The digitize Package: Extracting Numerical Data from Scatterplots

by Timothée Poisot

Abstract I present the small R package **digitize**, designed to extract data from scatterplots with a simple method and suited to small datasets. I present an application of this method to the extraction of data from a graph whose source is not available.

The package **digitize**, that I present here, allows a user to load a graphical file of a scatterplot (with the help of the `read.jpeg` function of the **ReadImages** package) in the graphical window of R, and to use the `locator` function to calibrate and extract the data. Calibration is done by setting four reference points on the original graph axis, two for the x values and two for the y values. The use of four points for calibration is justified by the fact that it makes calibrations on the axis possible, as y data are not taken into account for calibration of the x axis, and vice versa.

This is useful when working on data that are not available in digital form, e.g. when integrating old papers in meta-analyses. Several commercial or free software packages allow a user to extract data from a plot in image format, among which we can cite PlotDigitizer (`http://plotdigitizer.sourceforge.net/`) or the commercial package GraphClick (`http://www.arizona-software.ch/graphclick/`). While these programs are powerful and quite ergonomic, for some lightweight use, one may want to load the graph directly into R, and as a result get the data directly in R format. This paper presents a rapid digitization of a scatterplot and subsequent statistical analysis of the data. As an example, we will use the data presented by Jacques Monod in a seminal microbiology paper (Monod, 1949).

The original paper presents the growth rate (in terms of divisions per hour) of the bacterium *Escherichia coli* in media of increasing glucose concentration. Such a hyperbolic relationship is best represented by the equation

$$R = R_K \frac{C}{C_1 + C}$$

where R is the growth rate at a given concentration of nutrients C, R_K is the maximal growth rate, C_1 is the concentration of nutrients at which $R = 0.5R_K$. In R, this function is written as

```
MonodGrowth <- function(params, M) {
   with(params, rK*(M/(M1+M)))
}
```

In order to characterize the growth parameters of a bacterial population, one can measure its growth rate in different concentrations of nutrients. Monod (1949) proposed that, in the measured population, $R_K = 1.35$ and $C_1 = 22 \times 10^{-6}$. By using the **digitize** package to extract the data from this paper, we will be able to get our own estimates for these parameters.

Values of R_K and C_1 were estimated using a simple genetic algorithm, which minimizes the error function (sum of squared errors) defined by

```
MonodError <- function(params, M, y) {
   with(params,
        sum((MonodGrowth(params, M)-y)^2))
}
```

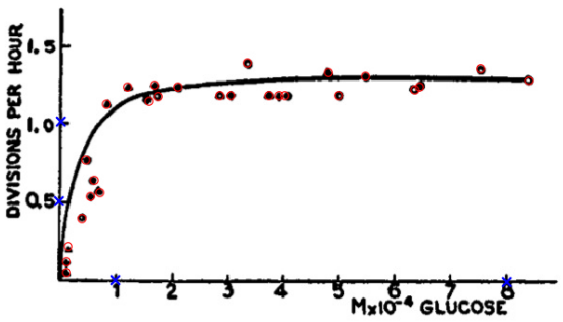

Figure 1: Original figure, as obtained after pointing and clicking the data. Calibration was made using the points $x_1 = 1$, $x_2 = 8$, $y_1 = 0.5$ and $y_2 = 1$. All the points that have been clicked are marked by a red point. The digitization of each series is stopped by right-clicking or pressing the esc. key.

The first step when using the **digitize** package is to specify four points on the graph that will be used to calibrate the axes. They must be in the following order : leftmost x, rightmost x, lower y, upper y. For the first two of them, the y value is not important (and vice versa). For this example, it is assumed that we set the first two points at $x_1 = 1$ and $x_2 = 8$, and the two last points at $y_1 = 0.5$ and $y_2 = 1$, simply by clicking in the graphical window at these positions (preferentially on the axes). It should be noted that it is not necessary to calibrate using the extremity of the axes.

Loading the figure and printing it in the current device for calibration is done by

```
cal <- ReadAndCal('monod.jpg')
```

Once the graph appears in the window, the user must input (by clicking on them) the four calibration points, marked as blue crosses. The calibration values will be stocked in the `cal` object, which is a list with x and y values. The next step is to read the data,

simply by pointing and clicking on the graph. This is done by calling the `DigitData` function, whose arguments are the type of lines/points drawn.

```
growth <- DigitData(col = 'red')
```

When all the points have been identified, the digitization is stopped in the same way that one stops the `locator` function, i.e. either by right-clicking or pressing the escape key (see `?locator`). The outcome of this step is shown in figure 1. The next step for these data to be exploitable is to use the calibration information to have the correct coordinates of the points. We can do this by using the function `Calibrate(data,calpoints,x1,x2,y1,y2)`, and the correct coordinates of the points of calibration (`x1`, `x2`, `y1` and `y2` correspond, respectively, to the points x_1, x_2, y_1 and y_2).

```
data <- Calibrate(growth, cal, 1, 8, 0.5, 1)
```

The internals of the function `Calibrate` are quite simple. If we consider $X = (X_1, X_2)$ a vector containing the x coordinates of the calibration points for the x axis on the graphic window, and $x = (x_1, x_2)$ a vector with their true value, it is straightforward that $x = aX + b$. As such, performing a simple linear regression of `x` against `X` allows us to determine the coefficients to convert the data. The same procedure is repeated for the y axis. One advantage of this method of calibration is that you do not need to focus on the y value of the points used for x calibration, and reciprocally. It means that in order to accurately calibrate a graph, you only need to have two x coordinates, and two y coordinates. Eventually, it is very simple to calibrate the graphic by setting the calibration points directly on the tick mark of their axis.

The object returned by `Calibrate` is of class `"data.frame"`, with columns `"x"` and `"y"` representing the x and y coordinates so that we can plot it directly. The following code produces Figure 2, assuming that `out` is a list containing the arguments of `MonodGrowth` obtained after optimization (`out$set`), and `paper` is the same list with the values of Monod (1949).

```
plot(data$x, data$y, pch=20, col='grey',
     xlab = 'Nutrients concentration',
     ylab = 'Divisions per hour')
points(xcal, MonodGrowth(out$set, xcal),
       type = 'l', lty = 1, lwd = 2)
points(xcal, MonodGrowth(paper, xcal),
       type = 'l', lty = 2)
legend('bottomright',
legend = c('data', 'best fit', 'paper value'),
          pch = c(20, NA, NA),
          lty = c(NA, 1, 2),
          lwd = c(NA, 2, 1),
          col = c('grey', 'black', 'black'),
          bty = 'n')
```

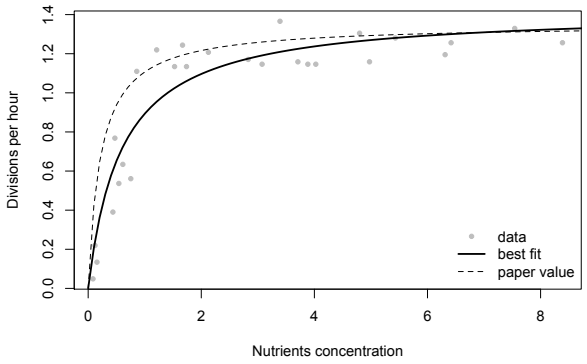

Figure 2: Result of the digitization. Original data are in grey, the proposed fit is in the dashed line, and the estimate realized from the raw data is in the bold line. Our estimates for the growth parameters are $R_K = 1.43$ and $C_1 = 62 \times 10^{-6}$.

While using the `MonodError` function with the proposed parameters yields a sum of squares of 1.32, our estimated parameters minimizes this value to 0.45, thus suggesting that the values of R_K and C_1 presented in the paper are not optimal given the data.

Conclusion

I presented an example of using the **digitize** package to reproduce data in an ancient paper that are not available in digital form. While the principle of this package is really simple, it allows for a quick extraction of data from a scatterplot (or any kind of planar display), which can be useful when data from old or proprietary figures are needed.

Acknowledgements

Thanks are due to two anonymous referees for their comments on an earlier version of this MS.

Bibliography

J. Monod. The growth of bacterial cultures. *Annual Reviews in Microbiology*, 3(1):371–394, 1949.

Timothée Poisot
Université Montpellier II
Institut des Sciences de l'Évolution, UMR 5554
Place Eugène Bataillon
34095 Montpellier CEDEX 05
France
tpoisot@um2.fr