Computer Lab 7 – Deterministic and Stochastic Modeling

One of the themes in this course is the comparison of two ways to consider the structure and function of proteins. On the one hand, we have a single average structure representing the population of protein species and we consider that this average structure uniformly changes conformation while carrying out its function.

On the other hand, we have a distribution of protein species in various different energy states and conformations; with the states sampled according to their Boltzmann energies; and the functionally competent state of the protein selected by random, diffusional interaction with its substrate/ligand. We have learned how to use simulations of models to create the population that defines a Boltzmann probability distribution of states. The Boltzmann state average structure is the same as the X-ray diffraction crystal structure.

When one does the modeling and simulation of biochemical reactions or processes there is an analogous dichotomy. Consider an enzymatic reaction such as shown in scheme (1) below,

$$\mathrm{E} + \mathrm{S} \underset{k_{-1}}{\overset{k_{+1}}{\rightleftharpoons}} \mathrm{ES} \underset{k_{-2}}{\overset{k_{+2}}{\rightleftharpoons}} \mathrm{EP} \underset{k_{-3}}{\overset{k_{+3}}{\rightleftharpoons}} \mathrm{E} + \mathrm{P} \qquad (1)$$

where E = enzyme, S = substrate, ES is the initial enzyme-substrate complex, EP is the complex after conversion of substrate to product in the active site and P is the product. Each forward and reverse reaction is associated with its respective rate constant, k, where rate constants are used in connection with concentrations of reacting species; volume is not considered.

From the reaction scheme one can model the process using a set of differential equations containing the rate constants and reacting species and solve these with respect to time. The results will give one the varying concentrations of all reacting species with time. We assume that all such enzyme reactions in a specified volume proceed with the same average rate(s). This approach also assumes that the time evolution of an enzymatic reaction is both continuous and deterministic. It describes the behavior of an "average" enzyme. However, such a reaction is not continuous; it depends on individual molecules diffusing and colliding. There will be fluctuations in the molecular populations in sub-volumes of the total volume and these fluctuations may be important to the biochemical process.

Another approach to modeling and simulating a biochemical reaction is to use a discrete, stochastic process instead of a continuous, deterministic (average) process. With a stochastic simulation one can evaluate the random nature of molecular collisions. Reactions are treated in a probabilistic way, similar to the interactions of a Boltzmann probability distribution of protein states. *If a stochastic simulation of the same system is run multiple times the average of the results should agree with the deterministic results.*

In this lab, you will model and simulate an enzyme reaction using the MATLAB application for the **deterministic method** and the COPASI application for the **stochastic method** and you will compare the results from the two methods.

Answer the questions highlighted in yellow and send the answers to **achin14@jhu.edu** either in the body of an email or as an attached file with your JHEDID in the name (e.g. *JHEDID_lab7.txt* if made with vi or *JHEDID_lab7.docx* if made with MSWord).

A. Deterministic Modeling - MATLAB

MATLAB is an integrated numerical computing environment. The name, MATLAB, stands for MATrix LABoratory, a reflection of the program's origins. Indeed, one of the most distinctive features of MATLAB is its integrated use of matrices and the operations of linear algebra. We will use MATLAB to solve a set of ordinary differential equations that describe the reactions of an enzyme catalyzed reaction. MATLAB can be used through a graphical user interface (GUI) or as an environment for executing computer language code. The MATLAB scripting code is similar to Python in syntax and we will interact with MATLAB mainly through the scripting language rather than the GUI.

For reasons that will become clear later we will assume that once a product molecule is released from the enzyme that dissociation is irreversible. This, is not literally true, but is a useful assumption at early times in an enzyme catalyzed reaction. So, the reaction scheme we will model is not that described above in scheme (1), but rather the scheme below,

$$E + S \underset{k_{-1}}{\overset{k_{+1}}{\rightleftharpoons}} ES \underset{k_{-2}}{\overset{k_{+2}}{\rightleftharpoons}} EP \xrightarrow{k_{+3}} E + P \qquad (2)$$

The following ordinary differential equations can be used to compute the time dependent changes in species in reaction scheme (2), These are variously described as **rate expressions** or **differential rate laws**. More information on some related concepts in chemical kinetics can be found in the **Appendix** at the end of this document.

$$d[E]/dt = - k_{+1}[E][S] + k_{-1}[ES] + k_{+3}[EP] \qquad (3)$$
$$d[S]/dt = - k_{+1}[E][S] + k_{-1}[ES] \qquad (4)$$
$$d[ES]/dt = k_{+1}[E][S] - k_{-1}[ES] - k_{+2}[ES] + k_{-2}[EP] \qquad (5)$$
$$d[EP]/dt = k_{+2}[ES] - k_{-2}[EP] - k_{+3}[EP] \qquad (6)$$
$$d[P]/dt = k_{+3}[EP] \qquad (7)$$

These equations represent a set of five equations with five unknowns and the values for the differentiated elements can be solved by providing values for the initial concentrations and individual rate constants and using an algorithm in MATLAB. 1. What is the difference between ordinary and partial differential equations?

We start the simulation with initial conditions of [S]=50 µmolar, [E]=1 µmolar, [ES]=0, [EP]=0, [P]=0; define the rate constants; and observe how the variables [S], [ES], [EP] and [P] change with time. Again, this type of modeling assumes that all substrate molecules and all enzyme molecules are identical and that the reaction proceeds according to an analytical, homogeneous average description. This is called **deterministic** modeling because the exact same progress will be observed for different runs of the simulation if the initial conditions are identical. 2. What part of the reaction scheme determines that the reaction goes to 100% completion rather than some equilibrium ratio of *[S]/[P]*?

1. Deterministic Simulation

Fetch the *DetStoch_files.tar* tarball from */home/compbio2/Shared/* on kirin; unpack it and move the new *DetStoch_files/* directory to your home directory or flash drive. Launch MATLAB from the Applications folder by clicking on the MATLAB_R2015b application. Click the change-folder icon,  , navigate to the *DetStoch_files/* directory and open it. You should now see a list of files in this directory displayed in the left window of MATLAB. Open the two files, *Enz_progress.m* and *Enz_progFunc.m* in the MATLAB editor by double clicking on them. Use the MATLAB editor to code in the above differential equations in *Enz_progFunc.m* under the `%Differential equations` comment line. In MATLAB one cannot use +/- in the name of a variable so we use `kp1` for k_{+1}, `km1` for k_{-1}, etc. Use the following syntax to enter the differential equations:

```
dE= -kp1*E*S + km1*ES +kp3*EP;
```

I.e., no brackets and do not include the /d*t*. Don't forget to put a semicolon at the end of each line. Save the file.

Inspect the *Enz_progress.m* file in the MATLAB editor. 3. What are the initial concentrations of substrate (`S0`) and enzyme (`E0`)? The command, `t=linspace(0, tmax, N);` may not be familiar to you. We want to create a vector of time values to pass to the MATLAB solver; these are the time steps at which the differential equations will be solved. The command means: Create a vector of values from 0 to tmax with intervals of N. Each time interval will be 5 seconds (3000/600). The next command, `y0=[E0;S0;ES0;EP0;P0];`, creates a vector containing our initial concentrations so that we can pass that along with the time vector to the MATLAB solver.

The next command, calls the solver (`ode23s`) and tells the solver that the differential equations are in the function file, `'Enz_progFunc'`, that the time steps are listed in the variable, `t`, and that the initial concentrations are listed in the variable, `y0`.

The rest of the code is for plotting and we won't worry about that now.

Run the *Enz_progress.m* file by clicking on the Run button in the menu bar at top. You should get a plot of the progress of [S], [ES]*10, [EP]*10 and [P] changing with time (note that the [ES] and [EP] have been multiplied by 10 to better visualize these components).

You should see that there is a rapid formation of [ES] (blue curve) and [EP] (cyan curve) and that the concentrations of these species remain at a steady state for approximately 1000-1500 seconds. As the [S] becomes low the [ES] and [EP] species decrease.

This plot represents the average results of a specified enzymatic reaction in some volume containing certain concentrations of reactants. You may remember from biochemistry that the assumption that [ES] is at steady state is important in the derivation of the Michaelis-Menten equation.

B. Stochastic Modeling - COPASI

COPASI (Complex PAthway SImulator) is a computer application that can be used through either a GUI or a command line. We will use COPASI through the GUI. The actual scripting code behind the COPASI GUI is in the systems biology markup language (SMBL). SMBL is a machine-readable format for representing models; it is not meant to be edited by humans as is the case for MATLAB code.

In COPASI a model is defined by enumerating each possible individual reaction in the complete reaction scheme rather than by a set of differential equations. So, the above enzyme reaction scheme (2) would be modeled by the following set of individual reactions.

S + E -> ES	(SBinding, reaction constant = c_{+1})	(8)
ES -> S + E	(SDissociation, reaction constant = c_{-1})	(9)
ES -> EP	(SConversion, reaction constant = c_{+2})	(10)
EP -> ES	(PReversion, reaction constant = c_{-2})	(11)
EP -> P + E	(PDissociation, reaction constant = c_{+3})	(12)

Each of the above reactions has an associated stochastic reaction constant (c) that depends only on the properties of the reacting species and the temperature of the system. The stochastic reaction constant (c) is analogous to a rate constant (k) but is proportional to the *average probability that a particular set of reactant molecules will react during a given time step*. Consider a volume containing X substrate molecules and Y enzyme molecules. We say that $XYc_{+1}dt$ = the average probability that two molecules, S and E, will react in the next infinitesimal time interval dt. The value of c defines the average of a reaction probability density function. We use a random number generator to choose a value from the reaction probability density function and assign it to a specific reaction probability at each iteration; after many iterations, the average reaction probability is equal to c. In practice, for a second order reaction such as (8) above, $k \approx cV$ (where V = volume) because the deterministic equations use molecular concentration whereas the stochastic equations use total numbers of molecules. (The theoretical relationship between c and k is more complicated and if you are curious see the paper by Gillespie linked below.)

As stated above, in conjunction with the use of stochastic reaction constants one considers the *number* of molecules in a reaction volume rather than the *concentration* of molecules. COPASI has the ability to go back and forth between these two unit systems so it will be convenient to enter species concentrations and let the program convert to numbers per volume. Volume becomes important in stochastic simulations as you will see.

The time evolution of the stochastic model can be calculated using a method devised by Daniel Gillespie in the 1970's. COPASI implements the Gillespie algorithm (called Stochastic (Direct method) in the program). This (famous) algorithm is described in his 1977 paper and is available on the course website;
pages.jh.edu/~pfleming/compbio/files/gillespie_jphyschem_1977.pdf.

1. Stochastic Simulation
Launch COPASI from the Applications folder by clicking on the CopasiUI application. From the File | Open menu, browse to your *DetStoch_files/* folder and open the *Enz_progress.cps* file. The first screen you should see is the **Model** page. Notice that the units include µmol for concentrations, time is in seconds, and volume is in liters. The model has been partially created for you with these units and we will inspect the some of the parameters now.

Expand the Model submenu on the left-hand side of the screen by clicking the arrowhead. You should see three new options indented under Model: Biochemical, Mathematical, and Diagrams. Click on the arrowhead next to Biochemical to open up some new options. Click on the arrowhead next to Compartments and then click on the Cell item. Notice the Initial Volume is 1e-17 liters. A typical bacterial cell is ~1e-15 liters so we will be simulating a small compartment of a bacterial cell or a sub-compartment of a eukaryotic organelle.

Click on the arrowhead next to Species to see the list of molecular species in the model. Inspect the S species row. Notice that we set the initial concentration as 50 µmol (this means µmolar, same as in the deterministic simulation). At the top of the COPASI GUI change the Concentrations menu item to Particle Numbers. Notice that 50 µmolar equates to only ~301 substrate molecules in 1e-17 liters. Changing the volume will change the number of reactants in a stochastic simulation.

Instead of writing differential equations that describe the progress of [S], [ES], etc., with stochastic modeling we list each reaction that a reactant species is involved in within the model. For the substrate, there are two reactions; binding to, and dissociation from, the enzyme:

S + E -> ES
ES -> S + E

Inspect the E species row. 4. About how many enzyme molecules are present in the simulation? Similar lists of reactions are entered for each of the species (ignore the species with the word "Tag" or "_10" in their name for now).

Next, click on the Reactions menu item (Click on the word, not just the arrowhead). This will bring up a new screen that shows some existing reactions (the ones with Tag in the name that you can ignore for now). This is where you create your actual reactions. There are five reactions you will want to create in this model as described in reaction schemes (8-12) above. Click on New Reaction and change its name to **`SBinding`**. Expand the Reactions option. Click on the new, indented SBinding item that appears under Reactions. This will bring up a new screen. Type **`S + E -> ES`** in the space next to Reaction (include spaces) and enter **<return>**. COPASI automatically assigns the reaction the rate law Mass action (irreversible). Go down to where it says Parameter. Under the Mapping dropdown menu, select cp1 (for c_{+1}). Then click on the Commit button at the bottom.

Now repeat the procedure described in the above paragraph for reactions (9-12) above using the reaction labels: **`SDissociation`**, **`SConversion`**, **`PReversion`**, **`PDissociation`**. Map the appropriate stochastic reaction constant (c_{+1}=cp1, c_{-1}=cm1, etc.) to each reaction.

Click on the Global Quantities menu item (the name) and make sure the values for each rate constant are the same as in the deterministic *Enz_progress.m* file.

Finally, expand the Tasks menu and click on the Time Course item (the word, not just the arrowhead). On the Time Course page, you can see that we have decreased the time Interval Size to 0.1 seconds (compared to 5 s in the MATLAB simulation) so that we can better delineate individual stochastic events during the simulation. Click the Run button at the bottom of the Time Course page. Nothing seems to happen!

Actually, the simulation was run and an output file called, *TaggedParticle.txt*, was created in your pwd. There were, in fact, ~302 molecules of substrate in your simulation (not just ~301); one was a "tagged" molecule and although it competed with the other 301 substrate molecules for binding to the enzyme molecules we kept track of its binding and conversion to product. The time evolution of the tagged particle state is reported in the *TaggedParticle.txt* file. Inspect the contents of this file. It contains five columns: `Time` (in seconds), and then four columns that are either "`1`" or "`0`". A "`1`" indicates that your tagged particle was present in the state indicated by each column: S, ES, EP, P (where the labels are unfortunately long, e.g., TagS.ParticleNumber = S).

To view the time evolution of your tagged substrate, (i.e. when did it bind to E to form ES; when did it get converted to P on the enzyme; and when did it dissocitate to free P) do the following: In MATLAB open the Tag_movie.m in the MATLAB editor. Read the annotation to see what this MATLAB script does. For example, we want to plot time (*x* variable) versus one of four states (*y* variable). 5. What number is assigned to the *y* variable for state P? Click the green Run arrowhead at the top of the editor and watch (patiently) what appears on your screen. You should see a red circle appear in a plot that indicates whether the tagged molecule is in state ES or EP at each time point during the simulation. Wait for the blue lines to appear at the end of the movie. We eliminated the possibility of P rebinding to E so that the progress of an individual molecule would end with formation of product; once a substrate is converted to P it stays in that state. This

arbitrary decision will also make it easier to compare deterministic and stochastic results as described below.

Each simulation will be different but you may see multiple binding events to EnzS, some conversions to EnzP, many reversions to EnzS and dissociations back to Subs before a final EnzP dissociates to Prod. Close the plot, go back to COPASI, click on the Time Course menu item and click Run again. This will overwrite the *TaggedParticle.txt* file with output from the current simulation. Run the Tag_movie.m in MATLAB again. Satisfy yourself that each stochastic simulation of a tagged substrate is different.

Now you will look at the average time evolution of all 301 substrate molecules in your stochastic simulation. In the COPASI menu list on the left expand the Output Specifications item and then expand the Plots item. Click on Plot 1 and then on the window that appears click the active? button in the upper right corner to activate the plot; click Commit. Click on the Time Course item in the menu list and Run the simulation again. You should see a plot appear with the number of substrate molecules (S) represented by a green curve, ES *10 in blue, EP *10 in cyan, and product molecules (P) in red. These curves represent the progress of the reaction resulting from the random reactions occurring with all 301 substrate molecules and 6 enzyme molecules in a volume of 1e-17 liters.

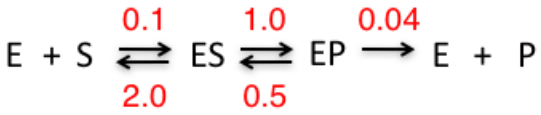
Compare this stochastic progress plot with the deterministic plot obtained using the *Enz_progress.m* script in MATLAB (run it again if you closed the deterministic plot). Are they identical? The stochastic curves are not as smooth as the deterministic curves because of the randomness of the reactions. 6. Why do the ES and EP curves alternate between zero and some value? You can get a better idea of the variability of the stochastic process by plotting multiple simulations on the same plot. Do the following: On the left menu list of the COPASI window click the Parameter Scan item. A new page should appear that has a window indicating 10 iterations. Click the Run button on this **Parameter Scan** page. Now your stochastic plot should have 10 separate simulation curves on it. If you averaged the data in these curves you should get curves identical to the deterministic plot but it's hard to tell if this is true just from inspection of this plot.

Another way to test whether or not your stochastic simulation agrees with your deterministic simulation is to increase the volume of the stochastic simulation to improve the N for averaging. If we increase the volume 100x we should observe the progress of our reaction starting with *~30,000 substrate molecules*. At the top of the COPASI GUI make sure that you have Concentrations instead of Particle Numbers. In the left menu window go to Model | Biochemical | Compartments | Cell and click on the Cell item. Change the Initial Volume to **`1e-15`** liters and click the Commit button.

Now change back to Particle Numbers, click on Model | Biochemical | Species | S. 7. How many substrate molecules are you starting with now?

Click the Time Course menu item and run the simulation. 8. Does your plot recapitulate the plot from the deterministic simulation?

Below is reaction scheme (2) that you simulated with the values for the rate/reaction constants in red.



9. From a comparison of the relative rate constants creating and eliminating ES versus the creation and elimination of EP, which species should have the longer lifetime? Let's plot the distributions of lifetimes for these species to find out if your answer is correct.

We will use MATLAB to plot the lifetimes of species during the COPASI simulation. This is simply a matter of binning and plotting the distributions of how many consecutive "1" there are in the third and fourth columns of *TaggedParticle.txt*. To obtain enough data to have meaningful distributions we will collect data from 50 separate COPASI runs in one output file. MATLAB has memory problems dealing with such a large file so you will extract the relevant columns (3 & 4) using your UNIX skills.

Change the COPASI simulation volume back to 1e-17 liters. Go to Output Specifications | Plots | Plot 1 and uncheck the active? box; click Commit. Go to the Tasks | Parameter Scan page and change the **Repeat** | Number of iterations to **50**. Then click the Run button. Wait until the simulation finishes and then in a terminal window open to the *DetStoch_files* directory, use your UNIX skills to determine how many lines are in the *TaggedParticle.txt* file. If the number is less than 1.5 million lines call the instructor or TA.

The *TaggedParticle.txt* file now contains 50 simulation output blocks and each begins with the column labels. We have to ignore the column labels and extract columns 3 and 4 for binning and plotting. The UNIX command to extract all the lines in a file that DO NOT contain the word `Time` is

```
grep –v 'Time' TaggedParticle.txt
```

Pipe the output of this command into an awk command that prints column 3 to a file called *ES.dat*. Also, extract column 4 to a file called *EP.dat*. 10. Enter the complete commands you used in your lab report.

Now run the files *Life_time_ES.m* and *Life_time_EP.m* in MATLAB. 11. From inspection of the plots that appear, and values that are printed to the command window, which species, ES or EP, has the longer lifetime?

This concludes your comparison of deterministic and stochastic simulations of biochemical processes. Usually deterministic simulations are much faster than stochastic simulations but one can obtain more detailed information about reacting species in stochastic simulations. Which method to use depends on your question.

Note that in both simulations today we assumed that the components in the volumes were "well stirred". There was no spatial information or local depletion of solutes – this is another level of detail that one can add to reaction modeling.

Acknowledgement: Thanks to Aaron Chum for the original idea and design of this lab.

Appendix – Simple chemical kinetics definitions.

The **overall reaction** for the process described by reaction scheme (2) is,

$$S \rightarrow P$$

The **elementary steps** in the reaction (at the level of chemical kinetics we are concerned with) are described by reaction scheme (2),

$$E + S \underset{k_{-1}}{\overset{k_{+1}}{\rightleftharpoons}} ES \underset{k_{-2}}{\overset{k_{+2}}{\rightleftharpoons}} EP \xrightarrow{k_{+3}} E + P$$

Each of these elementary steps is a result of molecular collisions. Therefore, the rate of each step will be proportional to the concentrations of reacting species. For example, the production of $P \propto [EP]$ and the *initial* production of $ES \propto [E][S]$.

We can write that the rate of production of $P = d[P]/dt = k_{+3}[EP]$ and the *initial* rate of production of $ES = d[ES]/dt = k_{+1}[E][S]$, where $\boldsymbol{k_{+3}}$ and $\boldsymbol{k_{+1}}$ are **rate constants** for the respective reactions.

The overall order of a reaction is equal to the number of molecules reacting in an elementary step. The formation of P, above, is dependent on one species, [EP] and is said to be **first order** in EP; therefore, the rate constant k_{+3} is a **1st order rate constant** with units of ((concentration time^{-1})/concentration) = time^{-1} (usually per second, $\boldsymbol{s^{-1}}$).

The initial formation of ES, above, is dependent on two species, [E][S], and is said to be **second order**; therefore, the rate constant k_{+1} is a **2nd order rate constant** with units of ((concentration time^{-1})/concentration^2) = concentration^{-1} time-1(usually per molar per second, $\mathbf{M^{-1}}$ $\boldsymbol{s^{-1}}$).

Rate equations may be written in different forms.

For a first order reaction (e.g. $A \rightarrow B$),

Differential form: $$-\frac{d[A]}{dt} = k[A]$$

Integrating over concentration and time: $$\int_{[A]o}^{[A]t} \frac{d[A]}{[A]} = -k \int_o^t dt$$

Yields the integrated form: $$\ln \frac{[A]t}{[A]0} = -kt \quad \text{or} \quad [A]_t = [A]_o \exp^{-kt}$$

For a second order reaction (e.g. $2A \rightarrow B$),

Differential form: $-\frac{d[A]}{dt} = k[A]^2$

Integrating over concentration and time: $\int_{[A]o}^{[A]t} \frac{d[A]}{[A]^2} = -k \int_o^t dt$

Yields the integrated form: $-\frac{1}{[A]t} + \frac{1}{[A]o} = -kt$ or $\frac{1}{[A]t} = \frac{1}{[A]o} + kt$

Both the half-life ($t_{1/2}$) and lifetime (τ) may be used to characterize how long a species will last during the reaction.

To define the **half-life**, consider that at,

$t = t_{1/2}$, $[A]_t = \frac{1}{2}[A]_o$

Substituting into the integrated rate equations yields,

First order: $t_{1/2} = (\ln 2)/k = 0.693/k$

Second order: $t_{1/2} = 1/[A]_o k$

The **lifetime** (τ, also called **residence time**) for a first order process is usually defined as the time it takes for the species concentration to fall to $1/e$ of its initial value (where $e = 2.718$). This result may be arrived at by starting with an alternate definition of τ,

$$\tau = \frac{\text{amount}}{\text{rate}} = \frac{[A]}{k[A]} = \frac{1}{k}$$

where rate is the rate of removal. Substituting the lifetime τ, for t into the integrated rate equation above for a first order reaction yields, $\ln [A]_t/[A]_o = -k(1/k) = -1$. So, at $t = \tau$ (i.e., $1/k$),

$[A]_t/[A]_o = e^{-1} = 1/e = 0.37$

Note that once a species (such as ES) is formed its duration is controlled by a stochastic removal process. To calculate the lifetime of a species such as ES in our stochastic simulation of the enzyme reaction scheme, one would tally all the instances of existence and how long each instance lasted (duration). These data are binned and plotted as number of instances of some duration *versus* duration. This should give an exponentially decreasing curve that can be fit to an equation of the form, $[ES]_t = [ES]_o e^{-kt}$ (where t = duration time). This assumes pseudo-first order kinetics controlled by a dominant rate constant, see below. From the fit value of k, the lifetime (τ) may be calculated.

Note also that,

$$t_{1/2} = 0.693 \text{ x } \tau$$

For the overall lifetime of a species that is removed by several independent processes such as the following,

$$B \overset{k1}{\leftarrow} A \overset{k2}{\rightarrow} C$$

the independent contributions to the lifetime of A are,

$$\tau_1 = \frac{[A]}{k_1[A]} \quad \text{and} \quad \tau_2 = \frac{[A]}{k_2[A]}$$

However, the overall lifetime of a species is

$$\tau = \frac{amount}{\text{rate1+rate2}} = \frac{1}{\sum k}$$

where rate1 and rate2 are the independent rates of removal. Thus,

$$\frac{1}{\tau} = \frac{1}{\tau_1} + \frac{1}{\tau_2}$$

If $\tau_1 >> \tau_2$, removal process 2 is more effective than process 1 and τ will essentially be equal to τ_2. If there are competing removal processes, use the fastest removal rate (shortest lifetime) as the estimate for the lifetime of a species. This, of course, is only valid if one removal rate is significantly greater than all others. If this is not the case then the duration plot described above will need to be fit to a multiple exponential equation.

(Note that we can define a lifetime for a species in a second order reaction ($2A \rightarrow B$ or $A+B \rightarrow C$) as,

$$\tau = \frac{\text{amount}}{\text{rate}} = \frac{[\text{A}]}{\text{k}[A]^2} = \frac{1}{\text{k}[\text{A}]} \quad \text{or} \quad \tau = \frac{\text{amount}}{\text{rate}} = \frac{[\text{A}]}{\text{k}[\text{A}][B]} = \frac{1}{\text{k}[\text{B}]}$$

but this is seldom used.)