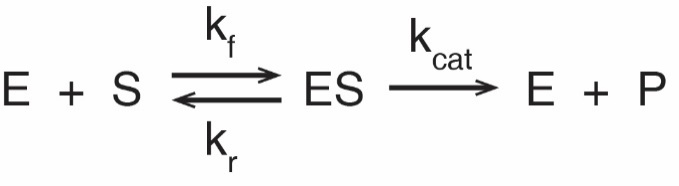
**Lab 4: The Michaelis-Menten Equation**

# What are we modeling and what assumptions do we make?

In biochemistry, *Michaelis-Menten kinetics* are commonly used as a model for enzyme behavior. Named after German biochemist Leonor Michaelis and Canadian physician Maud Menten, this model describes the dynamics of the reactions involved in the formation of a product, P, via the binding of a substrate, S, (the material with which an enzyme reacts) to an enzyme, E. An enzyme works in the following way:

1. The enzyme and the substrate are in the same solution. Sometimes an enzyme may catalyze a reaction that involves more than one substrate.
2. The enzyme binds to the substrate at a special area called the “active site.” This combination is called the enzyme/substrate complex. Enzymes are very specific in their choice of substrate molecules. The active site has a specific shape, and it also has specific chemical and charge properties, which influence the interaction with the substrate.
3. A chemical reaction occurs in the active site and the substrate is changed. The substrate could be broken down or could be fused to another molecule to make something new. In this process chemical bonds will be made or broken. At the end of these steps, the result is an enzyme/product complex.
4. The enzyme releases the product. When the enzyme lets go of the product, it returns to its original shape. It can then work on another substrate molecule. This process is called *catalysis*, because the enzyme causes a chemical reaction to occur, although it is not itself changed during this process. Usually, this release step is assumed to be very rapid.

A general enzyme-catalysis reaction can be expressed as



Here, *S* denotes the substrate, *E* the enzyme, *ES* the enzyme/substrate complex, and *P* the product. We measure concentration in units of number of molecules per unit volume (i.e., *M* or “molarity”), and we will use brackets to indicate when we are talking about a concentration. The *k* ’s are rate constants (as you may already know from chemistry). They tell you how fast the forward and reverse enzyme-substrate reactions are proceeding, as well as how quickly the product reaction is occurring.

# Our Mathematical Model

We can build a set of differential equations to describe the dynamics of these reactions. In the general case, there are four differential equations (for [*E*], [*S*], [*ES*], and [*P* ]).

The units of the left-hand side of each equation are *M/s* (i.e., concentration divided by time). What must the units be for kf so that the units match on the right side of the equation?

1/sec

In the first equation, why is the kf [E][S] term negative?

At equilibrium, kr[ES] must equal kf{E][S], but converting E + S to ES reduces the amounts of both E and S.

Often, biologists measure (and care about) the speed of the reaction (i.e., *d*[*P* ]*/dt* — how fast the product is being produced). Since *d*[*P* ]*/dt* is proportional to [*ES*], knowing the latter concentration gives us this rate. Observe that the equations for [*E*] and for [*ES*] do not involve [*P* ]. This means that they can be analyzed separately from the equation for [*P* ], as we will do in the next part.

In this lab you’re going to test the common approximation that *the Michaelis-Menten equation correctly predicts the rate of product formation (assuming that its assumptions are valid)*. The Michaelis-Menten equation, which we derived in class, is given by:

Where (Note that “*≡*” means “is defined as.”)

and

Keep in mind that in order to derive the Michaelis-Menten equation, we made the assumption that [ES] is in a pseudo-steady state, or that d[ES]/dt = 0. Additionally, [S] in the equation above refers to the free substrate, not total amount of substrate. However, we commonly make the approximation that we can use [STotal] in place of [S].

What would the value of Km approach if *kr >> kf* ?

If kr>>kf, then Km would approach infinity.

Based on your answer above, what would happen to the predicted rate of product formation rate if *kr >> kf* ?

Product formation rate will be very low if kr is >>kf, since there won’t be much ES for the catalyst to work on.

You’re going to test the Michaelis-Menten equation by simulating two experimental reactions, each specified by its initial concentrations of the reaction components:

Condition #1: [*S*] = 100 *µM,* [*E*] = 1 *µM,* [*ES*] = [*P* ] = 0 *µM*

Condition #2: [*S*] = 1 *µM,* [*E*] = 1 *µM,* [*ES*] = [*P* ] = 0 *µM*

1. **Using a script to explore the model in MATLAB**

Open MATLAB and open the provided m-file “Michaelis\_Menten”.

Edit the top block of the file to define each of your rate constants to be equal to 1 (with units of s-1 or µM-1s-1). For Km, enter the appropriate expression to compute Km from the rate constants. What is the value of Km?

Km = kr + kcat/kf, so if everything is equal to 1, then Km = (1+1)/1=2

What does it mean, in terms of the fate of ES complexes, to have *kcat* equal to *kr*?

There won’t be much ES complex around at any time, since it’s being depleted at equal rates from both ends at once.

For each of the two reaction initial conditions, edit the second block of the file to use the Michaelis-Menten equation to predict the rate of product formation. Enter the predicted rates below

Condition #1: 0.9804

Condition #2: 0.3333

1. ***in silico* Experiments with MATLAB**

“*in silico*” is a fancy term for experiments that are simulated (i.e., conducted on a *silicon* computer chip). It’s supposed to evoke more common biology terms such as *in vitro* (in a test tube) and *in vivo* (in a living organism). This term is showing up more and more often in biology research papers these days, because of the increasing popularity of useful and inexpensive computer simulations.

We will continue to use MATLAB’s built-in tool for solving a system of differential equations numerically called ode45.

In the third block of the file, we define our differential equations. First, we define a time span (tspan) over which we will have MATLAB compute the solutions to the equations. As in our models of disease, the interpretation of the units for time is up to us. We can consider time to be in units of seconds (but we could interpret them as minutes as well, so long as we use a consistent system for all our rate constants and graphs).

On the next lines, we define our differential equations and store them in dy:

dy = @(t,y) [-kf\*y(1)\*y(2) + kr\*y(3);

-kf\*y(1)\*y(2) + kcat\*y(3) + kr\*y(3);

kf\*y(1)\*y(2) - kcat\*y(3) - kr\*y(3)]

Remember that dy is being defined as a “function handle”. This is a tool in MATLAB that allows us to write equations for functions that depend on dependent variables. MATLAB can then interpret these functions as differential equations when we use ode45. The @ character tells MATLAB that we are defining a function handle. The (t,y) tells MATLAB that the function depends on two variables called t and y.

The square brackets define the output of our function as a (column) vector. Each row corresponds to one output. If you look carefully at the first row “-kf\*y(1)\*y(2) + kr\*y(3)”, you should see how it corresponds to our differential equation for d[S]/dt. Similarly, the second row is our differential equation for d[E]/dt, and the third row is our differential equation for d[ES]/dt. Notice that we have replaced [S] with y(1) and [E] with y(2). What does y(3) represent?

ES

If we were to add a fourth row for d[P]/dt, what would it be?

dP/dt = v = kcat[ES], so it would be identical to v\_predicted: kcat\*[(E\*S)/(S + Km)]

Execute the third block to define our differential equations. Now, we will use the fourth block to define our initial conditions and solve the differential equations. Edit the first line of the block to define the initial conditions for Condition #1. The initial conditions are stored as a vector y0, where each element of the vector corresponds to the initial state of one of our three variables ([S], [E], and [ES]). As before, it is essential that the order in y0 matches the order we had chosen for y and dy above. Paste your code for defining y0 below:

y0 = [100 1 0]; % [S E ES]

The ode45 function will return time values in t and timecourses of concentrations of our components in y. The first column of y will be the timecourse for S. The second column will be E, and the third column will be ES.

Remember, your goal with this simulation is to check whether the Michaelis-Menten equation makes a good prediction for the reaction rate *v*. Execute the fifth block to plot the results for the reaction rate as a function of time. Approximately, what is the reaction rate at *t* = 10?

0.978

Does this value match the prediction you made for the rate of change of the product concentration? Why or why not was the approximation accurate?

It’s reasonably close – prediction was 0.9804. Accuracy based on there being a lot more S than E.

Now it’s time to run a new simulation for the second set of initial conditions. Adjust the code to use Condition #2. Which blocks do you need to run again to do this second simulation?

Change E and S in Block 2, also y0 in Block 4

When you’re ready, run this new simulation. As before, approximate the reaction rate at *t* = 10. Does your simulation result match this prediction as well as it did before?

Reaction rate at t=10 is 0.01289, much less than the 0.3333 we predicted. When S is equal to E rather than being much higher, the predictive value of the model crashes.

Making models (in biology, or in finance or in public health) is all about *understanding the assumptions that the models rely on and evaluating whether a particular situation fits those assumptions*. Does this second case fit the assumptions that we made when we defined this model? Why or why not?

No, because we assumed initially that S was orders of magnitude greater than E and it isn’t in Condition 2.

The Michaelis-Menten equation is intended to give estimates of the reaction rate across a wide range of substrate concentrations, including those where [S] < Km. Try adjusting the initial enzyme concentration, while keeping [S] at time zero equal to 1. Under what conditions does the Michaelis-Menten rate estimate become accurate?

E = .1, S = 1 v\_pred = . 0.0333 vt10 = 0. 02577

E = .01 S = 1 v\_pred = .0033 v t10 = .003256

E = .001 S = 1 v\_pred = .00033 v t10 = .000332 <- When E is <= S/1000, the model works almost perfectly.

Now, set the initial conditions to [S] = 1, [E] = 1, and [ES] = 0. Keep *kf* =1 and *kr* =1. Try adjusting *kcat* to other nonzero values. What blocks of the model will you need to run again to simulate the model with the new rate constant?

Blocks 1-5

Under what conditions is the observed rate closer to the Michaelis-Menten prediction? Can you make the rate match the prediction? Why not?

When kcat = .1, v\_pred = 0.0476 and t10 = 0.0278

.01 0.005 0.0037

.001 0.000499 0.00038

Can’t get the v pred and t10 equal because S and Stotal aren’t equal.

Now try adjusting *kf*, *kr*, and *kcat*. Can you make the rate (nearly) match the Michaelis-Menten prediction? What did you need to do get it really close?

If kr is significantly larger than kf and kcat is small (.001), it nearly matches.

Make an extra block at the bottom of the script to plot [S] over time. Re-run the simulations with the rate constants you used for the previous two questions. Do the plots of [S] versus time give you insight into when and why the model failed?

The model failed when kr, kf, and kcat were within an order of magnitude of each other, since the model is based on the assumption that they are not.