Automated, semi-automated, and manual extraction of numerical data from scientific images with the *juicr* package for *R* [BETA]

Marc J. Lajeunesse (lajeunesse@usf.edu)

University of South Florida, February 18th 2026 (vignette v. 0.2 for juicr v. 0.2)

Marc J. Lajeunesse

Table of Contents

Introduction

The **juicr** package for R contains tools for facilitating the extractions of numerical data from scientific images – like scatter-plots, bar-plots, and other charts/figures found in publications. Below is a description of functionalities and layout.

Updates to this vignette will be posted on our research webpage at USF.

For the source code of **juicr** see: https://CRAN.R-project.org/package=juicr.

Acknowledgements

I thank everyone who watched my *YouTube* course Hard-boiled Synthesis and reached out to me about using **juicr** – you gave me the final push to complete this old project I started way back and abandoned in 2017!

How to cite? TBA, but for this beta version maybe:

Lajeunesse, M.J. (2021) Squeezing data from scientific images with the **juicr** package for *R*. R package, v. 0.1. CRAN

Installation and Dependencies

juicr has an external dependency that needs to be installed and loaded prior to use in R. This is the **EBImage** R package (Pau *et al.* 2010) available only from Bioconductor repository.

To properly install **juicr**, use the following script in R:

```r
# first load Bioconductor resources needed to install the EBImage package
# and accept/download all of its dependencies
install.packages("BiocManager");
BiocManager::install("EBImage")

# then load juicr
library(juicr)
```

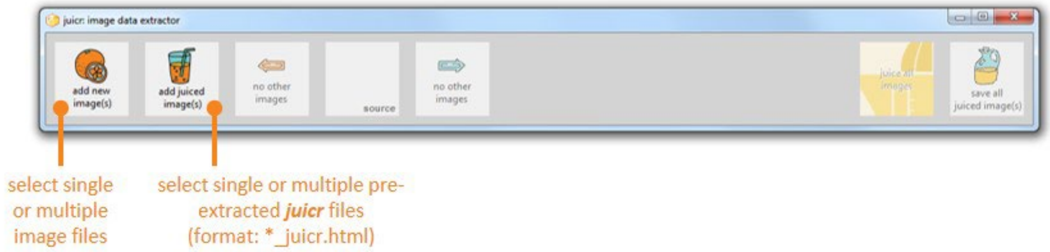
Finally for Mac OS users, installation is sometimes not straighforward, as the `GUI_juicr()` requires the Tcl/Tk GUI toolkit to be installed. You can get this toolkit by making sure that the latest X11 application (xQuartz) is installed from here: https://www.xquartz.org/.

Report a bug? Have comments or suggestions?

Please email me any bugs, comments, or suggestions and I'll try to include them in future releases: lajeunesse@usf.edu. Also try to include **juicr** in the subject heading of your email. Finally, I'm open to almost anything, but expect a lag before I respond and/or new additions are added.

GUI layout and loading images

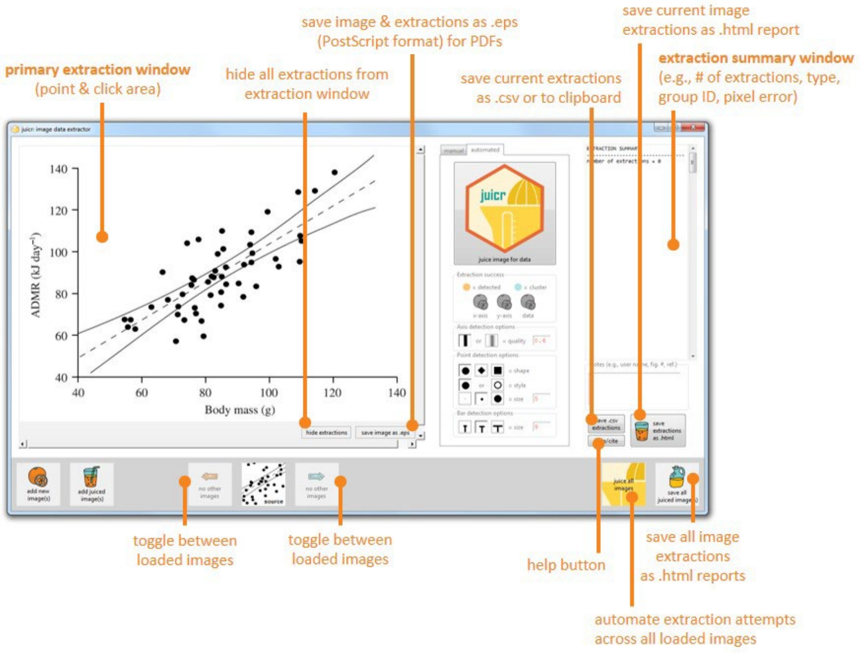
When running **juicr** without a file specified, the layout is simple:


juicr: image data extractor
add new image(s)
add juiced image(s)
no other images
source
no other images
juice all images
save all juiced image(s)
select single or multiple image files
select single or multiple pre-extracted *juicr* files (format: *_juicr.html)

Images can be loaded into **juicr** using the *add new image(s)* button. Alternatively one or many images (as a vector of file name strings) can be included via console:

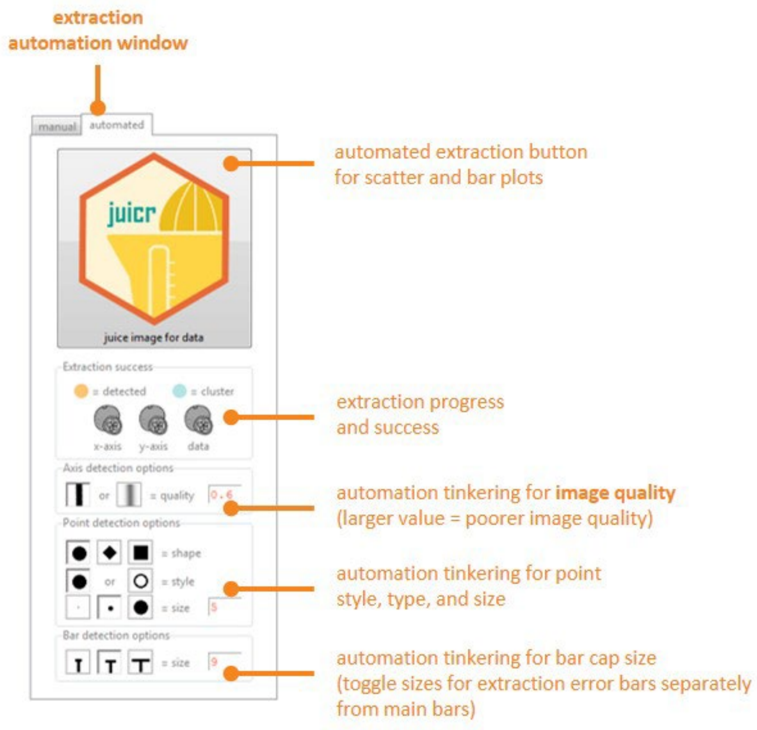
```
# then load juicr
library(juicr)
GUI_juicr("Kam_et_al_2003_Fig2.jpg")
# or many files
GUI_juicr(c("Kam_et_al_2003_Fig2.jpg", "Kortum_and_Acymyan_2013_Fig4.jpg"))
```

If an image is loaded, the main window will look like this:


save image & extractions as .eps (PostScript format) for PDFs
save current image extractions as .html report
primary extraction window (point & click area)
hide all extractions from extraction window
save current extractions as .csv or to clipboard
extraction summary window (e.g., # of extractions, type, group ID, pixel error)
juicr: image data extractor
140
120
100
80
60
40
ADMR (kJ day⁻¹)
40
60
80
100
120
140
Body mass (g)
hide extractions
save image as .eps
manual
automated
juicr
juice image for data
EXTRACTION SUMMARY
number of extractions = 0
save .csv extractions
save extractions as .html
add new image(s)
add juiced image(s)
no other images
source
no other images
juice all images
save all juiced image(s)
toggle between loaded images
toggle between loaded images
help button
save all image extractions as .html reports
automate extraction attempts across all loaded images

Automated extraction functionality

The center section of **juicr** includes two options: *automated* or *manual* tools for extracting data. The automated tools include:



extraction automation window

automated extraction button for scatter and bar plots

extraction progress and success

automation tinkering for **image quality** (larger value = poorer image quality)

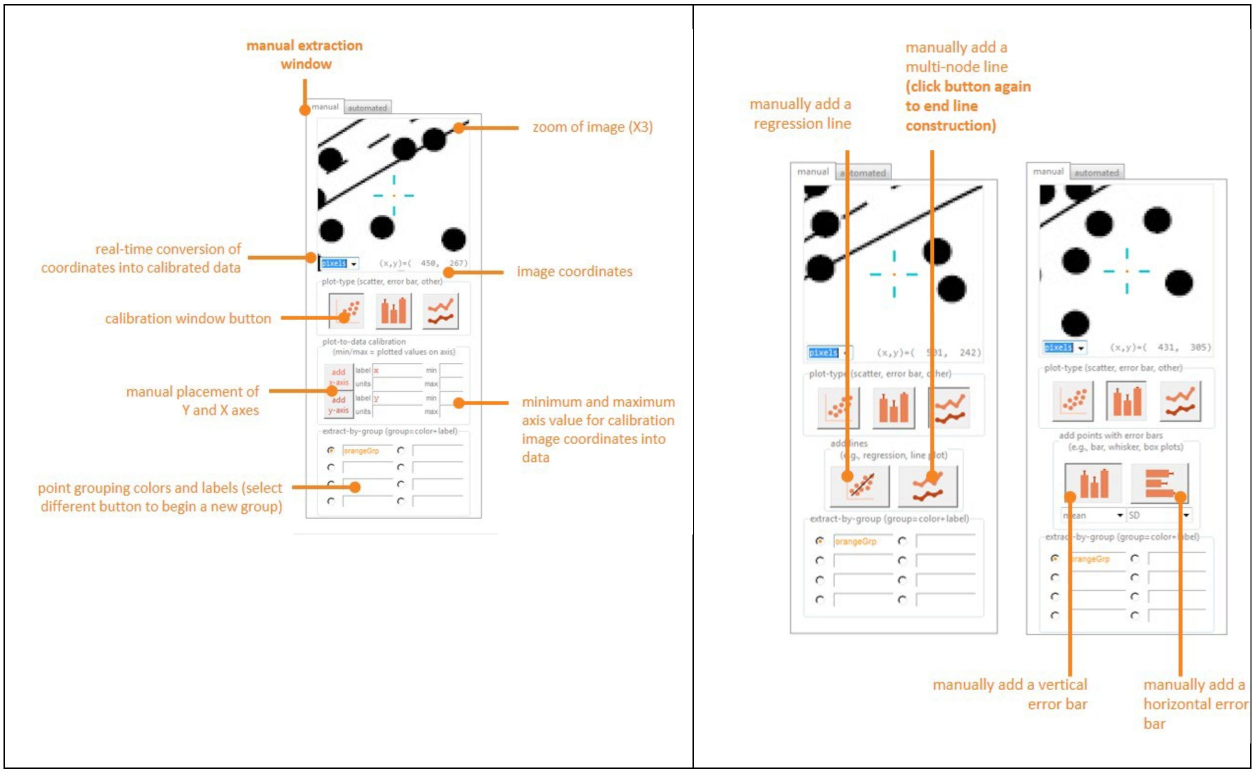
automation tinkering for point style, type, and size

automation tinkering for bar cap size (toggle sizes for extraction error bars separately from main bars)

By pressing the large **juicr-hex** button, it will first determine automatically plot type (e.g., scatter or bar for now), and then begin extracting data from the image. Successes of extractions, such as whether the y-axis was detected, will presented as an orange orange (detected), and failures as gray oranges (not-detected). Also included a suite of semi-automated tinkering options to help **juicr** detect the desired image objects.

Manual extraction functionality

A large diversity of manual extraction tools are available:


manual extraction window
zoom of image (X3)
real-time conversion of coordinates into calibrated data
image coordinates
calibration window button
manual placement of Y and X axes
minimum and maximum axis value for calibration image coordinates into data
point grouping colors and labels (select different button to begin a new group)
manually add a regression line
manually add a multi-node line (click button again to end line construction)
manually add a vertical error bar
manually add a horizontal error bar
manual
automated
pixels
plot-type (scatter, error bar, other)
plot-to-data calibration (min/max = plotted values on axis)
add x-axis
add y-axis
label
units
min
max
extract-by-group (group=color+label)
orangeGrp
add lines (e.g., regression, line plot)
add points with error bars (e.g., bar, whisker, box plots)
mean
SD

Saving extractions and generating reports

Currently, **juicr** offers extractions to be saved as .csv files, copied into clipboards, saved as postscript .eps files, or as a fully-embedded and standalone .html file that retains all information of extractions, **juicr** setup, and image modifications for permanent and replicable storage of data.

Here is an example of a report:
http://lajeunesse.myweb.usf.edu/juicr/Kam_et_al_2003_Fig2_juicr.html