

Enzyme Kinetics in R

Zane Billings

Background

What is Enzyme Kinetics?

Enzyme kinetics is a subfield of biochemistry which studies the dynamics of chemical reactions which are catalyzed by **enzymes**. Enzymes are a specialized class of protein molecule that have special properties allowing them to speed up biological reactions: this is called the catalytic function of an enzyme.

Enzyme kinetics experiments usually involve taking a known amount of an enzyme, adding it to several different concentrations of its **substrate** (the molecule that the enzyme can interact with), and using an analytical laboratory technique in order to analyze the velocity of the enzyme. Several methods exist for measuring enzyme velocities during a reaction, but the most common are spectrophotometry, radiometry, and fluorescence methods.

Michaelis-Menten Kinetics

The easiest type of enzymes to describe mathematically are the class of enzymes which are said to follow **Michaelis-Menten kinetics**. Leoner Michaelis and Maud Menten were two biochemists who gave the mathematical model for these enzymes in the beginning of the twentieth century.

In order for an enzyme to fit this model, we must make a few **assumptions**.

- All of the enzyme and substrate molecules can diffuse freely (i.e. move wherever they want without constraints) in their container.
- The enzyme only binds to one substrate. Enzymes which bind to multiple different substrates (often at the same time) become more complicated and the derivation of their kinematic equations is much more difficult.
- The rate of reaction of the enzyme is only affected by two things: intrinsic chemical properties of the enzyme, and the concentration of the substrate. The enzyme can work faster with more substrate, up until the “saturation point”, where all available enzymes are filled with substrate and the rate of the reaction is capped.
- The concentration of the enzyme is much lower than that of the substrate.

If we meet all of these conditions, we can describe the rate of the enzyme-substrate reaction with the **Michaelis-Menten Equation**.

$$V = \frac{V_{\max}[S]}{K_m + [S]}$$

In the equation, V is the reaction rate, V_{\max} is the maximum possible reaction rate, K_m is the Michaelis-Menten constant, a unique property of every enzyme which describes the concentration of the substrate required for the velocity to reach one half of V_{\max} , and $[S]$ is the concentration of the substrate.

The Lineweaver-Burk Plot

Lineweaver and Burk were two biochemists who studied enzyme kinetics and developed a method for using least-squares linear regression to estimate the values of K_m and V_{\max} from kinetics data. The LB-plot is also known as a **double-reciprocal plot**.

Suppose we are testing an enzyme that we will call Hoopsase (almost all enzyme names end with the suffix “-ase”). We have done a lot of spectrophotometry over the last few days, and we have collected the following data set.

[S], μM	Velocity μMs^{-1}
2	2.9
3	3.8
4	4.4
5	5.0
6	5.4
7	5.8
8	6.2
9	6.4
10	6.7

Table 1: Velocity data for various substrate concentrations of the enzyme Hoopsase.

You may use the following code snippet to create a data frame with this information in R. (Change “ex” to whatever you want the name to be, of course. I recommend changing the column names as it will be less confusing in the long run.)

```
1 ex <- data.frame(matrix(c(2,3,4,5,6,7,8,9,10,
2                          2.9,3.8,4.4,5.0,5.4,5.8,6.2,6.4,6.7),
3                          ncol = 2, byrow = FALSE))
4 colnames(ex) <- c("S", "V")
```

The Lineweaver-Burk plot is a graphical method where the reciprocal of the y-data is plotted against the reciprocal of the x-data. Mathematically:

$$\frac{1}{V} = \frac{K_m + [S]}{V_{\max}[S]} = \frac{K_m}{V_{\max}} \frac{1}{[S]} + \frac{1}{V_{\max}}.$$

This is an equation of the form $y = mx + b$, suggesting that if we compute the line of best fit (using a least-squares linear regression method), we can use the slope and intercept of the line to calculate K_m and V_{\max} of our enzyme, which gives us a lot of information about how fast the enzyme works.

The Lineweaver-Burk plot was designed originally to be easy to do by hand, but with the power of Rat our disposal, we can do a linear regression model in less than 5 minutes. Let’s go through this process step by step on the next page.

1. Import the data. The code for doing this is above, underneath Table 1. You may also want to use `plot` (or `qplot` from the *ggplot2* package) to take a quick look at the data.
2. Take the reciprocals of the data series. You can do this using the `transform` command if you like that method, but I prefer to declare a new data frame instead.

```
1 lb.ex <- data.frame(1/ex$S, 1/ex$V)
2 colnames(lb.ex) <- c("s.recip", "v.recip")
```

3. Generate the linear regression model using the `lm` command. You can view almost everything you want to know about the model using the command `summary`.

```
1 model <- lm(
2     formula = v.recip ~ s.recip,
3     data = lb.ex)
4 summary(model)
```

```
> summary(model)

Call:
lm(formula = v.recip ~ s.recip, data = lb.ex)

Residuals:
    Min       1Q   Median       3Q      Max
-0.0018856 -0.0013560 -0.0001700  0.0005518  0.0032940

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.102373   0.001224   83.64 9.19e-12 ***
s.recip      0.486422   0.004952   98.22 2.99e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.001829 on 7 degrees of freedom
Multiple R-squared:  0.9993,    Adjusted R-squared:  0.9992
F-statistic: 9648 on 1 and 7 DF,  p-value: 2.988e-12
```

Figure 1: The output from the `summary` command. Several relevant outputs are displayed including the parameter estimates and their standard errors, with t-statistics and p-values for each, along with some information about the residuals, R^2 value, and F-test information.

4. In order to judge the goodness-of-fit of the regression line, we can either look at statistics (such as the R^2 value printed above) we can take a qualitative look at a plot of the regression line.

```
1 plot(x = lb.ex$s.recip,
2     y = lb.ex$v.recipe,
3     main = "Lineweaver-Burke_Plot",
4     xlab = "1/[S]",
5     ylab = "1/V")
6 abline(model, lty=2)
```

Note that the `plot` command generates a scatterplot of the data, and the `abline` command can take our linear model object as an input and create a line based on the model (`lty=2` specifies that the line should be a dashed line).

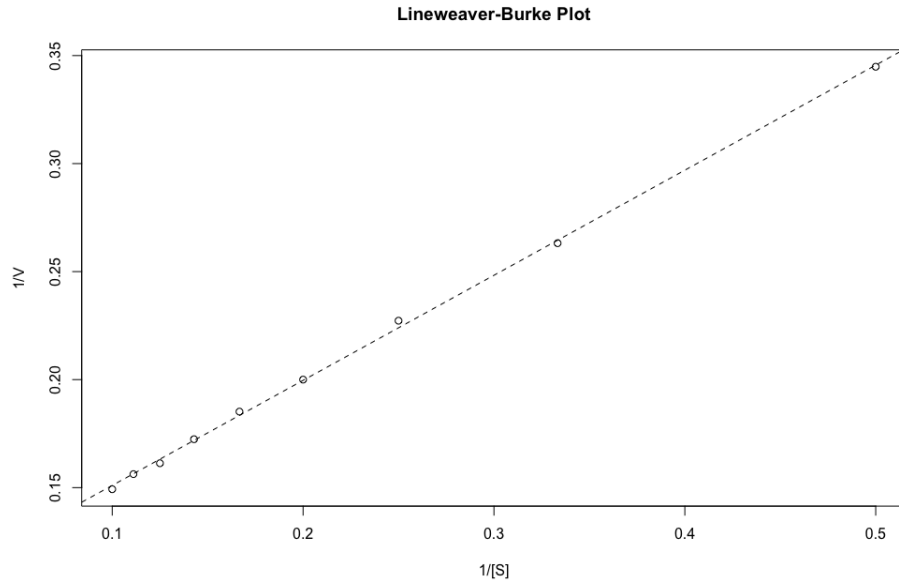


Figure 2: The Lineweaver-Burk plot for the above data.

- Now we can use the function `coef` to extract the slope and the y-intercept values from the linear model object. As we saw from the Lineweaver-Burk equation, we can extract the values of K_m and V_{\max} from the parameters of the model.

```
> cfs <- coef(model); cfs
(Intercept)      s.recip
    0.1023731    0.4864224
> vmax <- as.numeric(1/cfs[1])
> km <- as.numeric((1/cfs[1])*cfs[2])
> vmax;km
[1] 9.76819
[1] 4.751466
```

Figure 3: Notice that the `coef` function returns names with our values (the first value is the y-intercept, and the second value is the slope). Using `as.numeric()` around our values gets rid of the header part of the data and ensures that our parameter values are stored as numbers only. Look at the Lineweaver-Burk equation and make sure you can explain why we used these formulae in order to get the values of K_m and V_{\max} .

- Now we can plot the fitted model against our data. All we need to do is generate a vector of \hat{y} , or predicted, values, and plot it against our original x values. How do we generate the predicted values? Use the computed values of K_m and V_{\max} we just found and plug them into the Michaelis-Menten equation from Page 1.

```

1 vhat <- (vmax*ex$S)/(km+ex$S)
2 plot(ex, main = "Lineweaver-Burke Example") +
3   lines(x = ex$S, y = vhat, lty=2)

```

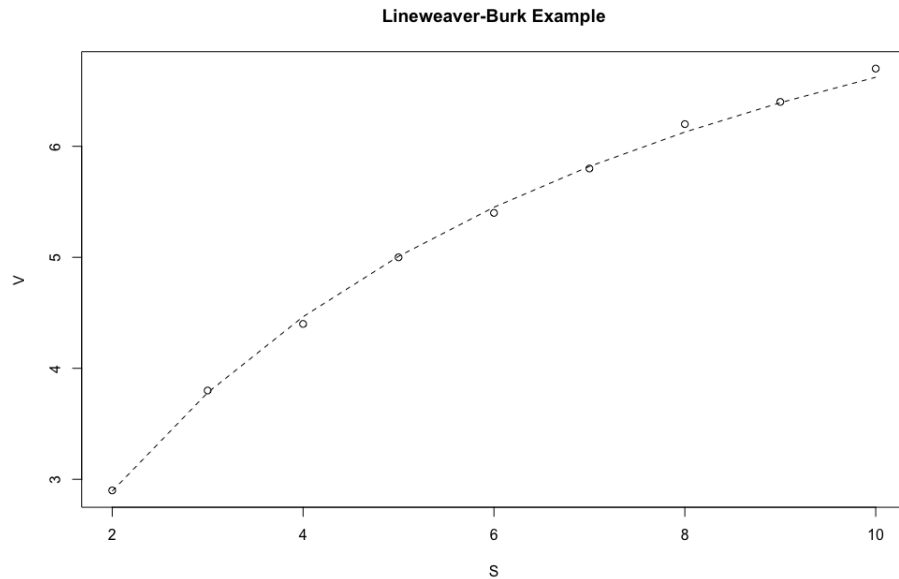


Figure 4: The Michaelis-Menten plot with our calculated parameter values: it seems to be (qualitatively) a fairly good estimate.

7. **You try it!** Make sure you can run and understand all of the code needed to do this example. All of the questions in this assignment will use very similar concepts so make sure you understand how this example works!

Analysis of More Realistic Data

Now that you've seen an example, let's look at some data that might be considered more "realistic". Enzyme kinetics data can be hard to collect, and experimental and instrumental error in data collection can lead to some weird data.

As you complete this assignment, create a TeX document where you include all of your code in a listing file and include all plots you generate in the file as well. Answer all of the questions below. For your final submission, you should submit a compiled .pdf document as well as a .R file containing your R script.

Now, take a cursory look at the kinetics dataset found on Blackboard. The columns of this dataset are: the substrate concentration [S], and three v columns...each column represents a new experiment, so we see the experiment was repeated three times.

1. Import the data file "kinetics.csv" which can be found on Blackboard. Rename the columns to make analysis easier. The first thing you need to do is take the average of all three velocity values. (Hint: You can use the following code:

```
kinetics <- transform(kinetics, vave = rowMeans(kinetics[,-1]))
```

Where "kinetics" is whatever you want to name your dataframe, and "vave" is what you want to name your new column.)

For the rest of this analysis, we will only be working with the averages. Make a quick plot of the average velocity vs. [S].

2. Create a Lineweaver-Burk plot for the data, and use `coef` to look at the slope and y-intercept. Noting that K_m and V_{\max} must be **strictly positive** to make any sense, what is wrong with this? Is this model accurate?
3. We can make a second plot called the Hanes-Woolf plot to solve this issue: both sides of the Lineweaver-Burk equation are multiplied by [s], which helps to correct for sampling error in the data. Use the following equation

$$\frac{[S]}{V} = \frac{1}{V_{\max}}[S] + \frac{K_m}{V_{\max}}$$

instead of the Lineweaver-Burk equation to create a plot and a linear regression model. Does this model (qualitatively) seem accurate? What are the K_m and V_{\max} values you get from this plot? (Use the above equations to figure out how to extract K_m and V_{\max} from the output of `coef`.)

Then, use your estimated parameters in the Michaelis-Menten equation to see how well your predicted values match your true values.

4. Now that you've seen how to use linearization methods to calculate these values, I will let you in on a secret: in the modern computer era, we don't really need (or want) to do any of these methods to estimate parameters. We can instead use an algorithm called **nonlinear least squares** regression. Use the following code:

```
nls.model <- nls(formula = vave ~ Vmax * concentration/(Km + concentration),
  data = kinetics,
  start = list(Km = max(kinetics$vave)/2,
    Vmax = max(kinetics$vave)))
```

with “kinetics” replaced by the name of your data frame, “concentration” replaced by the name of your first data frame column, and “vave” replaced by the name of the average velocity column you created in question 1.

Again, use `coefs` to extract the parameter values, and make a plot using the Michaelis-Menten equation. Does this model fit the data better than the Hanes-Woolf method?

5. What are some advantages and disadvantages of the linear models (Lineweaver-Burk and Hanes-Woolf)? What about the nonlinear least squares model? (Note that before about the the 90's, nonlinear least squares regression required inaccessible amounts of computing power.)

BONUS: The Eadie-Hofstee diagram is a linearization method similar to Lineweaver-Burk and Hanes-Woolf, but is not often used as it can magnify experimental error. However, if you have very clean data, the Eadie-Hofstee diagram has one major advantage. By simplifying the Eadie-Hofstee equation below, can you determine what this advantage is?

$$\frac{V_{\max}}{v} = V_{\max} \left(\frac{K_m + [s]}{V_{\max}[s]} \right) \quad (1)$$