RStudio - 03; Enzymology data

Jean-Yves Sgro

May 2, 2017

Table of Contents

[1 Enzymology data 1](#_Toc481487533)

[2 Data 2](#_Toc481487534)

[3 Method 3](#_Toc481487535)

[3.1 Create an RMarkdown document 3](#_Toc481487536)

[3.1.1 Top section 3](#_Toc481487537)

[3.1.2 Body text 4](#_Toc481487538)

[3.1.3 Report Output 4](#_Toc481487539)

[3.1.4 Making a "Freddie" report 4](#_Toc481487540)

[4 Fill-in the report 6](#_Toc481487541)

[4.1 Import and explore data 6](#_Toc481487542)

[4.2 Theoretical equation 9](#_Toc481487543)

[4.3 Nonlinear least square 9](#_Toc481487544)

[5 Theoretical curve 11](#_Toc481487545)

[6 Final Freddie report 13](#_Toc481487546)

[7 R Session. 14](#_Toc481487547)

[8 Acknowledgments 14](#_Toc481487548)

[9 REFRENCES: 15](#_Toc481487549)

# Enzymology data

Our student colleague and friend "Freddie" is running an enzymology experiment and is asking our help to present the result and update it later if necessary.

He first wants to plot the data with the Michaelis-Menten method (Michaelis and Menten 1913) and deduct values for the parameters of the equation.

*Michaelis and Menten showed that the rate of an enzyme-catalyzed reaction is proportional to the concentration of the enzyme–substrate complex predicted by the Michaelis–Menten equation.* (Michaelis et al. 2011)

The Michaelis-Menten equation can be written as a function of with constant parameters to be evaluated:

where is the value for the substrate also denoted in some equations, is the asymptote Vmax and is the value half-way between and the asymptote.

# Data

Freddie is not very good with electronic records and has written down the data on a piece of paper that contains the results if his experiements:

Freddie's data:

|  |  |
| --- | --- |
| S | v |
| 0 | 0.0 |
| 1 | 11.1 |
| 2 | 25.4 |
| 5 | 44.8 |
| 8 | 54.5 |
| 12 | 58.2 |
| 30 | 72.0 |
| 50 | 60.1 |

***How can we help Freddie to write his report?***

# Method

We will create a "dynamic" report for Freddie that can be updated with new data if necessary.

## Create an RMarkdown document

If we are continuing from the previous section we already have created an Rstudio project file *e.g.* Project\_1 potentially stored in a directory on the desktop.

You can create a new project or use this existing project Project\_1. The project will contain the files that we want to use or save.

You can start a new R Markdown file with the following menu cascade:

**File > New File > R R Markdown**

In the selection window that opens provide a name for your document, the default is "untitled" but it could be call *e.g.* Freddie1.Rmd where .Rmd is the standard R markdown filename extension.

The document will come pre-filled with sample text and data that we will need to erase.

### Top section

But first, note that the title you gave to your file is found at the very top of the document and will be used as the document title at the top of the final report. In turn this information is found within a structure bonded by 2 lines with 3 dashes:

For example my document had this default top section:

---  
title: "TEST"  
author: "Jean-Yves Sgro"  
date: "May 2, 2017"  
output:  
 html\_document:  
 toc: yes  
 pdf\_document:  
 toc: yes  
---

This is an important section in a special language called [YAML](http://yaml.org/) that provides a space to place special options.

For our immediate purpose this is sufficient for our goal and we can explore and tweak these options later.

### Body text

The rest of the document is written in R markdown and we can immediately get a glimpse of its structure organization.

The sample data is a built-in dataset about cars from the 1970's in tabular form, used here as a quick example.

We can recognise immediately the code that will be executed by R as marked by {r} and usually referred to as code chuncks.

The first chunck is a set-up parameter:

Here include=FALSE means that the code will be executed but the code itself will be not be shown in the final output.

The rest of the file is a simple plot and a simple table output.

### Report Output

To obtain an output from this page:

* click the Knit HTML button at the top of the Scripts quadrant.
* a new windowm, in HTML format will appear containing the report.

This is how we create dynamic documents with R code executed on the fly and the "narrative" story telling.

### Making a "Freddie" report

We can now erase what we don't need, that is everything except the first {r} chunk that we can keep.

We can then prepare the document for writing a nice report for Freddie. As stated before the Title of the report will be the title provided in the title: section in the YAML description at the top.

To start we can set-up the report with section names that we can fill later. For example:

---  
title: "Report for Freddie"  
author: "your name"  
date: "May 5, 2017"  
output: html\_document  
---  
  
# Introduction  
  
In this report we present ....  
  
# Data source and method  
  
Enzyme was from \*\*Promega\*\* etc.  
  
# Raw data  
  
The data was obtained from an experiement performed by.... on this date:...  
  
# Data exploration  
  
We can explore the data with a simple first plot describing the reaction....  
  
# Model building  
  
This enzymatic data follows a Michaelis-Menten plot   
and we can use a statistical method to create a model  
and calculate constant parameters Km and Vmax.  
  
# Final plot based on data and model  
  
This plot depicts...  
  
# Conclusion  
  
In conclusion we can report that...

This "raw" template can be filled as we go but for now it would be rendered as follows by the Knit HTML button. Note that the YAML section is not shown in the final output.

Introduction

In this report we present ....

Data source and method

Enzyme was from **Promega** etc.

Raw data

The data was obtained from an experiement performed by.... on this date:...

Data exploration

We can explore the data with a simple first plot describing the reaction....

Model building

This enzymatic data follows a Michaelis-Menten plot and we can use a statistical method to create a model and calculate constant parameters Km and Vmax.

Final plot based on data and model

This plot depicts...

Conclusion

In conclusion we can report that...

# Fill-in the report

Now that you have a rough template to fill-in, we can explore **how** we can fill it (!) with R code and descritions.

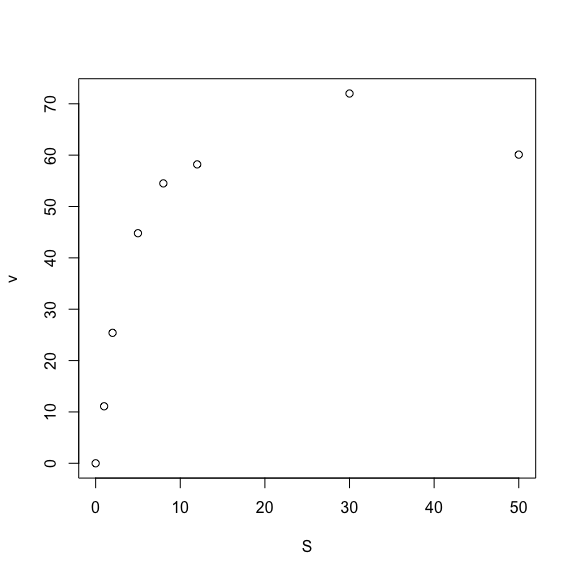
## Import and explore data

First we need to import the data within R and since there are only a few numbers we can first type the data into vectors. We can call S the substrate concentration and v the observed "velocity" result.

# Data can be entered as vectors S and v   
S <- c(0,1,2,5,8,12,30,50)  
v <- c(0,11.1,25.4,44.8,54.5,58.2,72.0,60.1)

We can then create a simple plot and see if the data "makes sense" in the plot:

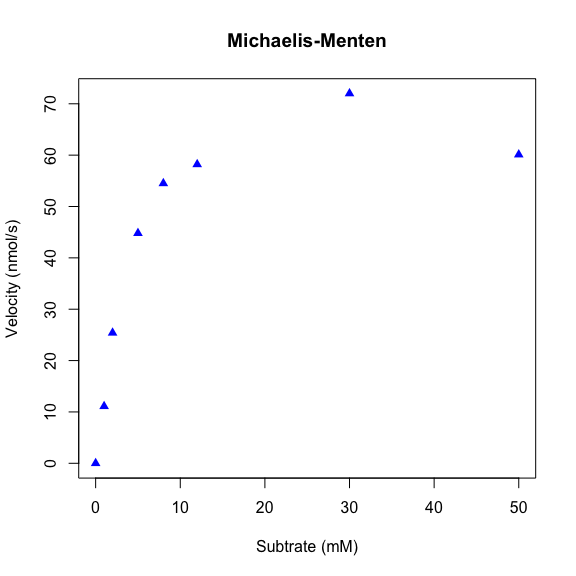
plot (S,v)



What do you think?

We can make the plot beter by adding a plot title, axis labels, change the default circles to filled triangles and make them blue:

plot (S,v,   
 xlab="Subtrate (mM)",   
 ylab="Velocity (nmol/s)",   
 main="Michaelis-Menten",   
 pch=17, col="blue")



Does it look better? You can change the color to red or another color if you wish to try.

Does it look OK?

We can make the data into a data frame for S and v so that we don't have to specify both vectors each time.

We can create the dataframe with the command:

kinData <- data.frame(S,v)

## Theoretical equation

It would be nice to plot a theoretical curve in the midst of the data, and also find a way to evaluate and (Vmax) in the process.

The theoretical curve can be defined into an R object to contain the Michaelis-Menten curve, for example we can call it MMcurve.

# "velocity = Vmax times S divided by (Km plus S)", stored in MMcurve  
MMcurve<-formula(v ~ Vmax\*S/(Km+S))

Briefly: The ~ operator [...] is interpreted as a specification that the response v is modelled by a linear predictor Vmax\*S/(Km+S).

For more details on the formula() function:

?formula

From our enzymology classes we need to remember that:

is the maximum value, the asymoptote of the curve.

The value of is equal to when is half-way ()

By observing the plot we can make a "guess" as to what the values should be approximately, and then use an R method to solve the equation based on this rough data.

By observing the plot we can deduct that:

at () *i.e.*

## Nonlinear least square

Nonlinear least square is a mathematical method that can help solve the Michaelis-Menten equation.

Any basic R installation contains the stats (statistics) package and the nls() function is a built-in Nonlinear least square solver.

Now that we have an estimate of the parameters to initialize the calculation, we can ask R to help compute an answer for the values of the constants and (see estimation above.)

bestfit <- nls(MMcurve, kinData, start=list(Vmax=50,Km=2))  
# print the results  
bestfit

Nonlinear regression model  
 model: v ~ Vmax \* S/(Km + S)  
 data: kinData  
 Vmax Km   
73.261 3.437   
 residual sum-of-squares: 156.4  
  
Number of iterations to convergence: 7   
Achieved convergence tolerance: 5.971e-06

Therefore we have established that:

= 73.26

= 3.44

**Important note**: We are learning here to make "dynamic" document and therefore ***all*** the numbers have to be calculated by R when the document is updated.

Therefore here is the "under the hood" code to write the results so that **no manual data entry** will be required!

First, we could see the results of bestfit by simply having it print to the screen and in the document.

Now we need to be able to "grab" the values presented for Vmax and Km and for now we'll dispense of the "pretty" nomenclature of and to simply concentrate on what's important.

These two parameters can be extracted from bestfit with the coef() function, a generic function to extract model coeficients from objects returned by modeling functions, in our case it was nls(). The function can be written as either coef() or coefficients(). Hence:

coef(bestfit)

Vmax Km   
73.261388 3.437164

We can use subsetting methods to "grab" them independently:

# Vmax is the first:  
coef(bestfit)[1]

Vmax   
73.26139

# We can also round and show only 2 significant digits:  
round(coef(bestfit)[1], digits = 2)

Vmax   
73.26

#Km is the second:  
coef(bestfit)[2]

Km   
3.437164

# rounded:  
round(coef(bestfit)[2], digits = 2)

Km   
3.44

To write the "narrative" and use these values within the text we make use of the fact that R can compute code within a code text if marked properly as we have seen in a previous section.

Since R can extract and round-off these numbers we can indeed include them in the text as such:

which is processed as:

**Vm = 73.26** and **Km = 3.44**

# Theoretical curve

We can use the bestfit calculation to now calculate and plot the theoretical curve.

First, we create a set of theoretical concentration values from 0 to 50 in step increments of 0.1 (intervals). We can then use these substrate concentrations to calculate ***predicted velocities*** that will make the theoretical curve line

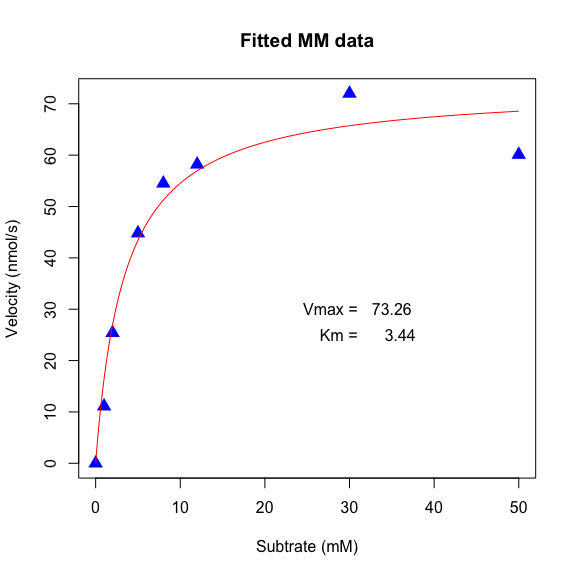
# Concentration ranges for S:  
SconcRange <- seq(0,50,0.1)

Now we can use the stats package predict() function, *"a generic function for predictions from the results of various model fitting functions."*

# Then, calculate the predicted velocities using the predict() function:  
theorLine <- predict(bestfit,list(S=SconcRange))

We can now redo the plot with all the data, theoretical line and constant values:

# Now plot the data, the best fit line, and put the best fit coefficients in the plot  
plot (kinData,   
 xlab="Subtrate (mM)",   
 ylab="Velocity (nmol/s)",   
 title(main="Fitted MM data"),   
 pch=17, col="blue", cex=1.5)  
  
# Now add the theoretical line with the lines() function:  
lines(SconcRange,theorLine,col="red")  
  
# Add text with the values:  
text(28,30, "Vmax = ")  
text(35,30,round(coef(bestfit)[1],2))  
text(29,25, "Km = ")  
text(36,25,round(coef(bestfit)[2],2))



# Final Freddie report

We now have all the elements to write a final report for Freddie...

When you press Knit HTML the HTML file is saved in one single file (containing all figures as well) and you can hand it to Freddie.

It is possible to export at PDF and MSWordin addition toHTML` but that might require the installation of at least one more package.

# R Session.

It is customary to add information about the R version and the packages that were loaded at the time of the calculation. This is accomplished with the simple command session() that will display all this information. This is extremely useful to record the R version etc.

sessionInfo()

R version 3.3.3 (2017-03-06)  
Platform: x86\_64-apple-darwin13.4.0 (64-bit)  
Running under: OS X El Capitan 10.11.6  
  
locale:  
[1] en\_US.UTF-8/en\_US.UTF-8/en\_US.UTF-8/C/en\_US.UTF-8/en\_US.UTF-8  
  
attached base packages:  
[1] stats graphics grDevices utils datasets methods base   
  
other attached packages:  
[1] knitr\_1.15.1  
  
loaded via a namespace (and not attached):  
 [1] backports\_1.0.5 magrittr\_1.5 rprojroot\_1.2 tools\_3.3.3   
 [5] htmltools\_0.3.5 yaml\_2.1.14 Rcpp\_0.12.10 stringi\_1.1.5   
 [9] rmarkdown\_1.4 highr\_0.6 stringr\_1.2.0 digest\_0.6.12   
[13] evaluate\_0.10

# Acknowledgments

The data, ploting method and nls calculation were from the script online at: <http://rforbiochemists.blogspot.com/2015/05/plotting-and-fitting-enzymology-data.html>

Other web examples for MM plots:

[Fitting a Michaelis-Menten model and drawing the results in R](https://rpubs.com/RomanL/6752)

[Fitting a Michaelis-Menten curve using R](http://davetang.org/muse/2013/05/17/fitting-a-michaelis-mentens-curve-using/)

# REFRENCES:

Michaelis, L., and M.L. Menten. 1913. “Die Kinetik Der Invertinwirkung.” *Biochemische Zeitschrift* 49: 333–69.

Michaelis, L., M. L. Menten, K. A. Johnson, R. S. Goody, L. Michaelis, and M. L. Menten. 2011. “The original Michaelis constant: translation of the 1913 Michaelis-Menten paper.” *Biochemistry* 50 (39): 8264–9.