Systems Biology, Homework # 3, Part 2: Biochemical Kinetics

Grayson Gerlich 10801535

1. **Reversible Michaelis-Menten kinetics**
Recall the derivation of the Michaelis-Menten rate law for an enzyme-catalysed reaction, which began with the scheme:

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} C \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} E + P. \tag{1}$$

(a) Differential equation model for system dynamics (as well as conservation statement).

$$\frac{ds}{dt} = k_{-1}c - k_1(s * e)$$
$$\frac{de}{dt} = (k_{-1} + k_2)c - k_1(se) - k_2(ep)$$
$$\frac{dc}{dt} = k_1(se) - k_{-2}(ep) - (k_{-1} + k_2)c$$
$$\frac{dp}{dt} = k_2 c - k_{-2}(e * p)$$
$$e_T = e(t) + c(t)$$

(b) Quasi-steady state assumption application.

$$0 = k_1(se) - k_{-2}(ep) - (k_{-1} + k_2)c_{qss}$$
$$0 = k_1 * (s(e_T - c_{qss})) - k_{-2}((e_T - c_{qss})p) - (k_{-1} + k_2)c_{qss}$$
$$0 = k_1 s e_T + k_{-2} e_T p - k_1 s c_{qss} - k_{-2} c_{qss} p - k_{-1} c_{qss} - k_2 c_{qss}$$
$$0 = k_1 s e_T + k_{-2} e_T p - c_{qss}(-k_1 s - k_{-2} p - k_{-1} - k_2)$$
$$c_{qss} = \frac{k_1 e_T s + k_{-2} e_T p}{k_1 s + k_{-2} p + k_{-1} + k_2}$$

(c) Overall reaction rate.

$$V = \frac{dp}{dt} = k_2 c_{qss} - k_{-2}((e_T - c_{qss}) * p)$$
$$V = \frac{dp}{dt} = k_2 \frac{k_1 e_T s + k_{-2} e_T p}{k_1 s + k_{-2} p + k_{-1} + k_2} - k_{-2}((e_T - \frac{k_1 e_T s + k_{-2} e_T p}{k_1 s + k_{-2} p + k_{-1} + k_2}) * p)$$
$$V = \frac{dp}{dt} = \frac{k_1 k_2 e_T s - k_{-1} k_{-2} e_T p}{k_1 s + k_{-2} p + k_{-1} + k_2}$$

(d) Take k_{-2} to 0 to recover the irreversible Michaelis-Menten rate law is recovered.

$$\frac{dp}{dt} = k_2 c_{qss}$$

$$\frac{dp}{dt} = k_2 \frac{k_1 e_T s + k_{-2} e_T p}{k_1 s + k_{-2} p + k_{-1} + k_2}$$

$$\frac{dp}{dt} = \frac{k_2 k_1 e_T s}{k_1 s + k_{-1} + k_2}$$

$$\frac{dp}{dt} = \frac{k_2 e_T s}{K_M + s} = \frac{V_{max} s}{K_M + s}$$

2. **First-order approximation of Michaelis-Menten kinetics**
Consider the four-step reaction chain shown below.

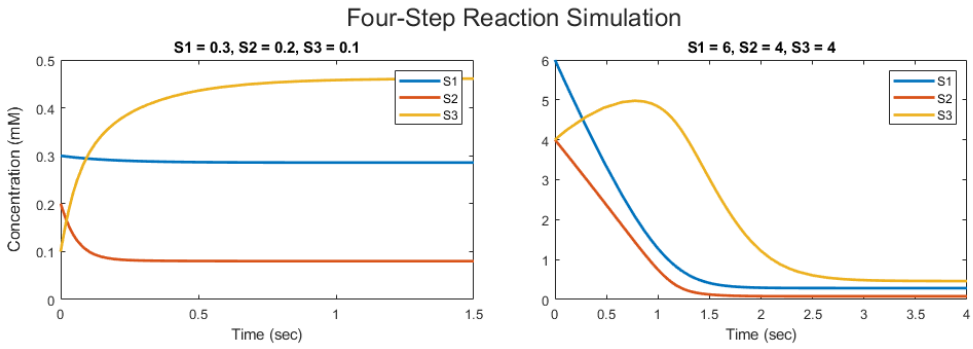
$$\xrightarrow{v_0} S_1 \xrightarrow{v_1} S_2 \xrightarrow{v_2} S_3 \xrightarrow{v_3} \tag{2}$$

Suppose the rate v_0 is fixed and presume the other rates are given by Michaelis-Menten kinetics, with

$$v_i = \frac{V^i_{\max} s_i}{K_{Mi} + s_i}. \tag{3}$$

Take parameter values $v_0 = 2$ mM/min, $V^1_{\max} = 9$ mM/min, $V^2_{\max} = 12$ mM/min, $V^3_{\max} = 15$ mM/min, $K_{M1} = 1$ mM, $K_{M2} = 0.4$ mM, $K_{M3} = 3$ mM.

(a) System simulation:

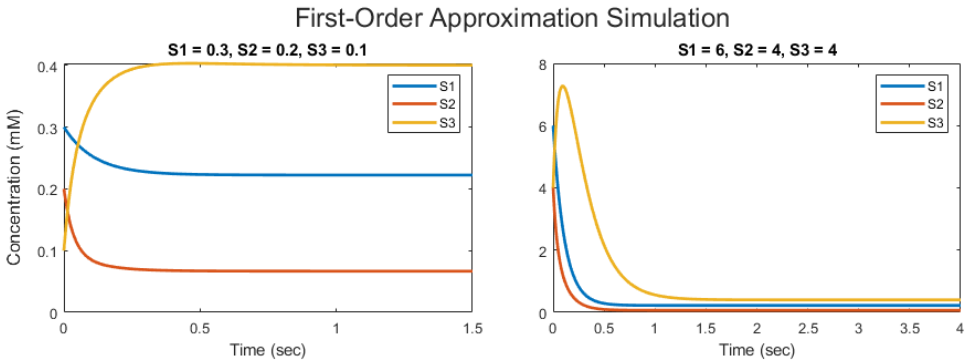


(b) First-order approximation model. Linearization done by expanding the Taylor series about $s_i = 0$ for each rate and taking the first order term. For example, for $v_1 = \frac{9s_1}{1+s_1}$ expands to $v_1 = 9s_1$ as a first order approximation.

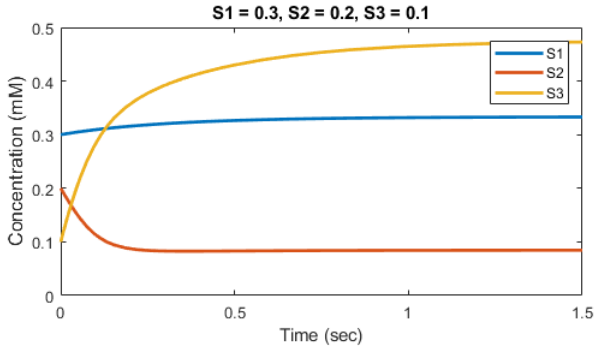
$$\frac{ds_1}{dt} = v_0 - k_1 s_1 = 2 - 9s_1$$

$$\frac{ds_2}{dt} = k_1 s_1 - k_2 s_2 = 9s_1 - 30s_2$$
$$\frac{ds_3}{dt} = k_2 s_2 - k_3 s_3 = 30s_2 - 5s_3$$

(c) System simulation:



The behavior of the first-order approximation is consistent with what I would expect. In this approximation, by centering around $s_i = 0$, we lose the ability to accurately characterize the behavior of the system when the concentration of the species is high. For this reason, the model with the first order approximation better characterizes the behavior of the system when the starting concentrations of the 3 species are quite small. In the second set of initial conditions, the behavior of the system is changed considerably. For my own amusement, I redefined my model using a second-order approximation, and recovered system dynamics for the first set of initial conditions quite nicely.



(d) Model and Driver Code for First Simulation:

```
function dYdt = q2_model(t,Y)
    % Extract S1, S2, S3 from input vector Y, calling them a,b,c for
    % ease of use
    a = Y(1); %a
    b = Y(2); %b
    c = Y(3); %c

    % Define rates v0-v4
    v0 = 2; %mM/min
```

```
    v1 = (9*a)/(1+a); % Michaelis-Menten rates
    v2 = (12*b)/(0.4+b);
    v3 = (15*c)/(3+c);

    % Define dadt, dbdt, dcdt from the ODEs
    dadt = v0 - v1;
    dbdt = v1 - v2;
    dcdt = v2 - v3;


    % Create output column vector dYdt
    dYdt = [dadt; dbdt; dcdt];
end
```

```
clear all

% Define the timespan to simulation
tRange = [0 1.5];
tRange1 = [0 4];

% Dfine the initial conditions
Y0 = [0.3,0.2,0.1];
Y1 = [6,4,4];

% Call the solver of choice
[tSol,YSol] = ode15s(@q2_model,tRange,Y0);
[tSol1,YSol1] = ode15s(@q2_model,tRange1,Y1);

% Plot solutions

t = tiledlayout(1,2);

t1 = nexttile;
plot(tSol,YSol(:,1),'LineWidth',2)
hold on
plot(tSol,YSol(:,2),'LineWidth',2)
plot(tSol,YSol(:,3),'LineWidth',2)
legend('S1','S2','S3','Location','northeast')
xlabel('Time (sec)')
title(t1, 'S1 = 0.3, S2 = 0.2, S3 = 0.1')
hold off

t2 = nexttile;
plot(tSol1,YSol1(:,1),'LineWidth',2)
hold on
plot(tSol1,YSol1(:,2),'LineWidth',2)
plot(tSol1,YSol1(:,3),'LineWidth',2)
legend('S1','S2','S3','Location','northeast')
xlabel('Time (sec)')
title(t2, 'S1 = 6, S2 = 4, S3 = 4')
hold off

title(t, 'Four-Step Reaction Simulation','FontSize',18);
ylabel(t,'Concentration (mM)', 'FontSize',12);
t.TileSpacing = 'compact';
```

(e) Model and Driver Code for First-Order Approximation

```
function dYdt = newq2_model(t,Y)
    % Extract S1, S2, S3 from input vector Y, calling them a,b,c for
    % ease of use
    a = Y(1); %a
```

```
    b = Y(2); %b
    c = Y(3); %c

    % Define mass action laws k0-k4
    k0 = 2; %mM/min
    k1 = 9; % Michaelis-Menten rates
    k2 = 30;
    k3 = 5;

    % Define dadt, dbdt, dcdt from the ODEs
    dadt = k0 - k1*a;
    dbdt = k1*a - k2*b;
    dcdt = k2*b - k3*c;


    % Create output column vector dYdt
    dYdt = [dadt; dbdt; dcdt];
end
```

```
clear all

% Define the timespan to simulation
tRange = [0 1.5];
tRange1 = [0 4];

% Define the initial conditions
Y0 = [0.3,0.2,0.1];
Y1 = [6,4,4];

% Call the solver of choice
[tSol,YSol] = ode15s(@newq2_model,tRange,Y0);
[tSol1,YSol1] = ode15s(@newq2_model,tRange1,Y1);

% Plot solutions

t = tiledlayout(1,2);

t1 = nexttile;
plot(tSol,YSol(:,1),'LineWidth',2)
hold on
plot(tSol,YSol(:,2),'LineWidth',2)
plot(tSol,YSol(:,3),'LineWidth',2)
legend('S1','S2','S3','Location','northeast')
xlabel('Time (sec)')
title(t1, 'S1 = 0.3, S2 = 0.2, S3 = 0.1')
hold off

t2 = nexttile;
plot(tSol1,YSol1(:,1),'LineWidth',2)
hold on
plot(tSol1,YSol1(:,2),'LineWidth',2)
plot(tSol1,YSol1(:,3),'LineWidth',2)
legend('S1','S2','S3','Location','northeast')
xlabel('Time (sec)')
title(t2, 'S1 = 6, S2 = 4, S3 = 4')
hold off

title(t, 'First-Order Approximation Simulation','FontSize',18);
ylabel(t,'Concentration (mM)', 'FontSize',12);
t.TileSpacing = 'compact';
```

3. Allosteric activation

Consider an allosteric *activation* scheme in which an allosteric activator must be bound before an enzyme can bind substrate:

$$R + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ER \tag{4}$$

$$ER + S \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} ERS \xrightarrow{k_3} P + ER, \tag{5}$$

(a) Quasi-steady-state assumption for complexes ER and ERS. First, differential equations and conservation statements.

$$\frac{dr}{dt} = k_{-1}er - k_1(r * e)$$

$$\frac{der}{dt} = k_1(r * e) + (k_{-2} + k_3)ers - k_1er - k_2(er * s)$$

$$\frac{ders}{dt} = k_2(er * s) - (k_{-2} + k_3)ers$$

$$\frac{ds}{dt} = k_{-2}ers - k_2(er * s)$$

$$\frac{dp}{dt} = k_3ers$$

$$e_T = e(t) + er(t) + ers(t)$$

Then, QSS approximation for ER:

$$0 = k_1(r * (e_T - er_{qss} - ers)) + (k_{-2} + k_3)ers - k_1er_{qss} - k_2(er * s)$$

$$0 = er_{qss}(-k_1r - k_{-1} - k_2s) + ers(-k_1r + k_{-2} + k_3) + k_1re_T$$

$$er_{qss} = \frac{ers(-K_1r + k_{-2} + k_3) + k_1re_T}{k_1r + k_{-1} + k_2s}$$

Then, QSS approximation for ERS:

$$0 = k_2(er_{qss} * s) - (k_{-2} + k_3)ers_{qss}$$

$$0 = k_2s(\frac{ers(-K_1r + k_{-2} + k_3) + k_1re_T}{k_1r + k_{-1} + k_2s}) - (k_{-2} + k_3)ers_{qss}$$

$$ers_{qss} = \frac{e_t * s * r}{r\frac{k_{-2}+k_3}{k_2} + \frac{k_{-1}(k_{-2}+k_3)}{k_1k_2} + sr}$$

Finally, find the rate equation:

$$V = \frac{dp}{dt} = k_3ers_{qss}$$

$$V = \frac{srk_3e_T}{r\frac{k_{-2}+k_3}{k_2} + \frac{k_{-1}(k_{-2}+k_3)}{k_1k_2} + sr} = \frac{V_{max}sr}{K_1r + K_2 + rs}$$

(b) Next, consider the case in which catalysis can only occur after n regulator molecules have bound.

The math is the same as above, except, because of the law of mass action, $nR+E$ generates a rate $k_1(r^n e)$ rather than $k_1(re)$. Propagating this change through the above math gives the reaction velocity as:

$$V = \frac{dp}{dt} = \frac{V_{max} s r^n}{K_1 r^n + K_2 + r^n s}$$

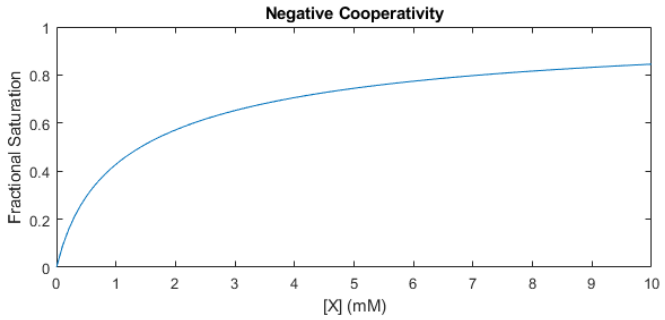
(c) Low substrate and regulator concentration:

If r^n and s are small, then K_2 dominates in the denominator, leaving the rate as:

$$v = \frac{V_{max} s r^n}{K_2}$$

4. **Negative cooperativity.**
Some proteins, such as the enzyme glyceraldehyde-3-phosphate dehydrogenase, exhibit *negative cooperativity*—substrate affinity drops as substrates bind.

(a) Negative cooperative case using $K_2 = 2, K_1 = 1$

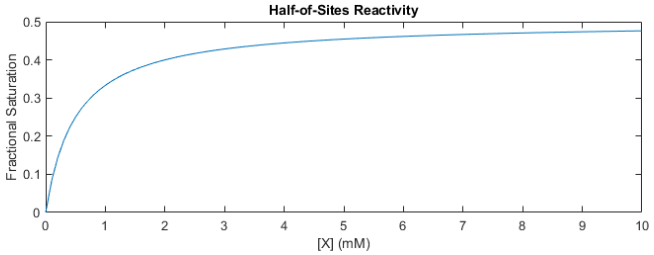


The curve is not sigmoidal, it is hyperbolic in much the same way that non-cooperative binding is.

(b) Half-of-the-sites reactivity is an extreme case in which there is 0 binding affinity once half of the sites have been filled. We can approximate this two ways. With the Adair equation, we can approximate it by saying $K_2 >> K_1$, or K_2 is approaching infinity. This reduces the Adair equation to:

$$Y = \frac{[X]/K_1}{1 + 2[X]/K_1}$$

This is also hyperbolic, although it approaches an asymptote at $Y = 0.5$, which makes sense, as half-of-the-sites reactivity should never exceed 50% fractional saturation.



(c) Code for First and Second Plots

```
x = linspace(0,10);
y = (x+x.^2/2)./(1+2*x+x.^2/2); % k1 and k2 are included as 1, 2 respectively
plot(x,y)
xlabel('[X] (mM)');
ylabel('Fractional Saturation');
title('Negative Cooperativity');
```

```
x = linspace(0,10);
y = (x)./(1+2*x); % k1 = 1
plot(x,y)
xlabel('[X] (mM)');
ylabel('Fractional Saturation');
title('Half-of-Sites Reactivity');
```