

20.320 Problem Set #5

Due on October 28, 2011 at 11:59am. No extensions will be granted.

General Instructions:

1. You are expected to state all of your assumptions, and provide step-by-step solutions to the numerical problems. Unless indicated otherwise, the computational problems may be solved using Python/MATLAB or hand-solved showing all calculations. Both the results of any calculations must be printed and attached to the solutions, and the corresponding code should be submitted on Course website. For ease of grading (and in order to receive partial credit), your code must be well organized and thoroughly commented with meaningful variable names.
2. You will need to submit the solutions to each problem to a separate mail box, so please prepare your answers appropriately. Staple the pages for each question separately and make sure your name appears on each set of pages. (The problems will be sent to different graders, which should allow us to get graded problem sets back to you more quickly).
3. Submit your completed problem set to the marked box mounted on the wall of the fourth floor hallway between buildings 8 and 16. MATLAB code when relevant should be submitted on Course website.
4. The problem sets are due at noon on Friday the week after they were issued. There will be no extensions of deadlines for any problem sets in 20.320. Late submissions will not be accepted.
5. Please review the information about acceptable forms of collaboration, which is available on the course website and follow the guidelines carefully. Especially review the guidelines for collaboration on code. NO sharing of code is permitted.

Problem 1: Pseudo first-order approximation

In class we saw the pseudo-first order approximation:

$$Y = \frac{[L]}{[L] + K_d} \approx \frac{[L_0]}{[L_0] + K_d}$$

where Y is the fraction bound, [L] is the concentration of unbound ligand, and [L₀] is the total amount of ligand added.

a) Under what initial protein and ligand concentrations (an inequality will suffice) can we be guaranteed we can approximate [L] with [L₀]?

We are guaranteed [L] will be approximately [L₀] when [L₀] >> [P₀] (even for arbitrarily small K_d).

b) Derive the exact formula for the amount bound. Hint: start with some amount of protein and some amount of ligand, and solve for the equilibrium value.

Start with [P] = P₀ and [L] = L₀, and [C] = 0. If the net number of forward reactions (per unit volume) is x, then our equilibrium will be at

$$K_d = \frac{[P_{eq}][L_{eq}]}{[C_{eq}]} = \frac{(P_0 - x)(L_0 - x)}{x}$$

Rearranging yields:

$$x^2 - (P_0 + L_0 + K_d)x + P_0L_0 = 0$$

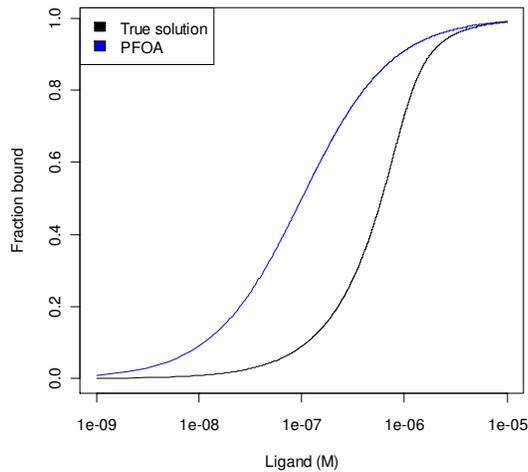
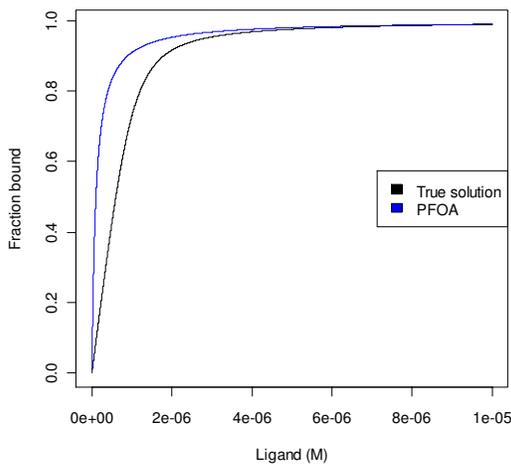
Solution is:

$$x = \frac{(P_0 + L_0 + K_d) \pm \sqrt{(P_0 + L_0 + K_d)^2 - 4P_0L_0}}{2}$$

A little bit of thought concerning the boundaries on x suggest that the correct root (so as to avoid negative concentrations) will be the one that subtracts the square root, not adds it. Since [C] = x, and we want the fraction bound [C]/P₀:

$$Y = \frac{(P_0 + L_0 + K_d) - \sqrt{(P_0 + L_0 + K_d)^2 - 4P_0L_0}}{2P_0}$$

c) Assume K_d is 100nM and the total amount of protein is 1uM. Plot the concentration bound with [L₀] varying from 1nM to 10uM, for both the pseudo first order approximation and the exact solution (both of these should be shown on the same plot). Make a second plot, which should be exactly the same as the first, but with the x axis on a log scale.

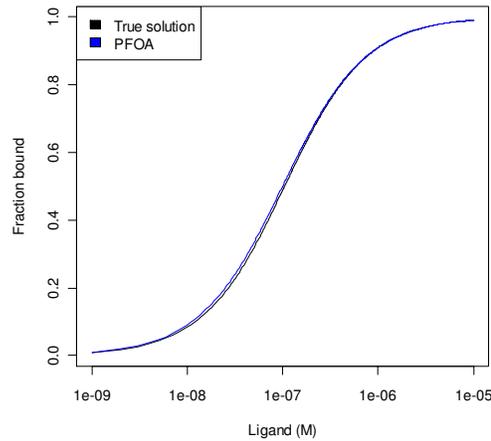


```

fracBound = @(p, l, kd) (p+l+kd - sqrt((p+l+kd).^2-4*p.*l))./(2*p)
PFOA = @(l, kd) l./(l+kd)
ligand = 1e-9:1e-9:1e-5
semilogx(ligand, fracBound(1e-6, ligand, 1e-7)) % or use plot
hold on
plot(ligand, PFOA(ligand, 1e-7), 'g')
legend('True solution', 'PFOA')

```

d) Make the same log plot for 10nM protein, same K_d . How does the discrepancy between the pseudo first-order approximation and the true solution change, as opposed to with 1uM protein? Explain this result.



The discrepancy between the PFOA and the true solution is much smaller because now the concentration of ligand is much higher with respect to the protein concentration (this was the assumption necessary for the PFOA in the first place).

Problem 2: ITC

a) In class it was stated without justification that the heat evolved after injection i was equal to the following:

$$Q_i = \Delta H \times V_{cell} \times ([C_i] - [C_{i-1}])$$

Explain what each part in this formula is and why this makes sense.

ΔH is the enthalpy per mol bound, and $C_i - C_{i-1}$ is the change in concentration of complex. $V_{cell}(C_i - C_{i-1})$ is thus the number of new binding reactions that occurred upon injection. Multiplying this times the binding enthalpy (energy/mol) gives us the total energy released.

b) You ran an ITC experiment and collected the data in ITC1.csv. The data look like

Injection	Heat evolved (cal)
1	2.99e-6
2	2.98e-6
...	...

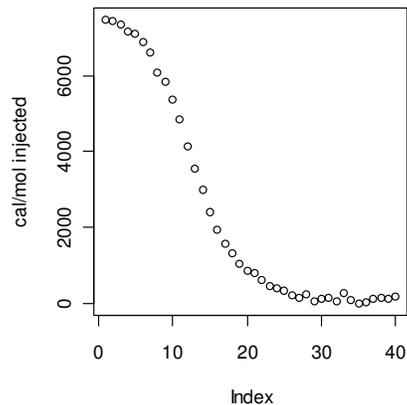
The conditions of your experiment were as follows:

Temperature: 298 K
 Initial protein concentration (P_0): 5 μM
 Cell volume (V_{cell}): 1 mL
 Ligand concentration increase per injection (L^+): 0.4 μM^*
 Number of injections: 40

*In this case, we will make the assumption that the addition of ligand does not change the total volume, and hence each injection increases the ligand concentration by $L^+ = 0.4\mu\text{M}$ without diluting any other components. You can think of this as we are injecting pure ligand, rather than solution containing ligand.

Note that L^+ is not the L_0 that was defined in lecture.

For each injection, calculate the heat released per mole of new ligand injected. Make a plot of this quantity versus injection number. Does this suggest any particular bound on the enthalpy of binding?



The first injection yielded 7472 cal/mol injected. If 100% of the injected ligand had bound, then this would be the actual enthalpy of binding. If less than 100% bound, then the energy released had to come from less than 4×10^{-9} mol binding events, and so the heat released per binding event must have been more than 7472 cal/mol. Thus 7472 cal/mol is a lower bound on the heat released by the binding reaction (ΔH_{bind} is less than or equal to -7472 cal/mol).

c) Now we are going to try to estimate ΔH and K_d . First, write down the formula for Q_i in terms of the variables i , P_0 , L_0 , K_d , ΔH , and V_{cell} . The concentration of complex should not appear explicitly in your formula. (Note, this formula won't be very pretty).

$$Q_i = \Delta H \times V_{\text{cell}} \times \left(\frac{L^+ - \sqrt{(P_0 + iL^+ + K_d)^2 - 4iP_0L^+} + \sqrt{(P_0 + (i-1)L^+ + K_d)^2 - 4(i-1)P_0L^+}}{2} \right)$$

d) Now fit this equation to the given data. What are K_d and ΔH ? Plot the data and overlay your fit.

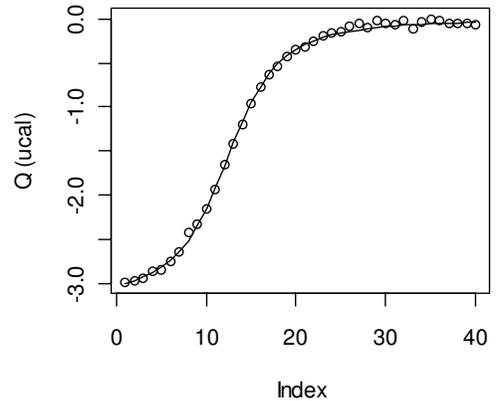
$\Delta H = -7983$, $K_d = 2.988e-07$

```
function [fit_k] = ITC_plot()
data = csvread('ITC1.csv');
injection = data(:,1);
heat = data(:,2);
heat_molinject = -heat./(0.4e-9);
figure(1)
plot(injection, heat_molinject)

k = [-10000, 1e-7]; %delH and Kd guesses
fit_k = nlinfit(injection, heat, @Q_fit, k)

Q_fitted = Q_fit(fit_k, 1:40);
figure(2)
plot(injection, heat, 'r.', injection, Q_fitted, 'b--')
legend('data', 'fit')
end

function y = Q_fit(k, x)
y = k(1)*1e-3*(0.4e-6-sqrt((5e-6+x*0.4e-6+k(2)).^2-4*x*5e-6*0.4e-6)+ ...
sqrt((5e-6+(x-1)*0.4e-6+k(2)).^2-4*(x-1)*5e-6*0.4e-6))/2;
end
```



e) Given this data, what is the entropy of binding?

$K_d = \exp(\Delta G/RT)$, so $\Delta G = -8884.4$ cal / mol.

$$-8884.4 = -7983 - 298 \Delta S$$

$$\Delta S = 3.02 \text{ cal}/(\text{mol K})$$

This originally had an error that the temperature was a 298 C (not K), so if they used that,

$$\Delta S = 1.578 \text{ cal}/(\text{mol K})$$

Problem 3: Estimating solutions to ODEs

So far, we've been dealing with systems at equilibrium, and haven't looked at how systems get to equilibrium. Let's first consider our typical system $P + L \rightleftharpoons C$.

a) What are the units of the rate constants k_{on} , k_{off} in this system?

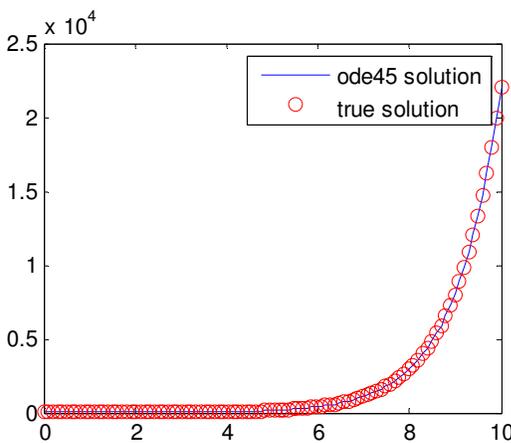
k_{on} has units of $(M s)^{-1}$

k_{off} has units of s^{-1}

b) We'll use MATLAB's ODE solver `ode45` (`ODEfunction`, `timespan`, `initialconditions`) for this problem. `ode45` is a Runge-Kutta based solver (if you don't recall this, it may be worth reviewing - see Euler's method, improved Euler's method, finally higher-order methods like the commonly used 4th-order Runge-Kutta method), which integrates over finite small timesteps (variable timesteps in the case of `ode45`).

"

Use ode45 to solve the differential equation: $dy/dt = y$ from time 0 to 10, given $y(0) = 1$. Plot this solution, and overlay a curve indicating the true solution.



```
[t,y] = ode45(@(t,y)y, linspace(0,10), [1]);
plot(t, y); hold on;
plot(linspace(0,10), exp(linspace(0,10)), 'ro')
legend('ode45 solution', 'true solution')
```

c) Now we'll simulate the interaction of a protein and a ligand to form a complex ($P + L \leftrightarrow C$). Say $k_{on} = 1e8$ and $k_{off} = 100$ (units omitted, but you should know them!) First, write down the ODEs governing the behavior of the system (three equations, because we have 3 chemical species). Is this system linear or nonlinear?

nonlinear.

$$\begin{aligned} d[P]/dt &= -k_{on} [P][L] + k_{off} [C] \\ d[L]/dt &= -k_{on} [P][L] + k_{off} [C] \\ d[C]/dt &= k_{on} [P][L] - k_{off} [C] \end{aligned}$$

d) For numerical stability, it will be preferable to reparameterize this system in terms of micromolar concentrations instead of molar concentrations. (You are welcome to experiment with solving without reparameterizing, but the solutions are likely to not be numerically stable). What are the resulting k_{on} and k_{off} values after reparameterizing? Include units!

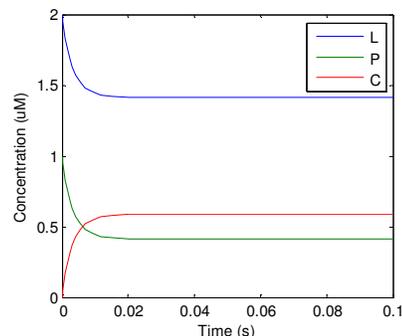
$$k_{on} = 100 / (\mu\text{M s})$$

$$k_{off} = 100 / \text{s}$$

e) Start with 2 μM ligand, 1 μM protein, and no complex. Simulate this system over 100 ms. Show a plot with the trajectories for each species over this time interval. Make sure to include a legend and axis labels. Roughly how long does the system take to reach equilibrium?

```
kon = .100; %1e2/(uM s)
koff = .100;
function dy = dY(t, y)
    dy=zeros(3,1);
    dy(1) = -kon*y(1)*y(2) + koff*y(3);
    dy(2) = -kon*y(1)*y(2) + koff*y(3);
    dy(3) = kon*y(1)*y(2) - koff*y(3);
end
[t,y] = ode45(@dY, [0 50], [2 1 0]);
```

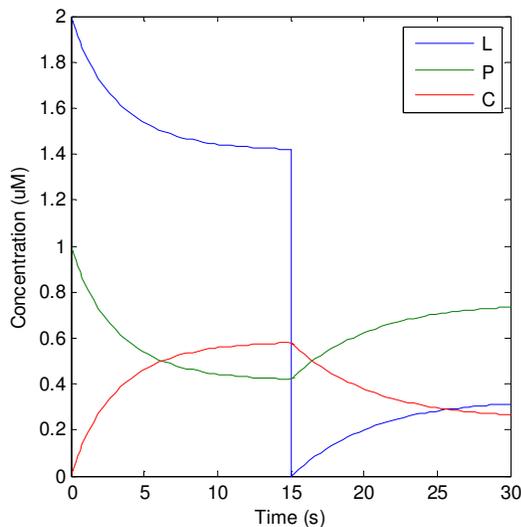
20 ms or so.



f) This system equilibrates too fast for us to easily study it. Suppose we make both k_{on} and k_{off} 1000 times smaller. Does this alter the equilibrium concentrations? With these new parameters, approximately how long does it take to reach equilibrium? (You do not have to solve this analytically, just plot it and estimate roughly where it equilibrates. You do not need to include a plot here.)

This does not alter the equilibrium concentrations (as K_d stays constant), and the time to equilibrium increases to 15-20 s.

g) Now we're going to perturb our system. Using the slow k_{on} and k_{off} in (f), we are going to simulate the system starting from $[P_0] = 1\mu\text{M}$, $[L_0] = 2\mu\text{M}$. However, 15 seconds after starting the binding reaction, we are going to 'magically' remove all the of unbound ligand and observe the system for another 15 seconds. Plot the concentrations for each species during this experiment.



h) This is similar to SPR, but in SPR the ligand solution or buffer solution is constantly flowing over the surface. Write down the modified differential equations for each phase of the experiment to take this into account. Are the equations linear or nonlinear? Simulate it and make a plot as in (g).

When we're flowing ligand over at concentration L_0 :

$$d[P]/dt = -k_{on} [P]L_0 + k_{off} [C]$$

$$d[L]/dt = 0$$

$$d[C]/dt = k_{on} [P]L_0 - k_{off} [C]$$

When we're washing with buffer

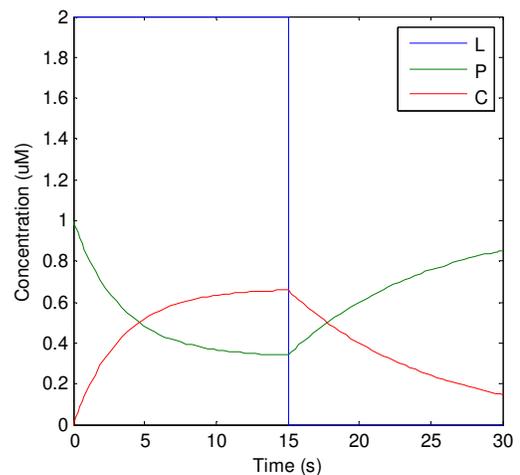
$$d[P]/dt = k_{off} [C]$$

$$d[L]/dt = 0$$

$$d[C]/dt = k_{on} [P]L_0 - k_{off} [C]$$

= $-k_{off} [C]$ because L_0 is zero.

Both systems of equations are now linear!



```

kon = .100; %1e2/(uM s)
koff = .100;
function dy = dY(t, y)
    dy=zeros(3,1);
    dy(1) = 0 %ligand
    dy(2) = -kon*y(1)*y(2) + koff*y(3); %protein
    dy(3) = kon*y(1)*y(2) - koff*y(3); %complex
end

[t,y] = ode45(@dY, [0 15], [2 1 0]);
[t2,y2] = ode45(@dY, [15,30], [0 y(length(t),2:3)]);
vertcat(t,t2)
plot(vertcat(t,t2), vertcat(y, y2));
legend('L', 'P', 'C');
xlabel('Time (s)');
ylabel('Concentration (uM)');
disp(y(length(y),:))

```

i) (Optional, no points) To understand why we solve this system numerically, suppose we started with $[P] = A$, $[L] = B$, and $[C] = 0$. Then if we were to write down a differential equation for just $[C]$, we could write it as

$$\frac{d[C]}{dt} = k_{on}(A - [C])(B - [C]) - k_{off}[C]$$

Go to Wolfram Alpha (wolframalpha.com) and type in 'solve ' followed by the above equation (you will need to use two single letter variables for k_{on} and k_{off}). What is the solution?

MIT OpenCourseWare
<http://ocw.mit.edu>

20.320 Analysis of Biomolecular and Cellular Systems
Fall 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.