20.320 — Problem Set # 2

October 1st, 2010

Due on October 1st, 2010 at 11:59am. No extensions will be granted.

General Instructions:

1. You are expected to state all 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 and the corresponding code must be printed and attached to the solutions. 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. Staples 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 the graded problem set 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.

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 was provided on the first day of class and follow the guidelines carefully.

1 Enzyme Kinetics

An enzyme E catalyzes the reaction of a substrate S into a product P. The product of this reaction is a powerful drug for headaches. An engineer at the production facility is interested in optimizing its production to make it more cost-effective. The enzyme has already been fully characterized previously with $k_{cat} = 50\ s^{-1}$, $k_1 = 10^7 M^{-1} s^{-1}$, $k_{-1} = 50 s^{-1}$. Answer the following questions:

a) If the reactor contains the enzyme at a concentration of 50nM, what is the maximal turnover rate of the reactor?

Solution:

$$\nu_{\max} = [\mathrm{E}]_0 \cdot \mathrm{k_{cat}} = 5 \cdot 10^{-8}\mathrm{M} \cdot 50\mathrm{s}^{-1} = 2.5 \cdot 10^{-6}\mathrm{M} \cdot \mathrm{s}^{-1}$$

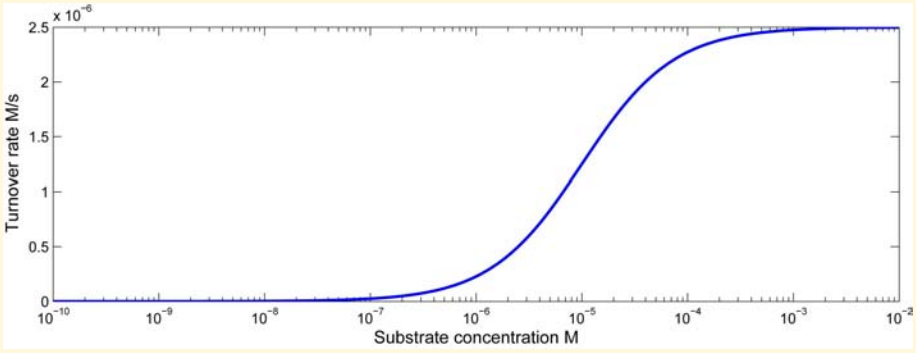
Total 1 point.

b) Using , plot the turnover over rate as a function of substrate concentration.

Solution:

Turnover rate as a function of substrate concentration is given by:

$$\nu = \frac{\mathrm{k_{cat}} \cdot [\mathrm{E}]_0 \cdot [\mathrm{S}]}{\mathrm{K_M} + [\mathrm{S}]}$$



Note that this expression is only valid if the QSSA is respected.
Total 1 point: 0.5 for correct mathematical expression, 0.5 for plot

c) How can the reactor be engineered so that maximal product turnover rate is guaranteed over an extended period of time (i.e. days or weeks)?

Solution:

You need a constant flow of substrate inside the reactor, and flow out the product. You might need to somehow capture the enzyme into the reactor (encapsuled in gel beads for example).

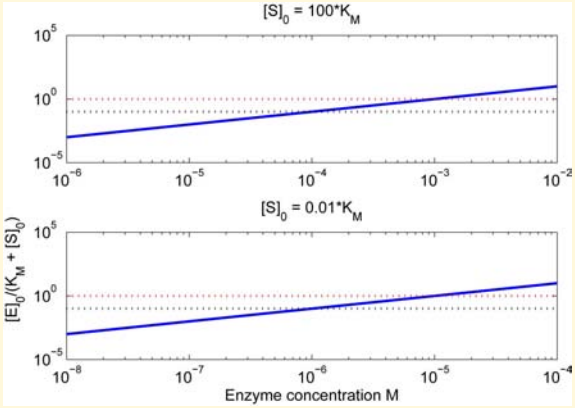
Total 2 point: Stating constant flow of substrate to maintain high concentration.

d) Using , determine the enzyme concentration range for which the QSSA is valid when i) $[S] = 100 \cdot K_M$, ii) $[S] = 0.01 \cdot K_M$

Solution:

The criteria for QSSA validity is:

$$\frac{[E]_0}{K_M + [S]_0} << 1$$



The QSSA is satisfied for enzyme concentration $<10^{-4}$ M and $<10^{-6}$ M for case i) and ii) respectively.

4 points total: 1 point for QSSA criteria, 1 point for ranges, 2 points for plot

e) We want to investigate how long it would take to deplete 90% of substrate if $[S]_0 = K_M$. To do so first obtain an analytical expression by making any assumptions you find necessary. Using that expression, obtain a confidence interval for the 90% depletion time. Then solve the problem using the ODE solver in . Discuss your results.

Solution:

1. Analytical solution:

   - First check whether QSSA applies here:

$$\frac{[\mathrm{E}]_0}{\mathrm{K_M} + [\mathrm{S}]_0} = \frac{5 \cdot 10^{-8}}{10^{-5} + 10^{-5}} = 0.0025 << 1$$

   - Since the QSSA criteria is satisfied, we can model the product formation rate using Mikaelis-Menten.

$$\frac{\mathrm{d[P]}}{\mathrm{dt}} = \frac{\mathrm{k_{cat}} \cdot [\mathrm{E}]_0 \cdot [\mathrm{S}]}{\mathrm{K_M} + [\mathrm{S}]}$$

   - As the substrate is being depleted, the product formation rate is decreasing. We can set therefore two boundaries for the 90% depletion time. Let us first assume that the product formation rate remains that of the initial rate throughout the experiment:
     - At the beginning $[\mathrm{S}]_0 = \mathrm{K_M}$, therefore the product fomration rate is:

$$\frac{\mathrm{d[P]}}{\mathrm{dt}} = \frac{\mathrm{k_{cat}} \cdot [\mathrm{E}]_0 \cdot [\mathrm{S}]}{\mathrm{K_M} + [\mathrm{S}]} \approx \frac{1}{2} \cdot \mathrm{k_{cat}} \cdot [\mathrm{E}]_0$$

     - We can assume that the overall change in substrate concentration is equivalent to that of product concentration change:

$$\frac{\Delta[\mathrm{P}]}{\Delta\mathrm{t}} \approx \frac{\Delta[\mathrm{S}]}{\Delta\mathrm{t}} \approx \frac{1}{2} \cdot \mathrm{k_{cat}} \cdot [\mathrm{E}]_0$$

     - Solving for Δt:

$$\Delta\mathrm{t} = \frac{\Delta[\mathrm{S}]}{0.5 \cdot \mathrm{k_{cat}} \cdot [\mathrm{E}]_0} = 7.2s$$

   - Now if we consider the product formation rate when there is 10% of substrate left:

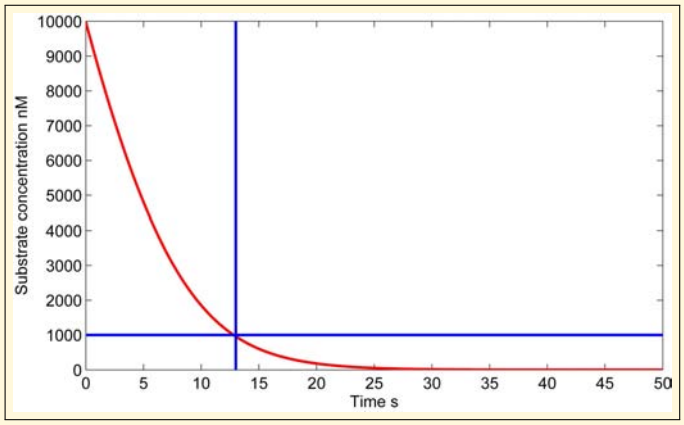
$$\frac{\mathrm{d[P]}}{\mathrm{dt}} = \frac{1}{11} \cdot \mathrm{k_{cat}} \cdot [\mathrm{E}]_0$$

   - Using the same approach as above we obtain Δ t = 39.6 s

2. The difference between the numerical solution and the approximation using Mikaelis-Menten is due to the fact that Mikaelis-Menten gives an initial rate, and the product turnover decreases as the substrate concentration drops.

Solution Continued:

2. : Solving the associated differential equation system with the ODE solver allow to find that 90% substrate depletion takes ≈ 13s, which is well within our confidence interval.



Total 12 points: 2 points for verifying QSSA, 4 points for analytical form of substrate depletion (other reasonable alternatives accepted if justified), 2 points for interval (any reasonable upper and lower bound interval accepted if justified), 2 points for routine, 2 points for extracting 90% substrate depletion from .

20 points overall for problem 1.

MATLAB code for Problem 1

enzyme_kinetics.m:

```
%-------------------------------------------------------------%
%                       Problem 1 - PSET 2
%-------------------------------------------------------------%

function enzyme_kinetics
close all;
clc;

kcat = 50; %s^-1
k1 = 1e7; % M^-1 s^-1
k_minus1 = 50; %s^-1
E0 = 50e-9; %M
Km = (kcat + k_minus1)/k1
vmax = E0*kcat

% Mikaelis-Menten product formation rate
rate = inline('C(1)*C(2)*S./(C(3)+S)', 'C', 'S');
% C(1) = kcat, C(2) = E0, C(3) = Km

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                             Part b)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(1);
S = logspace(-10,-2,100);
semilogx(S, rate([kcat E0 Km], S));
xlabel('Substrate concentration M', 'FontSize', 12)
ylabel('Turnover rate M/s', 'FontSize', 12)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                             Part d)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Criteria for QSSA: E0/(Km + S) << 1
QSSA_criteria = inline('E./(b(1) + b(2))', 'b', 'E');
% b(1) = Km, b(2) = S0

figure(2);
S01 = 100*Km;
S02 = 0.01*Km;

subplot(2,1,1);
E = logspace(-6,-2,100);
loglog(E, QSSA_criteria([Km S01], E),'b');
hold on;
plot(E,ones(1,100), 'r:', 'LineWidth', 2);
plot(E,0.1*ones(1,100), 'k:', 'LineWidth', 2);
title('[S]_0 = 100*K_M')
hold off;

subplot(2,1,2);
E = logspace(-8,-4,100);
loglog(E, QSSA_criteria([Km S02], E),'b');
hold on;
plot(E,ones(1,100), 'r:', 'LineWidth', 2);
plot(E,0.1*ones(1,100), 'k:', 'LineWidth', 2);
```

```
hold off;
title('[S]_0 = 0.01*K_M')
xlabel('Enzyme concentration M', 'FontSize', 12)
ylabel('            [E]_0/(K_M + [S]_0)', 'FontSize', 12);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                              Part e)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
kcat = 50; %s^-1
k1 = 1e7*1e-9; % nM^-1 s^-1
k_minus1 = 50; %s^-1
E0 = 50; %nM
Km = (kcat + k_minus1)/k1;
S0 = Km;

x0 = [S0 E0 0 0]; %Initial Conditions
P = [kcat k1 k_minus1]; % Vectors containing the system's constants
[T, Y] = ode23(@(t,y)Enzyme_Kinetics_Equadiff(t, y, P), [0 50], x0);

figure(3);
subplot(2,1,1);
plot(T, Y(:,4)) % Plot the product concentration
hold on;
plot(T,Y(:,1), 'r'); % Plot the substrate concentration
hold off;
xlabel('Time s');
ylabel('Concentration nM');
legend('Product', 'Substrate');

subplot(2,1,2);
plot(T, Y(:,2)+Y(:,3));
xlabel('Time s');
ylabel('Enzyme concentration nM');
title('Conservation of enzyme');

figure(4)
plot(T, Y(:,1), 'r'); % Substrate as a function of time. ODE solution
hold on;
line([0 50], [1000 1000]);
line([13 13], [0 10000]);
hold off;
xlabel('Time s');
ylabel('Substrate concentration nM');

%-------------------------------------------------------------------------%
function xdot = Enzyme_Kinetics_Equadiff(t, x, P)
    % P = [kcat k1 k_minus1]
    % x = [x(1) x(2) x(3) x(4)] == [S E ES P]
xdot = [-P(2)*x(2)*x(1) + P(3)*x(3);... % dS/dt
        -P(2)*x(2)*x(1) + (P(3)+P(1))*x(3);... % dE/dt
        P(2)*x(2)*x(1) - (P(3)+P(1))*x(3);...%dES/dt
        P(1)*x(3)]; % dP/dt
end

end
```

2 Protein Array

EGFR family receptors regulate cell processes such as proliferation, migration, apoptosis and differentiation, and are implicated in development and cancer. This problem refers to the microarrays in Jones *et al.*, but all the data given here has been artificially generated.

a) In microarrays, the fluorescent signal depends on binding between labeled protein and the tethered protein. Give a control that would allow you to take into account the variation in the number of tethered protein per spot.

Solution:

The tethered protein can be labeled with a fluorophore. It is important that this fluorophore exciting/emiiting wavelengths do not overlap with the ones from the fluorescently labeled ligand.
Total 2 points: 1 point for labeling, 1 point for a system that does not affects the detection signal. Other alternatives accepted.

b) Is that a useful control? Explain why or why not.

Solution:

It is important to check whether or not the proteins were immobilized on the chip. However, you do not need to normalize the signal by the amount of bound protein or peptide since the component of interest will be the fractional saturation.
Total 1 point

c) Jones *et al.*, performed a titration using 8 different concentrations. The data generated is given in Jones_Nat_2006.mat. The matrix is of size 10x10x8. The first dimension represents the phosphorylated peptides, the second dimension represents the various proteins and the third dimension represents each individual experiment. Use this data to extract the equilibrium dissociation constant for each of these interactions (there are 100, so you need to come up with an automated system).

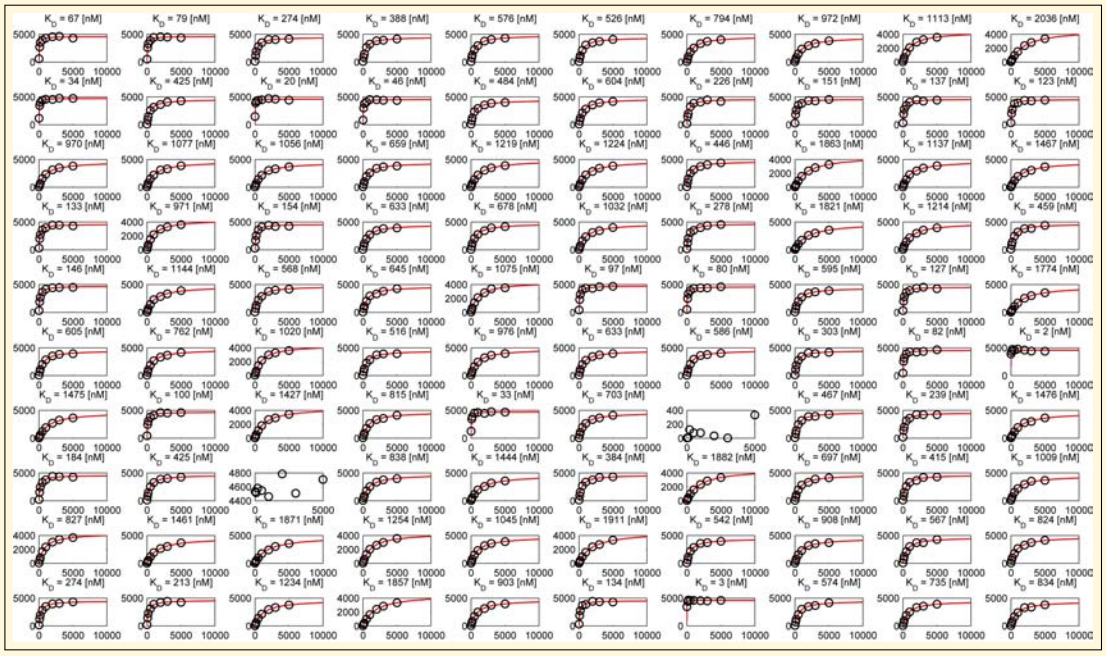
Index	**Phosphopetides**	**Adaptor proteins**	**Concentrations nM**
1	EGFR pY998	ABL2	10
2	EGFR pY1069	GRB2	100
3	EGFR pY1110	PIK3R2-N	200
4	EGFR pY1125	SH3BP2	500
5	ErbB2 pY1023	BMX	1,000
6	ErbB2 pY1221	TENC1	2,000
7	ErbB3 pY1197	TENS1	3,000
8	ErbB3 pY1262	GRAP2	5,000
9	ErbB3 pY1328	PIK3R3-C	
10	ErbB4 pY1188	PLCG1-C	

Solution:

This data represents protein-ligand interaction. Using the PFOA, the fluorescent signal collected by Jones *et al.* can be modeled as:

$$F_{Obs} = F_{Max} \cdot \frac{[L]_0}{[L]_0 + K_D}$$

Using the function `nlinfit`, the parameters F_{Max} and K_D can be extracted for each interaction:



Total 10 points: 2 points for stating PFOA, 2 points for fitting with monovalent binding isotherm, 4 points for iterative loop use, 2 points for correct K_D values

d) Were you able to estimate a relevant K_D for each interaction? If not why?

Solution:

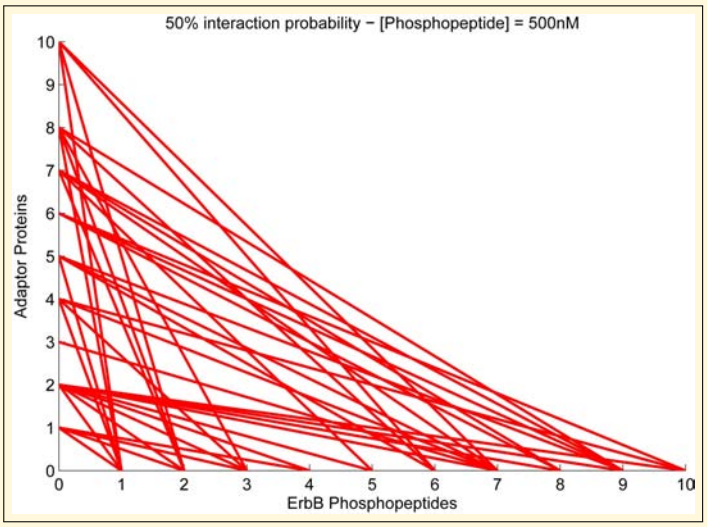
There are two two phosphopeptide-protein interaction for which nlinfit is unable to fit a monovalent binding isotherm. A quick investigation of the data set allows to identify these two interactions:

- ErbB3 pY1197 - TENS1: Here the signal is ranging from 0 to 300. This is indicative that the K_D is too weak to be detected with the concentration ranged used here.
- ErbB3 pY1262 - PIK3R2-N: Here the signal is ranginf from 4400 to 4800. This is indicative that the K_D is too strong for the explored concentration range, therefore the system is always saturated and no K_D values can be estimated.

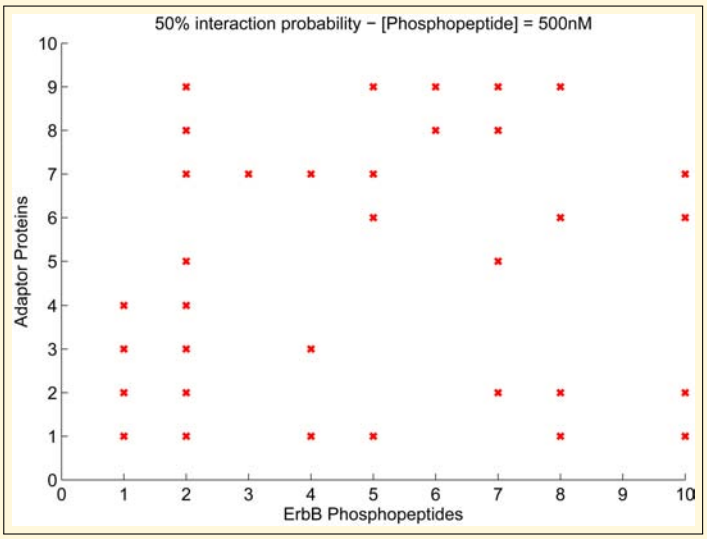
Total 6 points: 2 points for identifying partners, 2 points for each explanation.

e) A typical cell has on the order of 10^4 EGFR receptors. Let us imagine that upon EGF stimulation, there is a phosphopeptide local concentration of 500nM. Draw the interaction network generated. You should have on the x-axis the phosphopeptides, on the y-axis the interaction protein and you should connect a protein to a phosphosite if the probability that they are bound is higher than 50%. *Do not consider competition.*

Here we will assume that the PFOA is valid. In that case, if we consider one adaptor - one phosphopeptide system, for the peptide to be bound with >50% chance by the specific adaptor, the K_D for this interaction must be <= 500nM. For each interaction, if the K_D value respect the criteria, a line is drawn on the plot to connect the adaptor protein to the phosphopeptide:



Alternatively, another representation example:



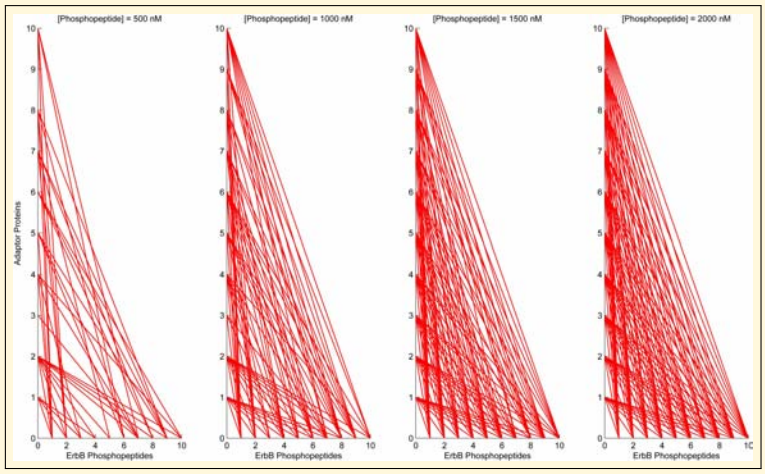
Total 12 points: 2 points for good use of iterative loops, 3 points for edge criteria, 2 points for leaving out the two non measurable interaction, 3 points for correct interaction indication, 2 points for plot

---

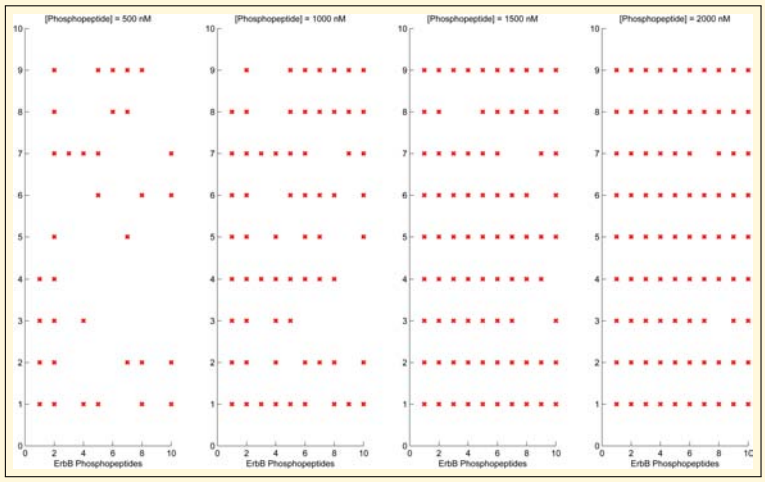
f) Some cancer cells upregulate the expression of EGFR and have on the order of 10^5 EGFR receptors on their surface. Let us consider 3 different scenarios where the concentration of phosphopeptides is i) 1μM, ii) 1.5μM and iii) 2μM. How does the system evolve?

Solution:

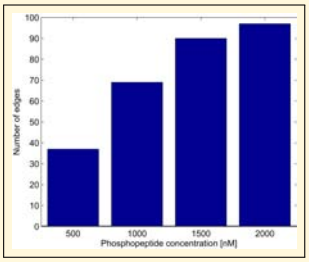
As the phosphopeptide cocnentration increases, the number of edges in the network increases as well:



Alternatively, another representation:



We observe that the number of interactions increases as the receptor expression increases:



Total 3 points: 2 points for plots (histogram is not required), 1 point for explanation

g) Let us consider one specific phosphosite on EGFR and two adaptor proteins A and B. The concentration of the adaptor proteins are 1nM and 500nM respectively. If the equilibrium dissociation constant for each of these adaptor proteins is 0.5nM and 250nM, plot the percentage occupation of the phosphosite by each of these adaptor proteins as a function of the phosphopeptide concentration. Hint: you can assume k_{on} is 10^5-10^6 $M^{-1}s^{-1}$ for protein-protein interactions.

Solution:

We represent here the phosphopeptide by R and the adaptor proteins by L_A and L_B. The system is represented by the following diffential equations:

$$\frac{dL_A}{dt} = -k_{on,A} \cdot R \cdot L_A + k_{off,A} \cdot C_A$$

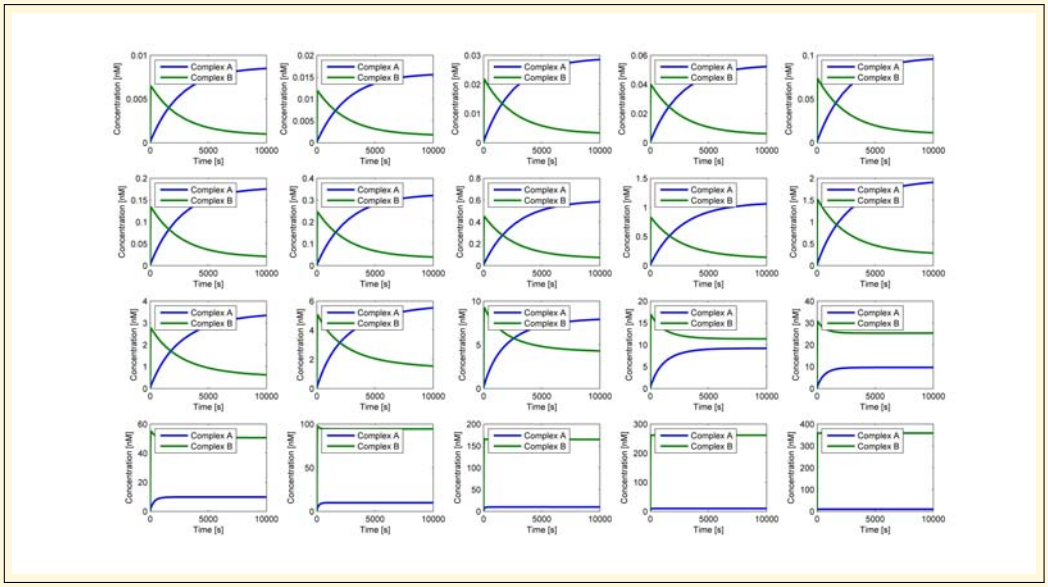
$$\frac{dL_B}{dt} = -k_{on,B} \cdot R \cdot L_B + k_{off,B} \cdot C_B$$

$$\frac{dC_A}{dt} = k_{on,A} \cdot R \cdot L_A - k_{off,A} \cdot C_A$$

$$\frac{dC_B}{dt} = k_{on,B} \cdot R \cdot L_B - k_{off,B} \cdot C_B$$

$$\frac{dR}{dt} = -k_{on,A} \cdot R \cdot L_A + k_{off,A} \cdot C_A - k_{on,B} \cdot R \cdot L_B + k_{off,B} \cdot C_B$$

This system can be solved using an ODE solver in . However, we must first check that the concentrations extracted from the ODE solver are at equilibrium. The complex concentrations are at equilibrium as shown in the figure below:

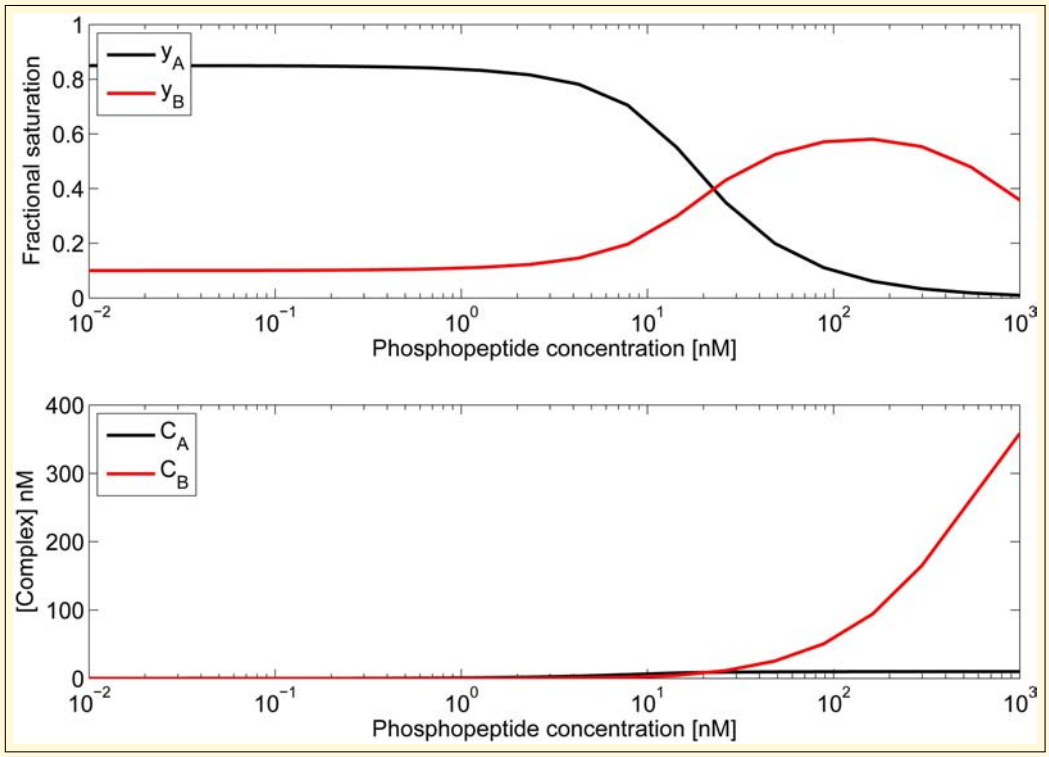


Solution contiuned:

The fraction of phosphopeptide occupied by adaptor A and B give by:

$$y_{A,B} = \frac{[C]_{A,B}}{[R]_0}$$

Where $[C]_{A,B}$ is the concentration of adaptor complexed to the phosphopeptide, and R is the total concentration of phosphopetide. The results are shown in the figure below:



Total 14 points: 4 points for correct system of ODEs, 3 points for correct use of ode solver in , 3 points for checking complex equilibrium, 2 points for correct fractional saturation expression, 2 points for fractional saturation plot.

h) What do the results obtained in e) and f) tell you about the importance of EGFR overexpression in cancer cells?

The EGFR network is a tightly regulated system. Overexpression of ErbB receptors yields to the activation of downstream effectors that are not involved in signaling in normal expression cell lines.
Total 1 point

49 points overall for problem 2.

MATLAB code for Problem 2

Jones_Solution.m:

```
%-------------------------------------------------------------------------%
%                          Problem 2 - PSET 2
%-------------------------------------------------------------------------%

function Jones_Solution
clc;
close all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     Part c)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load Jones_Nat_2006 Data

f1 = inline('C(1)*L./(L+C(2))', 'C', 'L');
L = [10 100 200 500 1000 2000 3000 5000];
C0 = [2000 1000];
Kd = zeros(10,10);
x = logspace(0, 4, 25);
v = zeros(1,8);

figure(1)
for i=1:10
    for j=1:10
        for t=1:8
                v(t) = Data(i,j,t);
        end
        if i==7 && j==7 || i==8 && j==3 % Removes odd data from analysis
            subplot(10,10, 10*(i-1)+j);
            plot(L, v, 'ko')
        else
            Fit = nlinfit(L, v, f1, C0);
            Kd(i,j) = Fit(2);
            subplot(10,10, 10*(i-1)+j);
            plot(x, f1(Fit, x), 'r-');
            hold on;
            plot(L, v, 'ko');
            hold off
            title(sprintf('K_D = %.0f [nM]', Kd(i,j)));
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                 Part e)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

f2 = inline('a(1)*x + a(2)', 'a', 'x'); % for connections
x= linspace(0,10,4);
figure(2); % Interaction plot - with connections
hold on;
axis([0 10 0 10])
for i = 1:10
    for j=1:10
```

```
        if Kd(i,j) ≤ 500
            constants = [-i/j i];
            plot(x, f2(constants,x), 'r');
        end
    end
end
hold off;
xlabel('ErbB Phosphopeptides');
ylabel('Adaptor Proteins');
title('50% interaction probability - [Phosphopeptide] = 500nM');

figure(10); % Interaction plot - with crosses
hold on;
axis([0 10 0 10])
for i = 1:10
    for j=1:10
        if i==7 && j==7 || i==8 && j==3
        else if Kd(i,j) ≤ 500
            plot(i,j, 'rx');
            end
        end
    end
end
xlabel('ErbB Phosphopeptides');
ylabel('Adaptor Proteins');
title('50% interaction probability - [Phosphopeptide] = 500nM');
hold off;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                  Part f)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(3)
condition = [500 1000 1500 2000];
counter = [0 0 0 0];
for k= 1:4
    subplot(1,4,k)
    hold on;
    axis([0 10 0 10])
    for i = 1:10
        for j=1:10
            if i==7 && j==7 || i==8 && j==3
            else
                if Kd(i,j) ≤ condition(k)
                    constants = [-i/j i];
                    plot(x, f2(constants,x), 'r');
                    counter(k) = counter(k) + 1;
                end
            end
        end
    end
    title(sprintf('[Phosphopeptide] = %.0d nM', condition(k)));
    xlabel('ErbB Phosphopeptides');
    hold off;
end
subplot(1,4,1);
ylabel('Adaptor Proteins');
figure(11) % Connections with crosses
for k= 1:4
```

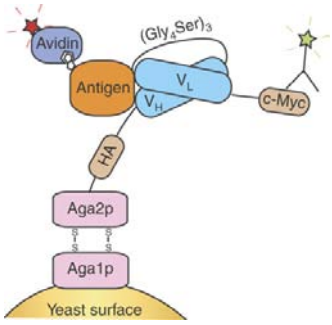
```
    subplot(1,4,k)
    hold on;
    axis([0 10 0 10])
    for i = 1:10
        for j=1:10
            if i==7 && j==7 || i==8 && j==3
            else
                if Kd(i,j) ≤ condition(k)
                    constants = [-i/j i];
                    plot(i,j, 'rx');
                end
            end
        end
    end
    title(sprintf('[Phosphopeptide] = %.0d nM', condition(k)));
    xlabel('ErbB Phosphopeptides');
    hold off;
end

figure(4)
bar(counter);
xlabel('Phosphopeptide concentration [nM]');
ylabel('Number of edges');
set(gca, 'XTickLabel', [500 1000 1500 2000]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                          Part g)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

LA_0 = 10; %nM
LB_0 = 500; %nM
R = logspace(-2, 3, 20); %nM

KD_A = 0.5; %nM
KD_B = 250; %nM

konA = 1e5*1e-9; %[nM^-1 s^-1]
konB = konA;
koffA = konA*KD_A;
koffB = konB*KD_B;

Fractional_sat = zeros(20,2);
Final_complex = zeros(20,2);

figure(5)
for i=1:20
    R_0 = R(i);

    x0 = [LA_0 LB_0 0 0 R_0];    % Initial Conditions
    P = [konA konB koffA koffB]; % Vectors containing the system's constants

    [T, Y] = ode15s(@(t,y)competition(t, y, P), [0 10000], x0);
    %10000 seconds should be long enough to reach equilibrium
    Final_complex(i,:) = [Y(end,3) Y(end,4)];
    Fractional_sat(i,:) = [Y(end,3)./R_0 Y(end,4)./R_0];
    subplot(4,5,i);
      %  to confirm that you have reached steady-state
    plot(T, Y(:,3), T, Y(:,4));
    xlabel('Time [s]');
```

```
    ylabel('Concentration [nM]');
    legend('Complex A', 'Complex B', 'Location', 'NorthWest');
end

figure(6);
subplot(2,1,1);
semilogx(R, Fractional_sat(:,1), 'k');
hold on;
semilogx(R, Fractional_sat(:,2), 'r');
hold off;
legend('y_A', 'y_B', 'Location', 'NorthWest');
xlabel('Phosphopeptide concentration [nM]');
ylabel('Fractional saturation');

subplot(2,1,2);
semilogx(R, Final_complex(:,1),'k');
hold on;
semilogx(R, Final_complex(:,2),'r');
hold off;
legend('C_A', 'C_B', 'Location', 'NorthWest');
xlabel('Phosphopeptide concentration [nM]');
ylabel('[Complex] nM');


%-----------------------------------------------------------------------%
function xdot = competition(t, x, P)
    % P = [konA konB koffA koffB]
    % x = [x(1) x(2) x(3) x(4) x5)] == [LA LB CA CB R]
xdot = [-P(1)*x(1)*x(5) + P(3)*x(3);... % dLA/dt
        -P(2)*x(2)*x(5) + P(4)*x(4);... % dLB/dt
        P(1)*x(1)*x(5) - P(3)*x(3);...%dCA/dt
        P(2)*x(2)*x(5) - P(4)*x(4);...%dCB/dt
        -P(1)*x(1)*x(5)+ P(3)*x(3) - P(2)*x(2)*x(5) + P(4)*x(4)]; % dR/dt
end

end
```

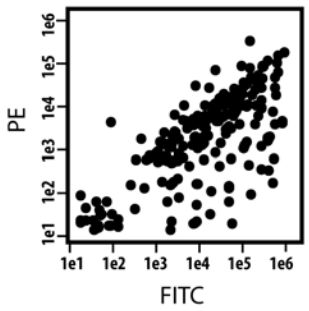
3 Yeast Surface Display (YSD)

In this problem, you will apply the principles for the analysis of binding interactions covered in class to a new experimental context.

Yeast Surface Display is an alternative method for measuring kinetic and thermodynamic parameters for binding interactions. By expressing the protein of interest as a fusion with Aga-2p cell surface protein, you can have it presented on the outside surface of the yeast cell. Furthermore, the protein is flanked by two short peptide tags (HA and cmyc) against which good antibodies are commercially available. In the diagram below, the protein of interest is a singel chain variable fragment (scFv) of an antibody, comprising the light variable domain V_L, the heavy variable domain V_H, and a flexible $(Gly_4Ser)_3$ linker. The ligand of interest is a biotinylated antigen. They will be referred to as P and L in this problem.



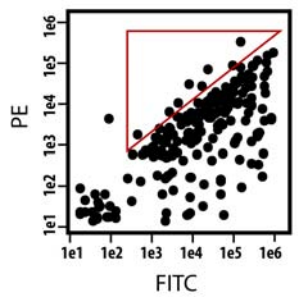
You have generated a library of mutated scFvs by using error-prone PCR on the gene for a ligand-binding antibody. You incubate a population of yeast cells expressing the library of P mutants on their surface first with FITC-labelled anti-myc antibody (green; binds to cmyc peptide) and then with phycoerythrein (PE)-labelled streptavidin (red) and biotinylated ligand. You run the library through a flow cytometer and plot red vs. green fluorescence for each cell. In the resulting plot, each dot corresponds to a single cell and its coordinates mark the red and green fluorescence intensities.



a) You want to sort the best 1% of binders for the next round of directed evolution. On the plot above, circle the part of the cell population which shows significantly above-average protein-ligand binding affinity (a very rough sketch is sufficient).

Solution:

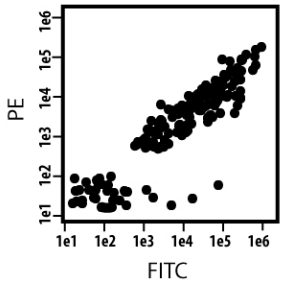
PE fluorescence is proportioanl to bound ligand while FITC fluorescence is proportional to the number of protein copies expressed per cell. The best binders will be to the top-left of the diagnonal. Far-left are expressed at low levels — they should be excluded.



3 points.

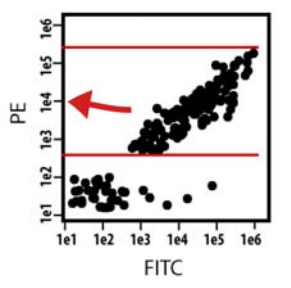
b) To quantitatively track the improvement of your library through successive rounds of evolution, you run another sample on the flow cytometer, this time containing only the top binders which you selected for the next round. You now want to determine the average K_D for this population. For this purpose, you will incubate a number of separate samples containing on average a few 100 cells displaying each mutant antibody with a number of different ligand concentrations, label with FITC-anti-cmyc antibody, and perform the measurement.

i) On the FACS plot below, indicate where you can read off a measure of the *average fractional saturation* of protein with ligand for all functional antibody mutants at any one given ligand concentration. Explain in one short sentence how you found that point.



Solution:

Two kinds of solutions are acceptable. Most correctly, the ratio of average (mean or median) PE fluorescence and average FITC fluorescence should be taken as a measure of the fractional saturation. However, it may be assumed (or better, experimentally verified) that expression is invariant for the several tubes into which the original population (grown and induced together in one flask) was split. In that case, mean FITC fluorescence is a constant and mean PE may directly be taken to represent the fractional saturation.

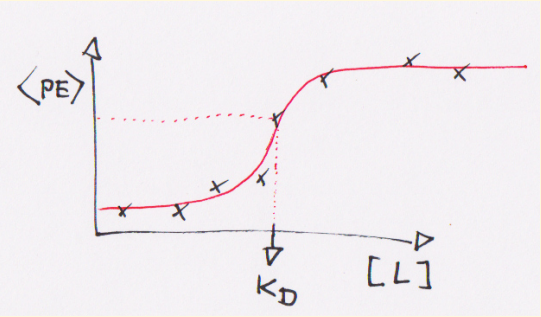


3 points: 1 excluding low-expression / uninduced cells, 1 for realizing that average PE is the key measure of the amount of ligand present per cell, and 1 for either normalizing it by FITC fluorescence or realizing that FITC is typically constant.

ii) Schematically (no numbers needed on the axes), draw a labelled graph to illustrate how you would plot the quantity you measure for each ligand concentration against [L]. Indicate how to graphically estimate K_D from this plot.

Solution:

This will be a binding isotherm, and the [L] at which half-maximal saturation is obtained is the K_D.



2 points: 1 for the the isotherm (linear or log), 1 for K_D.

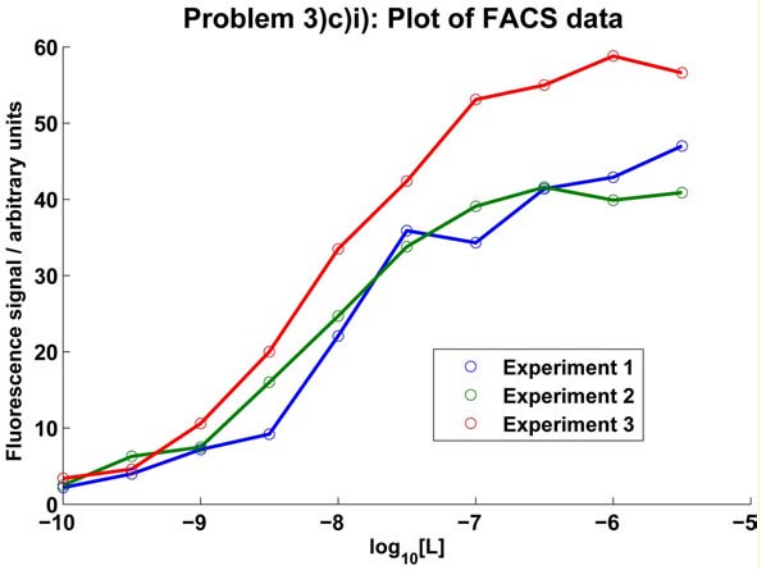
c) The following is data from an equilibrium binding experiment by YSD in triplicate [1]

[1]The interaction being measured was between an engineered antibody and a small molecule chelator of

[L] / nM	Experiment 1	Experiment 2	Experiment 3
3160	47.0	40.9	56.6
1000	42.9	39.9	58.8
316	41.4	41.6	55.0
100	34.3	39.1	53.1
31.6	35.9	33.8	42.4
10	22.1	24.7	33.5
3.16	9.2	16.0	20.0
1	7.2	7.5	10.6
0.316	4.0	6.3	4.6
0.1	2.2	2.5	3.4
0	2.0	1.9	0.0

i) Plot the raw data. Comment.

Solution:



The data broadly follows the sigmoidal (in a semilog plot!) shape of a binding isotherm. Clearly it must be normalized since the absolute fluorescence values differ markedly betweeb experiments, likely a consequence of slighly different staining times.

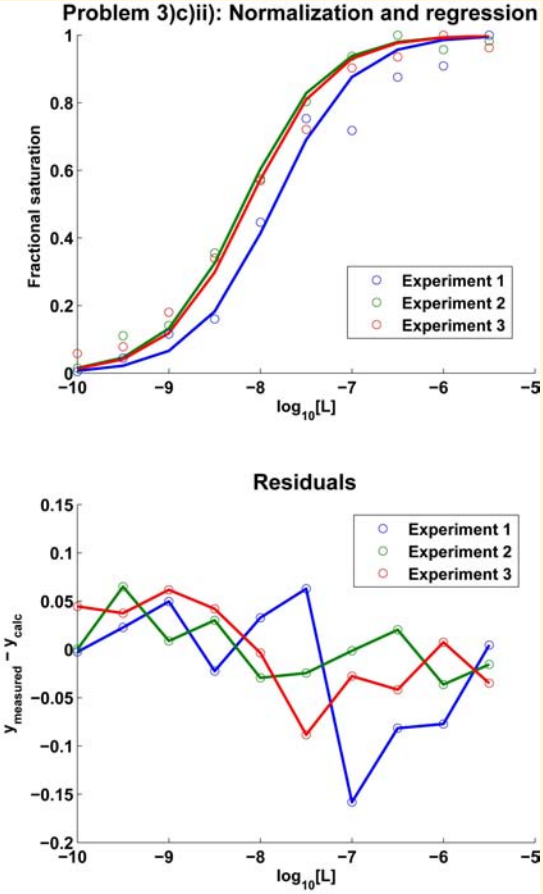
8 points: 2 each for code; plot; realizing a binding isotherm is indeed traced out; noting that the datasets have to be processed in some way before parameters can be extracted. Points can also be given for alternative reasonable and relevant observations.

ii) Extract K_D values by regression for each experiment. Using the `subplot` command in , plot the experimental data (normalized) and fitted isotherm in one panel, and

radioisotopes for pre-targeted radioimmunotherapy (PRIT). This antibody has since been tested in a mouse model for treatment of micrometastases; the results were accepted for publication three weeks ago.

the residuals in a second panel of the same figure. Are you satisfied with your results? Explain.

Solution:



KD from experiments 1, 2, and 3 is 14.16, 6.56, and 7.44 nM, respectively. The mean ± SD is 9.39 ± 4.16 nM (and given this margin of error, should be reported as 9 ± 4 nM).

11+2 points: 4 for normalization (substract the baseline and rescale), 3 for K_D's, 2 points for residuals (calculation and plot), 2 points for intelligent discussion of residuals (look mostly good; no clear systematic deviation; dataset 1 looks less good than others). 1 extra credit each for reporting mean K_D and SD.
K_D values may be different for several reasons: Did not subtract baseline (give 1/3 points for K_D), normalized correctly but regressed with 2 rather than 1 free parameter on normalized data (not ideal but give 3/3 for K_D), syntax error in fitted function (zero points for K_D).

d) In this type of experiment, incubation with ligand is performed in tubes. Why will ligand depletion become a constraint when testing very high-affinity interactions? What can you do to mitigate it?

Solution:

Will need large volumes to not deplete ligand while still incubating with small [L]. Ligand depletion will lead to lower-than-expected fluorescence intensities at low [L], giving too large an estimat of K_D.

7 points: 3 for realizing that high-affinity interactions require measurements at low [L for accurate K_D determination; 2 for for realizing that at low [L], the tube volume becomes a constraint (need larger volume to prevent depletion); 2 point for any reasonable mitigating strategy (use larger tubes; use fewer cells; induce cells less so that fewer proteins per cell are expressed; ...).

34 points +2 EC overall for problem 3.

MATLAB code for Problem 3

firstname_lastname_ysd.m:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%  SOLUTION FOR 20.320 PROBLEM SET 2, PROBLEM 3
%  FALL 2010
%
%  YEAST SURFACE DISPLAY
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function firstname_lastname_ysd
clc;
close all;

%Load data
[L, data] = ysd_init();

%Problem 3)c)i): Plot the data
datalabels = {'Problem 3)c)i): Plot of FACS data',
              'log_{10}[L]',
              'Fluorescence signal / arbitrary units'};
figure;
plot_data(L, data, data, datalabels);

%Problem 3)c)ii)
%Normalize data, perform regression, then plot isotherms and normalized
%data
regression(L, data);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Lout, dataout] = ysd_init()

% Vector of ligand concentrations in 1e-9 M
Lout = [ 0;
         0.1;
         0.316;
         1;
         3.16;
         10;
         31.6;
         100;
         316;
         1000;
         3160];

% Vector of raw fluorescence readouts, in arbitrary units. Each col is a
% separate expt.
dataout = [ 2.0     1.9     0.0 ;
            2.2     2.5     3.4 ;
            4.0     6.3     4.6 ;
            7.2     7.5     10.6 ;
            9.2     16.0    20.0 ;
            22.1    24.7    33.5 ;
            35.9    33.8    42.4 ;
```

```
            34.3     39.1    53.1 ;
            41.4     41.6    55.0 ;
            42.9     39.9    58.8 ;
            47.0     40.9    56.6  ];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function plot_data(LArray, dataPoints, dataLines, labels)
hold on;
plot(log10(LArray .* 1e-9), dataLines, '-', 'LineWidth', 2);
hl1 = plot(log10(LArray .* 1e-9), dataPoints, 'o');
hold off;
legend([hl1], 'Experiment 1', 'Experiment 2', 'Experiment 3', 'Location', 'Best')
title(labels(1),  'FontSize', 16, 'FontWeight', 'bold');
xlabel(labels(2),  'FontSize', 12, 'FontWeight', 'bold');
ylabel(labels(3), 'FontSize', 12, 'FontWeight', 'bold');
set(gca,'FontSize',12, 'FontWeight', 'bold');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function regression(LArray, dataArray)

% For each experiment separately...
for i=1:3
   %Substract baseline and normalize
   dataArray(:,i) = dataArray(:,i) - dataArray(1,i);
   dataArray(:,i) = dataArray(:,i) ./ max(dataArray(:,i));

   %Perform regression
   KD(i) = nlinfit(LArray, dataArray(:,i), @fracSatPFOA, 10);

   %Create isotherm
   fitdata(:,i) = fracSatPFOA(KD(i), LArray);

end

% calculate residuals
residuals = dataArray - fitdata;

% Report individual KDs, mean, and SD
KD
KDmean = mean(KD)
KDSD = std(KD)

%Plot normalized data as points, fitted isotherms as lines
figure;
subplot(2,1,1);
datalabels = {'Problem 3)c)ii): Normalization and regression';
              'log_{10}[L]';
              'Fractional saturation'};
plot_data(LArray, dataArray, fitdata, datalabels);
%Plot residuals
subplot(2,1,2);
datalabels = {'Residuals';
              'log_{10}[L]';
              'y_{measured} - y_{calc}'};
plot_data(LArray, residuals, residuals, datalabels);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function yeqPFOA = fracSatPFOA(KD,Lo)
   yeqPFOA = Lo ./ (Lo + KD);
```

20.320 Analysis of Biomolecular and Cellular Systems
Fall 2012