extraction, such as experimental group or variable names, and
57 sample sizes. This makes the downstream calculations more laborious, as the
58 information has to be added later, in most cases using different software.

59 Finally, existing programs do not import a set of images and allow the user to
60 systematically work through them. Instead they require the user to manually
61 import images one by one, and export data into individual files, that need to be
62 imported and edited using different software. In essence, existing software does
63 not provide an optimized research pipeline to facilitate data extraction, editing
64 and reproducibility.

65 These are major issues because extracting from figures can be an incredibly
66 time-consuming process. Furthermore, although meta-analysis is an important
67 tool in consolidating the data from multiple studies, many of the processes

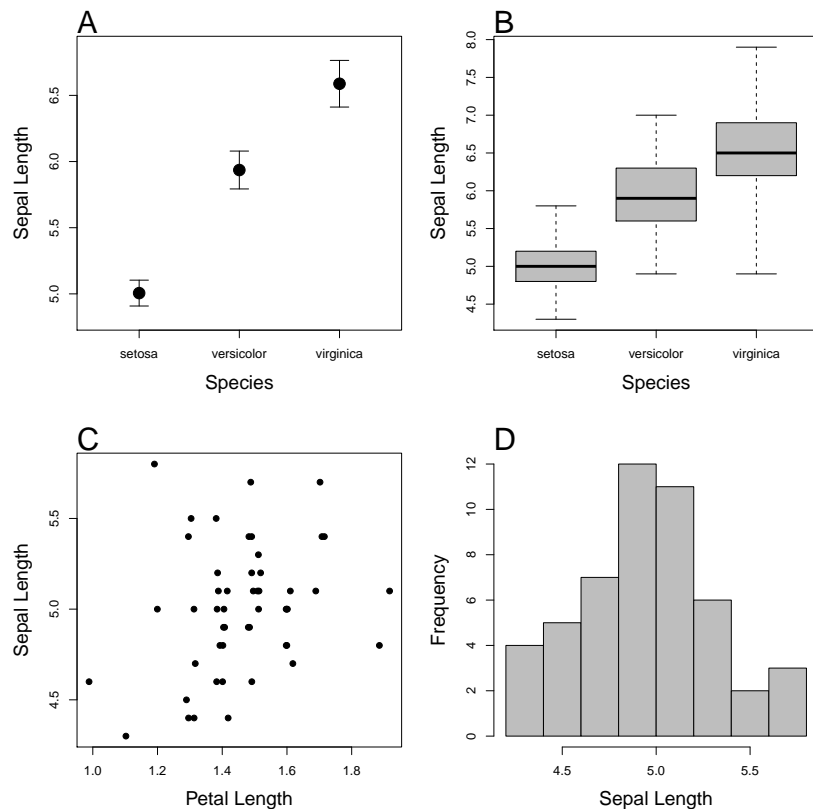


Figure 1: Four plot types that **metaDigitise** is designed to extract data from: A) mean and error plot, B) box plot, C) scatter plot and D) histogram. Data is taken from the iris dataset in R. A and B are plotted with the whole dataset, C and D are just the data for the species setosa.

68 involved in data extraction are opaque and difficult to reproduce, making
69 extending studies problematic. Having a tool that facilitates reproducibility in
70 meta-analyses will increase transparency and go a long way to resolving the
71 reproducibility crises we are seeing in many fields (Peng, Dominici & Zeger, 2006;
72 Peng, 2011; Sandve et al., 2013; Parker et al., 2016; Ihle et al., 2017).

73 Here, we present an interactive R package, **metaDigitise**, which is designed for
74 large scale data extraction from figures, specifically catering to the the needs of
75 meta-analysts. To this end, we provide tools specific to data extraction from

76 common plot types (mean and error plots, box plots, scatter plots and
77 histograms, see Figure 1). **metaDigitise** operates within the R environment
78 making data extraction, analysis and export more streamlined. It also provides
79 users with options to conduct the necessary calculations on processed data
80 immediately after extraction so that comparable summary statistics can be
81 obtained quickly. **metaDigitise** condenses summary data extracted from multiple
82 figures into a single data frame which can be easily exported. Processed data
83 can also be easily extracted and analysed in any way the user desires in
84 downstream analysis within R. Conveniently, when needing to process many
85 figures at different times **metaDigitise** will only import figures not already
86 completed within a directory. This makes it easy to add new figures at any time.
87 **metaDigitise** has also been built for reproducibility in mind. It has functions
88 that allow users to redraw their digitisations on figures, make corrections and
89 access the raw calibration data which is written automatically for each figure
90 that is digitised into a special folder within the directory. This makes sharing
91 figure digitisation and reproducing the work of others simple and easy, and
92 allows meta-analysts to update meta-analyses more easily.

93 **Directory Structure, Image Processing and** 94 **Reproducibility**

95 The **metaDigitise** package is designed to be flexible, yet simple to use. There is
96 one main function in the package, `metaDigitise()`, which interactively takes the
97 user through the process of extracting data from figures. `metaDigitise()` was

98 created with the idea that the user would likely have multiple images to extract
99 from. It therefore operates in the same way whether the user has one or multiple
100 images. `metaDigitise()` is designed to work on a directory containing images of
101 figures copied from primary literature, in .png, .jpg, .tiff, .pdf format. This
102 directory is specified to `metaDigitise()` through the `dir` argument. The user is
103 free to set their own broad directory structure (e.g. one directory for all images
104 or one directory for each paper extracted from). We would recommend having all
105 files for one project in a single directory with an informative and unambiguous
106 naming scheme for images to make it easy to identify the paper and figure the
107 data come from. This cuts out the need to change directories constantly. For
108 example the directory structure could look like:

```
* Main project directory
  + FiguresToExtract/
    + Paper1_Figure1_trait1.png
    + Paper1_Figure2_trait2.png
    + Paper1_Figure3_trait3.png
    + Paper2_Figure1_trait1.png
    + Paper2_Figure2_trait2.png
    + Paper2_Figure3_trait3.png
```

109 It is important for the user to think about their directory structure early on in
110 this process (also more generally in the context of their entire project), especially
111 if they plan to share the extractions with collaborators or when publishing the
112 project.

113 When `metaDigitise()` is run, it recognizes all the images in a directory and

114 automatically imports them one by one, allowing the user to click and enter
115 relevant information about a figure as they go. This expedites digitising figures
116 by preventing users from having to constantly change directories and / or open
117 new images. The data from a completed image is automatically saved as a
118 `metaDigitise` object in an `.RDS` file to a `caldat` directory that is created within
119 the parent directory when first executing the `metaDigitise()` function. These
120 files enable re-plotting and editing of images at a later point (see below).

121 A particularly powerful and flexible aspect of `metaDigitise()` is its ability to
122 identify images that have been previously digitised and only import images that
123 have not been digitised in subsequent calls of the function. This means that all
124 figures do not need to be extracted at one time and that new figures can be
125 added as the project develops. After each image is extracted, the user is asked
126 whether they wish to continue or quit the extraction process. Upon rerunning
127 `metaDigitise()`, previously digitised figures are simply ignored during
128 processing, but their data is re-integrated within the final output after new files
129 are completed automatically.

130 After completing all images, or upon quitting, the processed data (in a form
131 specified by the user) is then returned. From all plot types, `metaDigitise()`
132 summarises the data from a figure as a mean, standard deviation and sample
133 size, for each identified group within the plot (should multiple groups exist).
134 These are the descriptive statistics needed to create many of the relevant effect
135 sizes and sampling error for a meta-analysis. In the case of scatter plots,
136 `metaDigitise()` also returns the correlation coefficient between the points
137 within each identified group.

138 **Diverse Plot Types**

139 **metaDigitise** recognises four main types of plot; Mean and error plots, box plots,
140 scatter plots and histograms, shown in Figure 1. Each of these can be processed
141 together and integrated into a single output. Alternatively, users can keep like
142 figures together and process them separately.

143 In order to correctly extract data from figures `metaDigitise()` always requires
144 the user to calibrate the axes in the figure. To do this, the user is required to click
145 on two known points on the axis in question, and then enter the value of those
146 points in the figure. Using this information, `metaDigitise()` then calculates the
147 value of any clicked points in terms of the figure axes. In the case of mean and
148 error plots and box plots, it calibrates only the y-axis (assuming the x-axis is
149 redundant). For scatter plots and histograms both axes are calibrated.

150 **Mean and error plots**

151 `metaDigitise()` prompts the user to enter group names and allows the user to
152 enter sample sizes (n), which are used in downstream processing. The user is
153 then prompted to click on an error bar followed by the mean. Error bars above
154 or below the mean can be clicked - sometimes one is clearer than the other.
155 `metaDigitise()` assumes that the error bars are symmetrical. Where the user
156 has clicked the error is displayed in a different colour to the mean (Figure 2A).
157 The user can subsequently add more groups, edit groups or remove groups.
158 Finally the user is asked what type of error was used in the figure: standard
159 deviation (SD, σ), standard error (SE) or 95% confidence intervals (CI95).

160 Standard deviation is calculated from standard error as

$$\sigma = SE\sqrt{n} \quad (1)$$

161 and from 95% confidence intervals as

$$\sigma = \frac{CI}{1.96}\sqrt{n} \quad (2)$$

162 If the user does not enter a sample size at the time of data extraction (if, for
163 example, the information is not readily available) the SD is not calculated. This
164 can be entered at a later time, however (see below). A function, `error_to_sd()`,
165 that converts the different error types to SD is also available in the package.

166 **Box plots**

167 As with mean and error plots, `metaDigitise()` prompts the user to enter group
168 names and allows the user to enter sample sizes (n), which are used in
169 downstream processing. The user is then prompted to click on the maximum (b),
170 upper quartile (q_3), median (m), lower quartile (q_1) and minimum (a).
171 `metaDigitise()` will check that the maximum is greater than the minimum, and
172 return a warning if that is not the case. The user can subsequently add, edit or
173 remove groups. From the extracted data, the mean (μ) and SD are calculated
174 as

$$\mu = \frac{(n+3)(a+b) + 2(n-1)(q_1+m+q_3)}{8n} \quad (3)$$

175 following Bland (2015) and

$$\sigma = \frac{b - a}{4\Phi^{-1}\left(\frac{n-0.375}{n+0.25}\right)} + \frac{q_3 - q_1}{4\Phi^{-1}\left(\frac{0.75n-0.125}{n+0.25}\right)} \quad (4)$$

176 where $\Phi^{-1}(z)$ is the upper z th percentile of the standard normal distribution,
177 following Wan et al. (2014). As with mean and error plots, if the user does not
178 enter a sample size at the time of data extraction the SD is not calculated. Two
179 functions, `rqm_to_mean()` and `rqm_to_sd()`, that convert box plot data to mean
180 and SD respectively are also available in the package.

181 Scatter plots

182 `metaDigitise()` prompts the user to enter groups names and then to click on
183 points. Points added by mistake can be deleted. The user can subsequently add
184 groups, edit groups (add or remove points) or delete groups. Different groups are
185 plotted in different colours and shapes, with a legend at the bottom of the figure
186 (Figure 2C). Mean, SD and sample size are calculated from the clicked points, for
187 each group. Where the sample size from the clicked points does not match a
188 known sample size (e.g. if there are overlaid points), the user can enter an
189 alternate sample size.

190 Histograms

191 `metaDigitise()` prompts the user to click on the top corners of each bar. Bars
192 can subsequently be deleted. For each bar a midpoint (m ; mean x coordinates)

193 and a frequency (f_i ; mean y coordinates, rounded to the nearest integer) is
194 calculated. The sample size, mean and SD are calculated as:

$$n = \sum_{i=1}^n f_i \quad (5)$$

$$\mu = \frac{\sum_{i=1}^n m_i f_i}{n} \quad (6)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (m_i f_i - \mu f_i)^2}{n - 1}} \quad (7)$$

195 As with the scatterplots, if the sample size from the extracted data does not
196 match a known sample size, the user can enter an alternate sample size.

197 **Extracting Data From Plots**

198 We will now demonstrate how `metaDigitise()` works using figures generated
199 from the well known iris data set. Users can install the **metaDigitise** package
200 from GitHub as follows:

```
R> install.packages("devtools")  
R> devtools::install_github("daniel1noble/metaDigitise")  
R> library(metaDigitise)
```

201 Assume that the user would like to extract descriptive statistics from studies
202 measuring sepal length or width in iris species for a fictitious project. There are

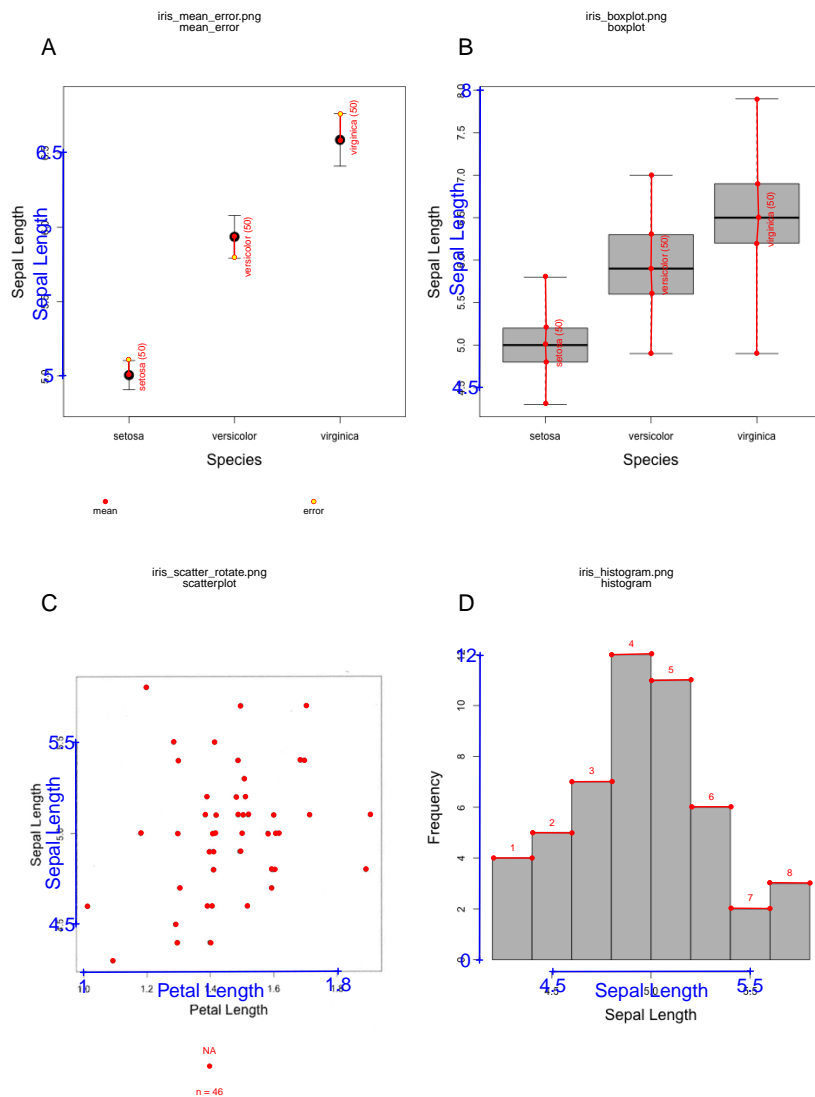


Figure 2: Demonstration of data extraction from different plot types

203 a few studies that only present these data in figures. As the user reads papers
204 found from a systematic search, they add figures with relevant data to a
205 "FiguresToExtract" folder as follows

```
*FiguresToExtract/  
+ 001_Anderson_1935_Fig1.png
```

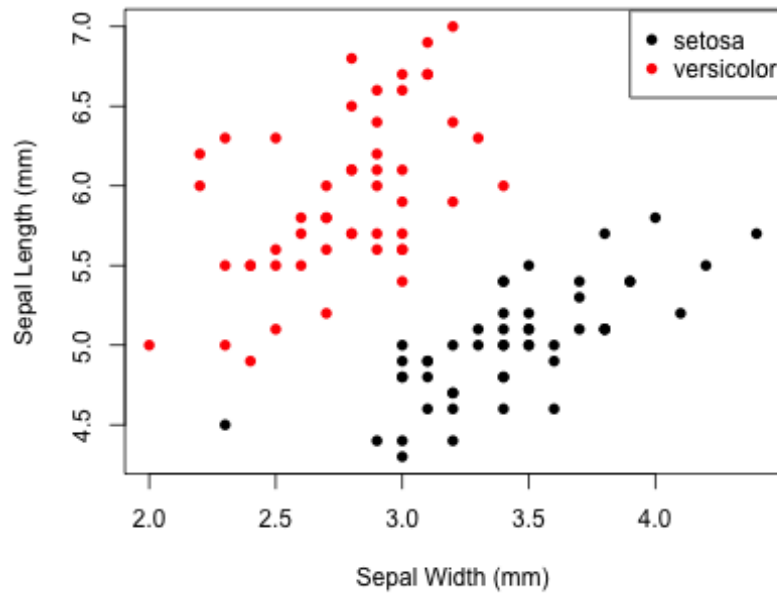


Figure 3: Example scatterplot (001_Anderson_1935_Fig1.png) of sepal length and width for two species of iris (setosa and versicolor)

206 Here, the naming of the files placed in the folder will contain the paper number,
207 first author and the figure number to keep data uniquely associated with figures.
208 At first there is one figure in the folder, shown in Figure 3. Running
209 `metaDigitise()` brings up a series of prompts for the user using a main menu
210 that provides access to a number of its features ("..." here represents the user's
211 path to the project directory):

```
R> digitised_data <- metaDigitise("../FiguresToExtract", summary = TRUE)
```

```
Do you want to...
```

```
1: Process new images
```

```
2: Import existing data
```

3: Edit existing data

Selection:

212 The user simply enters in the numeric value that corresponds to what they would
213 like to do. In this case they want to "Process new images". The user is then
214 asked whether there are different types of plot(s) in the folder. This question is
215 most relevant when there are lots of different figures in the folder because it will
216 then ask the user for the type of figure as they are cycled through.

Are all plot types Different or the Same? (d/s)

217 `metaDigitise()` then asks the user whether the figure needs to be rotated or
218 flipped. This can be needed when box plots and mean and error plots are not
219 orientated correctly. In some cases, older papers can give slightly off angled
220 images which can be corrected by rotating. So, in this prompt the user has three
221 options: `f` for "Flip", `r` for "rotate" or `c` for "continue".

`mean_error` and `boxplots` should be vertically orientated

```
-  
|  
I.E. o   NOT  |-o-|  
|  
-
```

If they are not then chose flip to correct this.

If figures are wonky, chose rotate.

Otherwise chose continue

Flip, rotate or continue (f/r/c)

R> c

222 After this, `metaDigitise()` will ask the user to specify the plot type. Depending
223 on the figure, the user can specify that it is a figure containing the mean and
224 error (**m**), a box plot (**b**), a scatter plot (**s**) or a histogram (**h**). If the user has
225 specified **d** instead of **s** in response to the question about whether the plot types
226 are the same or different, this question will pop up for each plot, but will only be
227 asked once if plots are all the same.

Please specify the `plot_type` as either:

m: Mean and error

b: Box plot

s: Scatter plot

h: Histogram

R> s

228 After selecting the figure type a new set of prompts will come up that will ask
229 the user first what the y and x-axis variables are. This is useful as users can keep
230 track of the different variables across figures and papers. Here, the user can just
231 add this information in to the R console. Once complete, details on how to
232 calibrate the x and y-axis appear, so that the relevant statistics / data can be

233 correctly calculated. When working with a plot of mean and standard errors, the
234 x-axis is rather useless in terms of calibration so `metaDigitise()` just asks the
235 user to calibrate the y-axis.

What is the y variable?

```
R> Sepal Length (mm)
```

What is the x variable?

```
R> Sepal Width (mm)
```

On the Figure, click IN ORDER:

y1, y2 , x1, x2

Step 1 ----> Click on known value on y axis - y1

```
|
|
|
|
y1
|-----
....
```

Step 3 ----> Click on known value on x axis - x1

```
|
|
```



```
|  
|  
|  
|_____x1_____
```

....

236 The user can just follow the instructions on screen step-by-step (instructions
237 above have been truncated by ‘...’ to simplify), and in the order specified. Before
238 moving on, the user is forced to check whether or not the calibration has been set
239 up correctly. If n is chosen because something needs to be fixed then the user can
240 re-calibrate.

What is the value of y1 ?

R> 4.5

What is the value of y2 ?

R> 7

What is the value of x1 ?

R> 2

What is the value of x2 ?

R> 4

Re-calibrate? (y/n)

R> n

241 Often, plots might contain multiple groups that the meta-analyst wants to
242 extract from. `metaDigitise()` handles this nicely by prompting the user to enter
243 the group first, followed by digitisation of this groups data. After digitising the
244 first group, and having exited, `metaDigitise()` will ask the user whether they
245 would like to add another group. Users can continually add groups (a), delete
246 groups (d), edit groups (e) or finish a plot and continue to the next one (f - if
247 another plot exists). The number of groups are not really limited and users can
248 just keep adding in groups to accommodate the different numbers that may be
249 presented across figures (although it can get complicated with too many).

If there are multiple groups, enter unique group identifiers (otherwise press enter)

Group identifier:

```
R> setosa
```

Click on points you want to add.

If you want to remove a point, or are finished with a group,
exit by clicking on red box in bottom left corner, then follow prompts

250 To finish selecting points, the user can exit by clicking on the red button that
251 appears when extracting points. The user is then asked if they want to add or
252 delete points from that group.

Add or Delete points to this group, or Continue? (a/d/c)

```
R> c
```

253 Once we are done digitising all the groups our plot will look something like
254 Figure 4.

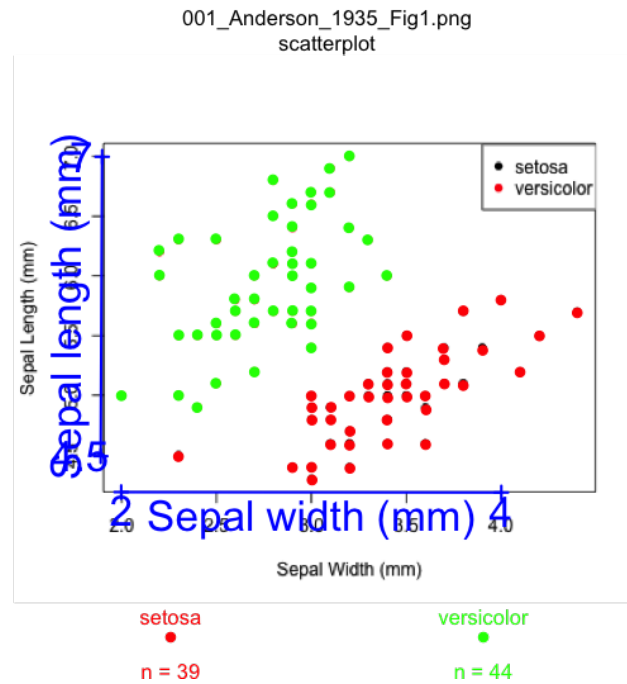


Figure 4: Digitisation of sepal length and width for two species of iris (setosa and versicolor). Names of the variables and calibration (in blue) are plotted alongside the digitised points (green = versicolor; red = setosa). The sample sizes for each group are provided on the lower part of the plot. All figures are clearly labelled at the top to remind users of the filename and plot type. This reduces errors throughout the digitisation process.

255 When completed `metaDigitise()` will write the digitised data as a
256 `metaDigitise` object to a RDS file in the `caldat` directory, such that our new
257 directory structure is as follows

```
*FiguresToExtract/  
  + caldat/  
    + 001_Anderson_1935_Fig1  
    + 001_Anderson_1935_Fig1.png
```

258 Users can access the `metaDigitise` object created (`001_Anderson_1935_Fig1`) at
259 any time using the `metaDigitise()` function. In the R console, the summarised

260 data for the digitised figure can be printed on screen or even written to a .csv
261 file:

```
R> digitised_data
```

filename	group_id	variable	mean	error	error_type	n	r	sd	plot_type
001_Anderson_1935_Fig1.png	setosa	Sepal width (mm)	3.42	0.40	sd	39	0.75	0.40	scatterplot
001_Anderson_1935_Fig1.png	setosa	Sepal length (mm)	5.00	0.38	sd	39	0.75	0.38	scatterplot
001_Anderson_1935_Fig1.png	versicolor	Sepal width (mm)	2.77	0.32	sd	44	0.52	0.32	scatterplot
001_Anderson_1935_Fig1.png	versicolor	Sepal length (mm)	5.95	0.53	sd	44	0.52	0.53	scatterplot

262 The mean for each of the two variables, along with the two species, are provided.
263 Since this is a scatterplot, the user also gets the Person's correlation coefficient
264 between sepal length and width for each species. These match reasonably well
265 with the actual means of sepal length and width for each of the species in the full
266 'iris' dataset:

	Species	meanSL	meanSW
1	setosa	5.006	3.428
2	versicolor	5.936	2.770

267 One thing anyone with a familiarity with the iris dataset will notice is that the
268 sample sizes for each of these species (which are $n = 50$ each) are quite a bit
269 lower. This is an example of some of the challenges when extracting data from
270 scatter plots. Often data points will overlap with each other making it impossible
271 (without having the real data) to know whether this is a problem. However, a
272 meta-analyst will probably realise that the sample sizes here conflict with what is
273 reported in the paper. Hence, **metaDigitise** also provides the user with options to
274 input the sample sizes directly (see Editing section below), even for scatter plots
275 and histograms where, strictly speaking, this should not be necessary.

276 Nonetheless, it is important to recognise the impact that overlapping points can
277 have on summary statistics, particularly its effects on standard deviation (SD)
278 and standard error (SE). Here, the mean point estimates are nearly exactly the
279 same as the true values, but the SD's are slightly over-estimated:

	Species	meanSL	meanSW
1	setosa	0.3524897	0.3790644
2	versicolor	0.5161711	0.3137983

280 **Adding new figures**

281 Users can add additional figures as new papers with relevant information are
282 found. Each figure should be in its own file with unique naming, even if a single
283 paper has multiple figures for extraction. For example, another paper on
284 different populations (and one new species) of iris contained two additional
285 figures where important data could be extracted. These figures can simply be
286 named accordingly and added directly to the same extraction folder:

```
*FiguresToExtract/  
+ caldat/  
  + 001_Anderson_1935_Fig1  
+ 001_Anderson_1935_Fig1.png  
+ 002_Doe_2013_Fig1.png  
+ 002_Doe_2013_Fig3.png
```

287 The user has already processed one figure (001_Anderson_1935_Fig1.png). We
288 can tell this because the caldat folder has digitised data in it

289 (caldat/001_Anderson_1935_Fig1). Now the user has two new figures that have
290 not yet been digitised. This example will nicely demonstrate how users can easily
291 pick up from where they left off and how all previous data gets re-integrated. It
292 will also demonstrate how different plot types are handled. All we have to do to
293 begin, is again, provide the directory where all the figures are located:

```
R> digitised_data <- metaDigitise("../FiguresToExtract", summary = TRUE)
```

294 The user gets the same set of prompts and simply chooses option one. This will
295 permit users to digitise new figures, and will integrate previously completed
296 digitisations along with newly digitised data together at the end of the session, or
297 when the user decides to quit. This time, 001_Anderson_1935_Fig1.png is ignored
298 and the new plots cycle on screen; first 002_Doe_2013_Fig1.png and then
299 002_Doe_2013_Fig3.png. Since there are a few different figure types, the user
300 answers the first question in the R console as "d":

```
Are all plot types Different or the Same? (d/s)
```

```
R> d
```

```
**** NEW PLOT ****
```

```
mean_error and boxplots should be vertically orientated
```

```
  -  
  |  
I.E. o   NOT  |-o-|  
  |  
  -
```

If they are not then chose flip to correct this.

If figures are wonky, chose rotate.

Otherwise chose continue

Flip, rotate or continue (f/r/c)

R> c

Please specify the plot_type as either:

m: Mean and error

b: Box plot

s: Scatter plot

h: Histogram

R> m

301 Here, the user specifies the new plot type as m for 002_Doe_2013_Fig1.png
302 because the user has a plot of the mean and error of sepal length for each of the
303 three species. The user is then prompted a bit differently from our scatter plot as
304 the x-axis is not needed for calibration:

What is the y variable?

R> Sepal length

On the Figure, click IN ORDER:

y1, y2

Step 1 ----> Click on y1

```
|  
|  
|  
|  
y1  
|-----
```

Step 2 ----> Click on y2

```
|  
y2  
|  
|  
|  
|  
|-----
```

What is the value of y1 ?

R> 5

What is the value of y2 ?

R> 6.5

Re-calibrate? (y/n)

R> n

Do you know sample sizes? (y/n)

R> y

If there are multiple groups, enter unique group identifiers (otherwise press enter)

Group identifier:

R> setosa

Group sample size:

R> 50

Click on Error Bar, followed by the Mean

Add group, Edit Group, Delete group or Finish plot? (a/e/d/f)

R> a

305 Again, `metaDigitise()` will simply guide the user through digitising each of
306 these figures describing to them exactly what needs to be done. At any point if
307 mistakes are made the user can choose relevant options to edit or correct things
308 before ending the figure. This process continues for each plot so long as the user
309 would like to continue and after completing a single plot the user is always
310 prompted as follows:

Do you want continue: 1 plots out of 2 plots remaining (y/n)

R> y

311 This continues until users have completed all non-digitised figures in the folder,
312 at which point `metaDigitise()` concatenates the new data with previously
313 digitised data in the object:

```
data
      filename      group_id      variable  mean  error error_type n    r    sd  plot_type
001_Anderson_1935_Fig1.png    setosa  Sepal width (mm)  3.42  0.40  sd      39  0.75  0.40  scatterplot
001_Anderson_1935_Fig1.png    setosa  Sepal length (mm)  5.00  0.38  sd      39  0.75  0.38  scatterplot
001_Anderson_1935_Fig1.png  versicolor  Sepal width (mm)  2.77  0.32  sd      44  0.52  0.32  scatterplot
001_Anderson_1935_Fig1.png  versicolor  Sepal length (mm)  5.95  0.53  sd      44  0.52  0.53  scatterplot
      002_Doe_2013_Fig1.png    setosa  Sepal length      5.00  0.11  se      50  NA    0.78  mean_error
      002_Doe_2013_Fig1.png  viriginica  Sepal length      6.59  0.18  se      50  NA    1.26  mean_error
      002_Doe_2013_Fig1.png  versicolor  Sepal length      5.94  0.14  se      50  NA    1.01  mean_error
      003_Doe_2013_Fig3.png    catana  Sepal length      4.95  0.36  sd      50  NA    0.36  histogram
```

314 Re-importing, Editing and Plotting Previously

315 Digitised data

316 A particularly useful feature of **metaDigitise** is its ability to re-import, edit and
317 re-plot previously digitised figures. We can do this from the initial options from
318 `metaDigitise()`

```
R> digitised_data <- metaDigitise("../FiguresToExtract")
```

```
Do you want to...
```

- 1: Process new images
- 2: Import existing data
- 3: Edit existing data

```
Selection:
```

319 If the user chooses "Import existing data", they have the option of either 1)
320 importing data from all digitised images or 2) importing data from a single
321 image that has been digitised. If 2, then a list of files are provided to the user
322 that they can select. Editing existing data allows users to easily re-plot or edit
323 information or digitisations that have previously be done for any plot. This is
324 accomplished by guiding the user through a new set of options:

Choose how you want to edit files:

- 1: Cycle through images
- 2: Choose specific file to edit
- 3: Enter previously omitted sample sizes

Selection:

325 If the user is unsure about the name of the specific figure they need to edit or
326 simply want to just check the digitisations of figures they can choose "Cycle
327 through images", which will bring up each figure, one by one, overlaying the
328 calibrations, group names (if they exist), sample sizes (if they were entered) and
329 the selected points. The user will then be given the choice to edit individual
330 images. Alternatively, choosing option 2, will bring up a list of the completed
331 files in the folder and the specific file can be chosen, at which point it will be
332 replotted. Either of these options will cycle through a number of questions
333 asking the user what they would like to edit:

Edit rotation? If yes, then the whole extraction will be redone (y/n)

R> n

Change plot type? If yes, then the whole extraction will be redone (y/n)

R> n

Variable entered as:

R> *Sepal length*

Rename Variables (y/n)

R> n

Edit calibration? (y/n)

R> n

Re-extract data (y/n)

R> y

Change group identifier? (y/n)

R> n

Add group, Delete group or Finish plot? (a/d/f)

R> d

1: setosa

2: versicolor

3: virginica

Selection:

R> 2

Add group, Delete group or Finish plot? (a/d/f)

R> a

334 A whole host of information can be edited including the rotation, plot type, the
335 variable name(s) that were provided, the calibration and even the digitisation of
336 groups. When editing the `metaDigitise` object is re-written to the `caldat` folder
337 and the edits are immediately integrated into the existing object once
338 complete.

339 **Additional Features**

340 **Figure Rotation and Adjustment**

341 Figures may have been extracted from old publications, for example from
342 scanned images, and so are not perfectly orientated on the image. This will make
343 the calibration of the points in the figure from the image problematic.
344 `metaDigitise()` allows users to rotate the image. By clicking two points on the
345 x-axis, `metaDigitise` calculates the angle needed to rotate the image so the x-axis
346 is horizontal, and rotates it. (Figure 5A,B)

347 Furthermore, some figures, including mean and error, boxplots or histograms,
348 may be presented with horizontal bars. `metaDigitise()` assumes that the bars
349 are vertical, but allows the user to flip the image so that the bars are vertical if
350 provided horizontally (Figure 5C,D).

351 Obtaining Processed Data

352 While `metaDigitise()` provides users with the summary statistics by default,

353 for all plot types, in many cases the user may actually be interested in obtaining

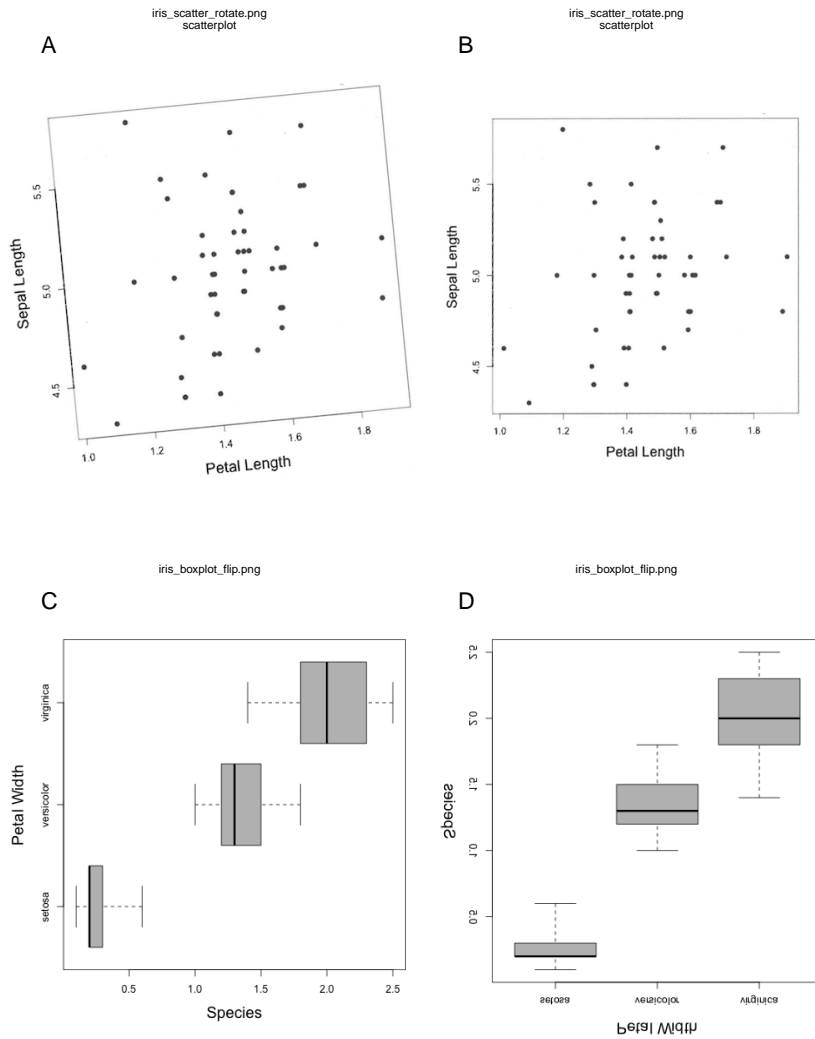


Figure 5: Figure rotation. A) and B) show how non-aligned images can be re-aligned through user defined rotation. C) and D) show how figures can be re-orientated so as to aid data input.

354 the processed digitised data from scatter plots (i.e. calibrated points). This is
355 very easy to do by changing the default `summary` argument from `TRUE` to
356 `FALSE` in `metaDigitise()`. Instead of providing the user with summary
357 statistics it will return a list containing four slots for each of the figure types
358 (mean error, box plot, histogram and scatter plots). An example of a data object
359 returned from digitising figures is as follows:

```
>R str(data)

List of 3
 $ mean_error :List of 1
  ..$ 002_Doe_2013_Fig1.png:'data.frame': 3 obs. of 5 variables:
  .. ..$ id      : Factor w/ 3 levels "setosa","versicolor",...: 1 2 3
  .. ..$ mean    : num [1:3] 5 5.93 6.59
  .. ..$ error   : num [1:3] 0.111 0.148 0.178
  .. ..$ n      : num [1:3] 50 50 50
  .. ..$ variable: chr [1:3] "Sepal length" "Sepal length" "Sepal length"
 $ hist       :List of 1
  ..$ 003_Doe_2013_Fig3.png:'data.frame': 8 obs. of 3 variables:
  .. ..$ midpoints: num [1:8] 4.3 4.5 4.7 4.9 5.1 ...
  .. ..$ frequency: num [1:8] 4 5 7 12 11 6 2 3
  .. ..$ variable : chr [1:8] "Sepal length" "Sepal length" ...
 $ scatterplot:List of 1
  ..$ 001_Anderson_1935_Fig1.png:'data.frame': 83 obs. of 8 variables:
  .. ..$ id      : Factor w/ 2 levels "setosa","versicolor": 1 1 1 1 1 ...
  .. ..$ x      : num [1:83] 2.3 2.9 3 3 3 ...
```

```
.. ..$ y          : num [1:83] 4.5 4.4 4.41 4.3 4.8 ...
.. ..$ group      : num [1:83] 1 1 1 1 1 1 1 1 1 1 ...
.. ..$ col        : Factor w/ 2 levels "red","green": 1 1 1 1 1 1 1 1 1 ...
.. ..$ pch        : num [1:83] 19 19 19 19 19 19 19 19 19 ...
.. ..$ y_variable: chr [1:83] "Sepal length (mm)" "Sepal length (mm)" ...
.. ..$ x_variable: chr [1:83] "Sepal width (mm)" "Sepal width (mm)" ...
```

360 Here, the user can easily access the list of processed scatter plot data by simply
361 extracting the scatter plot slot:

```
>R scatterplot <- data$scatterplot
```

362 Adding sample sizes to previous Digitisations

363 In many cases important information, such as sample sizes, may not be readily
364 available or clear when digitising figures. In these circumstances users will have
365 answered ‘no’ to the question about whether they have sample sizes or not while
366 digitising. To expedite finding and adding in these sample sizes to do the
367 necessary calculations (if for example a figure presented 95% CI’s or standard
368 errors), `metaDigitise()` has a specific edit option that allows users to enter in
369 previously omitted sample sizes. It works by first identifying the missing sample
370 sizes in the digitised output, re-plotting the relevant figure and then prompting
371 the user to enter the sample sizes for the relevant groups in the figure, one by
372 one. As an example, assume that we were missing sample sizes for two groups in
373 002_Doe_2013.Fig1.png:

filename	group_id	variable	mean	error	error_type	n	r	sd	plot_type
002_Doe_2013_Fig1.png	setosa	Sepal length	5.00	0.11	se	NA	NA	NA	mean_error


```
002_Doe_2013_Fig1.png  viriginica  Sepal length      6.59  0.18  se      NA  NA      NA  mean_error
```

374 Here, we can see that we are missing the sample sizes for setosa and viriginica,
375 and as a result, sd is not calculated because `metaDigitise()` needs this
376 information to make the calculation. If the user found this information after
377 contacting the authors for clarification then they can add these back in as
378 follows:

```
R> digitised_data <- metaDigitise("../FiguresToExtract")
```

Do you want to...

- 1: Process new images
- 2: Import existing data
- 3: Edit existing data

Selection:

```
R> 3
```

Choose how you want to edit files:

- 1: Cycle through images
- 2: Choose specific file to edit
- 3: Enter previously omitted sample sizes

Selection:

```
>R 3
```

379 `metaDigitise()` will replot the figure after this and list, only the groups missing
380 data, for which the user can then update the data. This is then re-integrated
381 back into the data automatically and the sd calculated.

```
Group " setosa ": Enter sample size
```

```
R> 50
```

```
Group " viriginica ": Enter sample size
```

```
R> 50
```

382 **Inter-observer Variability and Validation**

383 **Inter-observer variability in digitisations**

384 In order to evaluate the consistency of digitisation using **metaDigitise** between
385 users, we simulated a dataset of two traits with two different groups. These data
386 were then used to construct plots of the four different types (scatterplot, mean
387 and error, histogram and boxplots). Each variable was plotted twice for each
388 given plot type (figures were modified slightly to give users a sense that they
389 were digitising new data) generating a total of 14 figures. 15 independent
390 digitisers were provided with a directory with all 14 figures in a randomised
391 order. Digitisers ran **metaDigitise** on their own computers, across different
392 operating systems (including Mac, Windows and Linux). Digitisers varied in
393 their level of experience, from people with experience of meta-analyses or
394 comparative work to those without any science background. We asked users to

395 digitise all 14 figures and collected the mean, standard deviation and correlation
396 coefficient (for scatterplots) generated by `metaDigitise()` for every plot
397 digitised. We transformed these data to standardized differences as

$$\frac{\theta - \hat{\theta}}{\hat{\theta}} \quad (8)$$

398 where θ is the estimate value and $\hat{\theta}$ is the true value, meaning that deviations
399 were percentage differences from the true summary statistics. The correlation
400 coefficient deviation was not divided by the true value, as it is already on a
401 standardised scale. This deviation can be seen as a measure of bias. The
402 resulting data was used to assess between- and within- user variability (i.e., the
403 intra-class correlation coefficient) in the data. This was done using linear mixed
404 effect models with user identify as a random effect. Standardised mean, standard
405 deviation and correlation coefficients were used as response variables in separate
406 models. Sampling variance for ICC estimates was generated based on 1000
407 parametric bootstraps of the model and the significance was tested using
408 likelihood ratio tests. These models were run using the **lme4** (Bates et al., 2015)
409 and **rptR** (Stoffel, Nakagawa & Schielzeth, 2017) packages in R.

410 If digitisations were consistent across all users then we should find no significant
411 between user variability in the data. Indeed, across plot types we found no
412 evidence for any inter-observer variability in digitisations for the mean (ICC = 0,
413 95% CI = 0 to 0.029, $p = 1$), standard deviation (ICC = 0, 95% CI = 0 to 0.033,
414 $p = 0.5$) or correlation coefficient (ICC = 0.053, 95% CI = 0 to 0.296, $p =$
415 0.377). There were was little bias between digitised and true values, on average

416 1.63% (mean = 0.02%, SD = 4.9%, $r = -0.03\%$) and overall there were only
417 small absolute differences between digitised and true values, deviating, on
418 average 2.18% (mean = 0.40%, SD = 5.81%, $r = 0.33\%$) across all three
419 summary statistics.
420 SD estimates from digitisations are clearly more prone to error than means or
421 correlation coefficients. If the mean absolute difference is calculated for each
422 plot type, we can see that this effect is driven mainly by extraction from boxplots
423 and histograms (% difference):

boxplot	histogram	mean_error	scatterplot
15.805	5.210	1.500	0.433

424 This is because SD estimation from the summary statistics extracted from
425 boxplots is more error prone, especially at small sample sizes (Wan et al.,
426 2014).

427 **Testing the accuracy of digitisations**

428 To test how accurate **metaDigitise** is at matching points to their true values, we
429 generated four random scatterplots, each with 20 data points, and digitised these
430 with `metaDigitise()`. This was done by one digitiser, as there is no detectable
431 between user variation. Data digitised using **metaDigitise** was essentially
432 perfectly correlated with the true simulated data for both the x -variable
433 (Pearson's correlation; $r = 0.9999915$, $t = 2137.4$, $df = 78$, $p < 0.001$) and
434 y -variable ($r = 0.9999892$, $t = 1897.8$, $df = 78$, $p < 0.001$).

435 Discussion and Conclusions

436 Although **metaDigitise** is already very flexible, and provides functionality not
437 seen in any other package (Table 1) it is clear that there are some functions that
438 it does not perform. A notable feature that **metaDigitise** lacks is automated
439 point detection. Point detection is available in several packages (Table 1).
440 However, from our experience of using these functions, manual digitising is more
441 reliable and often equally as fast. Particularly given that calibration (for point
442 detection) needs to be done for each plot individually in any case. Additionally,
443 auto-detection often misses many points which then subsequently need to be
444 manually added. Based on tests of **metaDigitise** (see above), figures can be
445 extracted in around 1-2 minutes, including the entry of metadata. As a result, we
446 do not believe that current automated point detection provides substantial
447 benefits in terms of time or accuracy.

448 Another feature that **metaDigitise** (currently) lacks, is an ability to zoom in on
449 plots. Zooming may enable users to gain greater accuracy when clicking on
450 points. However, from our own experience (and indeed from the results above), if
451 you are using a reasonably sized screen then the accuracy is already high from
452 these programs, and there is not much gain to be had from zooming in on points
453 in many circumstances.

454 In contrast to some other packages, **metaDigitise** currently also does not extract
455 lines from figures. In our own experience, line extraction is not particularly
456 useful for meta-analysis, although we recognise that it may be useful in other
457 fields. Should a user like to extract lines with **metaDigitise**, we would

Function	metaDigitise	GraphClick ¹	DataThief ²	DigitizeIt ³	WebPlotDigitizer ⁴	metagear ⁵	digitize ⁶
Scatterplots	✓	✓	✓	✓	✓	✓ ⁷	✓
Mean and error plots	✓	✓	✓	×	×	✓ ⁷	×
Boxplots	✓	×	×	×	×	×	×
Histograms	✓	×	×	×	✓ ⁷	×	×
Graph rotation ⁸	✓	✓	✓	✓	✓	×	×
Groups	✓	✓	×	✓	✓	×	×
Entry of metadata	✓	×	×	×	×	×	×
Summarising data	✓	×	×	×	×	×	×
Multiple image processing	✓	×	×	×	×	×	×
Reproducible ⁹	✓	✓	✓	×	✓	×	×
Automated point detection	×	✓	×	✓	✓	✓	×
Line extraction	×	✓	✓	✓	✓	×	×
Zoom	×	✓	✓	✓	✓	×	×
Log axis	×	✓	✓	✓	✓	×	×
Dates	×	×	✓	×	✓	×	×
Asymmetric error bars	×	×	✓	×	✓	×	×
Freeware	✓ ¹⁰	✓ ¹¹	✓ ¹¹	×	✓ ¹¹	✓ ¹⁰	✓ ¹⁰

¹ Arizona-Software (2008) ² Tummers (2006) ³ Bormann (2012) ⁴ Rohatgi (2017) ⁵ Lajeunesse (2016) ⁶ Poisot (2011)

⁷ Only automated, no manual extraction.

⁸ Or handles rotated graphs.

⁹ Allows saving, re-plotting and editing of data extraction.

¹⁰ R package.

¹¹ Standalone software.

Table 1: Comparison of functionality between different digitisation softwares.

458 recommend extracting data as a scatter plot, and clicking along the line in
459 question. A model can then be fitted to these points (setting the argument
460 "summary = FALSE" in **metaDigitise** - will provide access to the processed
461 data) to estimate the parameters needed.

462 Finally, **metaDigitise** currently does not allow for asymmetric error bars. At
463 present this is a deliberate omission, as it is not clear how best to derive SD from
464 such data, given also that such asymmetric error bars may represent different
465 things in different figures.

466 Descriptive statistics are usually the most robust sources of information for
467 calculating effect size statistics (Noble et al., 2017). These are most often
468 presented in figures. Users may therefore also want to compare effect size
469 estimates from inferential statistics with those derived from descriptive statistics
470 (obtained for example using **metaDigitise**) from a paper. Comparing these
471 different effects sizes can be useful in identifying uncertainties and problems
472 within a paper. In the future, we hope to provide functions to easily convert
473 inferential statistics to standardised effect size estimates, which can seamlessly be
474 integrated with summary statistics from **metaDigitise**, to calculate equivalent
475 standardised effect size estimates and their sampling variance.

476 Increasing the reproducibility of figure extraction for meta-analysis and making
477 this laborious process more streamlined, flexible and integrated with existing
478 statistical software will go a long way in facilitating the production of high
479 quality meta-analytic studies that can be updated in the future. We believe that
480 **metaDigitise** will improve this research synthesis pipeline, and will hopefully
481 become an integral package that can be added to the meta-analysts toolkit.

482 **Acknowledgments**

483 We thank the I-DEEL group at UNSW for incredibly useful feedback, and a host
484 of colleagues for testing, providing feedback and digitising including: Rose
485 O’Dea, Fonti Kar, Malgorzata Lagisz, Julia Riley, Diego Barneche, Erin
486 Macartney, Ivan Beltran, Gihan Samarasinghe, Dax Kellie, Jonathan Noble,
487 Yian Noble and Alison Pick. JLP was supported by a Swiss National Science
488 Foundation Early Mobility grant (P2ZHP3_164962), DWAN was supported by an
489 Australian Research Council Discovery Early Career Research Award
490 (DE150101774) and UNSW Vice Chancellors Fellowship and SN an Australian
491 Research Council Future Fellowship (FT130100268).

492 **References**

- 493 Arizona-Software (2008) *GraphClick Software, Version 3.0*.
- 494 Bates, D., Maechler, M., Bolker, B. & Walker, S. (2015) Fitting Linear
495 Mixed-Effects Models Using lme4. *Journal of Statistical Software*, **67**, 1–48.
- 496 Bland, M. (2015) Estimating Mean and Standard Deviation from the Sample
497 Size, Three Quartiles, Minimum, and Maximum. *International Journal of*
498 *Statistics in Medical Research*, **4**, 57–64.
- 499 Borenstein, M., Hedges, L., Higgins, J. & Rothstein, H. (2009) Introduction to
500 meta-analysis. *John Wiley Sons. Ltd. West Sussex, UK*.
- 501 Bormann, I. (2012) *Digitizelt Software, Version 2.0*. Braunschweig, Germany.

- 502 Glass, G. (1976) Primary, secondary, and meta-analysis research. *Educational*
503 *Researcher*, **5**, 3–8.
- 504 Ihle, M., Winney, I.S., Krystalli, A. & Croucher, M. (2017) Striving for
505 transparent and credible research: practical guidelines for behavioral
506 ecologists. *Behavioral Ecology*, **28**, 348–354.
- 507 Koricheva, J., Gurevitch, J. & Mengersen, K. (2013) Handbook of Meta-Analysis
508 in Ecology and Evolution. *Princeton University Press, Princeton, New Jersey*.
- 509 Lajeunesse, M.J. (2016) Facilitating systematic reviews, data extraction, and
510 meta-analysis with the metagear package for R. *Methods in Ecology and*
511 *Evolution*, **7**, 323–330.
- 512 Nakagawa, S., Noble, D.W., Senior, A.M. & Lagisz, M. (2017) Meta-evaluation of
513 meta-analysis: ten appraisal questions for biologists. *BMC Biology*, **15**, 18;
514 DOI 10.1186/s12915-017-0357-7.
- 515 Noble, D.W., Lagisz, M., O’Dea, R.E. & Nakagawa, S. (2017) Nonindependence
516 and sensitivity analyses in ecological and evolutionary meta-analyses.
517 *Molecular Ecology*, **26**, 2410–2425.
- 518 Parker, T.H., Forstmeier, W., Koricheva, J., Fidler, F., Hadfield, J., En Chee, Y.,
519 Kelly, C.D., Gurevitch, J. & Nakagawa, S. (2016) Transparency in Ecology and
520 Evolution: Real Problems, Real Solutions. *Trends in Ecology and Evolution*,
521 **31**, 711–719.
- 522 Peng, R.D. (2011) Reproducible research in computational science. *Science*, **334**,
523 1226.

- 524 Peng, R.D., Dominici, F. & Zeger, S.L. (2006) Reproducible epidemiologic
525 research. *American Journal of Epidemiology*, **163**, 783–789.
- 526 Poisot, T. (2011) The digitize package: extracting numerical data from
527 scatterplots. *The R Journal*, **3**, 25–26.
- 528 Rohatgi, A. (2017) *WebPlotDigitizer Software, Version 4.0*. Austin, Texas, USA.
- 529 Sandve, G.K., Nekrutenko, A., Taylor, J. & Hovig, E. (2013) Ten simple rules for
530 reproducible computational research. *PLoS Computational Biology*, **9**,
531 e1003285.
- 532 Stoffel, M.A., Nakagawa, S. & Schielzeth, H. (2017) rptR: repeatability
533 estimation and variance decomposition by generalized linear mixed-effects
534 models. *Methods in Ecology and Evolution*, **8**, 1639–1644.
- 535 Tummers, B. (2006) *DataThief Software, Version 3.0*.
- 536 Wan, X., Wang, W., Liu, J. & Tong, T. (2014) Estimating the sample mean and
537 standard deviation from the sample size, median, range and/or interquartile
538 range. *BMC Medical Research Methodology*, **14**, 135.