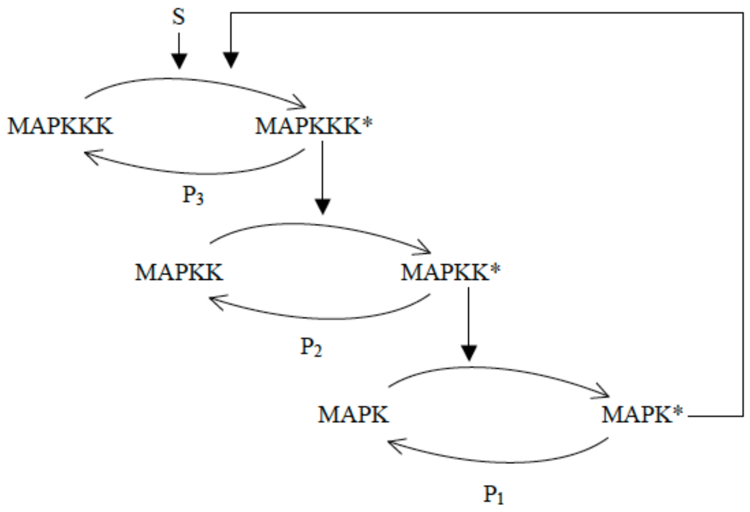
20.420J Principles of Molecular Biongineering
Homework 3: Kinetics & enzymes (due 10/31/17)

1. This problem concerns different signaling effects in MAP kinase pathways resulting from the action of positive versus negative feedback operating from the bottom to the top of the kinase cascade. Interestingly, Santos *et al.*, *Nat. Cell Biol. 9:* 324–330 (2007) report signaling through the MAP kinase pathway from the EGF receptor stimulates a negative feedback and induces transient activation leading to proliferation, whereas signaling through the same MAP kinase pathway from the TrkA receptor stimulates a positive feedback and induces sustained activation leading to differentiation.

a) Write a single ordinary differential equation for the rate of change of [MAPKKK*] in the simplified reaction scheme sketched below, representing the introduction of a signal S entering the MAP kinase cascade with a positive feedback loop connecting the MAPK* with the top of the cascade. This part asks only for a single differential equation.



Note that S, MAPKKK*, MAPKK*, and MAPK* are active kinase species that can be adequately treated with Michaelis–Menten kinetics (without explicit consideration of ATP or ADP). P_1, P_2, and P_3 are phosphatases whose kinetics can also be adequately treated with Michaelis–Menten kinetics. Unstarred species represent inactive versions of the starred ones (that is, MAPK is an inactive form of MAPK*). One simplification of the scheme here is the treatment of activation as due to a single action of kinase on substrate. Use KM(*enz*), kcat(*enz*), and [*enz*]0 to represent the Michaelis constant, catalytic constant, and total active enzyme concentration, respectively, for each of the enzymes *enz* in the scheme.

b) Simplify your equation from part a) for the limit of substrate concentrations much smaller than the relevant KM. For each term in your expression, state whether that term depends on a single time-varying concentration, or on the product of a pair of time-varying concentrations. Take note that some enzyme concentrations in the system are constant and others are dynamic, according to the given mechanism.

c) The same approximation can be applied to the entire network, leading to the following simplified mechanism of six reactions:

$$S + KKK \xrightarrow{\alpha_3} S + KKK^* \qquad KKK^* \xrightarrow{\gamma_1} KKK$$

$$KKK^* + KK \xrightarrow{\alpha_2} KKK^* + KK^* \qquad KK^* \xrightarrow{\gamma_2} KK$$

$$KK^* + K \xrightarrow{\alpha_1} KK^* + K^* \qquad K^* \xrightarrow{\gamma_3} K$$

where the α_i values represent bimolecular rate constants, and the γ_i values represent unimolecular rate constants. Write differential equations for *d*[KKK*]/*dt*, *d*[KK*]/*dt*, and *d*[K*]/*dt*.

d) Augment the mechanism above for a particular case of positive feedback whereby K* also catalyzes the activation of KKK via the reaction

$$K^* + KKK \xrightarrow{\alpha_0} K^* + KKK^*$$

How does this change your ODE model from part c)?

e) To look for stationary states of this simplified system, apply the steady-state approximation to [KKK*], [KK*], and [K*] using your augmented ODE model from part d). Are there stable activated states of the network in the absence of input signal S? That is, can [K*] be non-zero when [S] = 0 at steady state? If so, what values of [K*] are permitted?

f) Now replace the positive feedback in part d) with a negative feedback. Mathematically this can be accomplished by placing a minus sign in front of α_0. Name one biochemical mechanism that could accomplish this.

2.* You are interested in experimentally determining the ligand binding affinity of a cell surface receptor with K_d =10 pM and association rate constant k_{on} =10^5 $M^{-1}s^{-1}$. You incubate nine separate cell suspensions with the following concentrations of fluorescently labeled ligand: 0.1, 0.3, 1.0, 3.0, 10.0, 30.0, 100.0, 300.0, and 1,000.0 pM. Using Matlab OR Python (your choice), simulate the binding isotherm (binding curve) you would obtain if you measured cell-bound fluorescence at the following times: 15 minutes; 1 hour; 4 hours; 16 hours; 64 hours. You may assume pseudo-first order kinetics (*i.e.* the ligand is not used up by binding). Curve-fit each of these five isotherms to obtain an estimate of K_d. Is it necessary for every sample to reach equilibrium in order to obtain an accurate K_d estimate?

3. Use Matlab (we do not recommend Python for this) to generate 100 s timecourses of fluorescence with one data point per second, that decays according to the following double exponential function:

$f(t) = 0.5 \exp\{-k_1 t\} + 0.5 \exp\{-k_2 t\}$

For the following combinations of k_1 and k_2: a) 1 s^{-1}, 10 s^{-1}; b) 10 s^{-1}, 20 s^{-1}; c) 10 s^{-1}, 100 s^{-1}. Now use fitting functions (in Matlab) to compare a monoexponential model ($y = c_1 \exp\{-c_2 t\}$) to a biexpoential model ($y = c_1 \exp\{-c_2 t\} + c_3 \exp\{-c_4 t\}$) in fitting your artificial data. Report the fitted parameters, 95% confidence intervals, and R^2 values for each fit, and compare mono- and biexponential models using an *F* ratio test, reporting both associated *F* and *p* values. What determines which datasets are easiest to fit well, and to distinguish between models for?

Now repeat the same exercise in the presence of noise; to do this, regenerate your fake data but add gaussian white noise with standard deviation 0.1 (see below for instructions). Report the same values as above, and explain how the addition of noise affected the results.

Generating noise for problem 3.
For problem 3 part “add noise,” you’ll notice that A) the time courses look horrible and B) nlinfit is EXTREMELY sensitive to initial parameter values. However, it’s still possible to get nlinfit to run with noise. Unfortunately, if you run your code again and generate NEW noise, your code might not run even though it did before. To combat this sensitivity, we will all use the same noise vectors so that you only have to find initial guesses that work once, then you can pipe your parameter estimates through the rest of your algorithm to get F and p values.

Please download the accompanying .mat file titled “Noise.mat”. Place this .mat file (Noise.mat) in whatever folder you run Matlab from. Then do the following:

1. Load noise using load(‘Noise.mat’)
    a. This will return a 101x3 vector titled *All*
2. During the “add noise to fluorescence data” part, add the predefined noise to your *Y* vector.
    a. Each column of the matrix *All* corresponds to one set of *k* values.
        i. For k_1=1 s^{-1} k_2=10 s^{-1} use Y_noise = Y + All(:, 1)
            1. If your Y vector is a row vector
               Y_noise = Y+transpose(All(:, 1))
        ii. For k_1=10 s^{-1} k_2=20 s^{-1} use Y_noise = Y + All(:, 2)
        iii. For k_1=10 s^{-1} k_2=100 s^{-1} use Y_noise = Y + All(:, 3)

This doesn’t change the sensitivity of the algorithm to your initial guesses. You’ll have to play with those values until your code runs without crashing.

* Adapted from K. D. Wittrup & B. Tidor, *Molecular and Cellular Bioengineering*