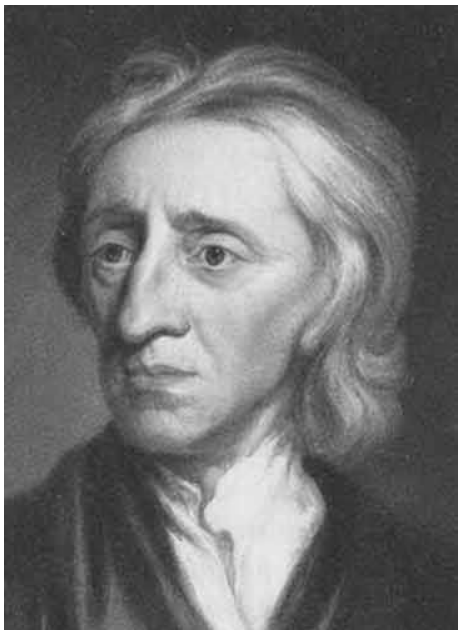
Engineering Genetic Circuits

Chris J. Myers

Lecture 7: Stochastic Analysis

John A. Locke



That which is static and repetitive is boring. That which is dynamic and random is confusing. In between lies art.

Richard Feynmann

A philosopher once said "It is necessary for the very existence of science that the same conditions always produce the same results." Well, they do not. You set up the circumstances, with the same conditions every time, and you cannot predict behind which hole you will see the electron.

Problems with Reaction Rate Equations

- Chemical reaction network model can be tranformed using law of mass action into ODEs known as reaction rate equations.
- Assume concentrations vary continuously and deterministically.
- Chemical systems satisfy neither of these assumptions.
- Number of molecules of a species is a discrete quantity.
- Chemical reactions occur after two molecules collide.
- Unless track exact position and velocity of every molecule, not possible to know when a reaction may occur.
- Should consider occurence of reactions to be stochastic.

Stochastic Process Description

- For systems which involve large molecular counts, ODE models give accurate picture of their behavior.
- If molecular counts are small, discrete and stochastic nature may have significant influence on observed behavior.
- Genetic circuits typically involve small molecule counts.
- Often only one strand of DNA and a few 10s or 100s of molecules of each transcription factor.
- Accurate analysis requires a stochastic process description.
- This lecture presents one such description, the *chemical master equation*, and algorithms to analyze it.

Overview

- Stochastic chemical kinetics
- The chemical master equation
- Gillespie's stochastic simulation algorithm
- Gibson/Bruck's next reaction method
- Tau-leaping
- Relationship to reaction rate equations
- Stochastic Petri-nets
- Phage λ decision circuit example
- Spatial Gillespie

A Stochastic Chemical Kinetic Model

- Composed of n chemical species $\{S_1, \ldots, S_n\}$ and m chemical reaction channels $\{R_1, \ldots, R_m\}$.
- Assume species contained within constant volume Ω.
- Assume system is *well-stirred* to neglect spatial effects.
- Assume system is in *thermal equilibrium* (i.e., at a constant temperature), but not necessarily *chemical equilibrium*.

State Updates

- $X_i(t)$ is the number of molecules of S_i at time t.
- $\mathbf{X}(t) = (X_1(t), \ldots, X_n(t))$ is the state of a system at time t.
- $\mathbf{X}(t_0) = \mathbf{x}_0$ is initial number of molecules at initial time t_0.
- After R_μ, the new state is $\mathbf{x}' = \mathbf{x} + v_\mu$ where $v_\mu = (v_{1\mu}, \ldots, v_{n\mu})$ is the *state-change vector* and $v_{i\mu}$ is the change in S_i due to R_μ.
- The 2-dimensional array $\{v_{i\mu}\}$ is known as the *stoichiometric matrix*.
- R_μ is *elemental* if it can be considered a distinct physical event that happens nearly instantaneously.
- For elemental R_μ, values of $v_{i\mu}$ are constrained to $0, \pm 1, \pm 2$.

Specific Probability Rate Constant

- Every R_μ has a *specific probability rate constant*, c_μ, which is related to the reaction rate constant, k_μ.
- $c_\mu dt$ is the probability that a randomly chosen combination of reactant molecules react as defined by R_μ inside Ω in $[t, t+dt)$.
- Multiplying c_μ by the number of possible combinations of reactant molecules for R_μ in a state $\mathbf{x}$ yields the *propensity function*, a_μ.
- $a_\mu(\mathbf{x})dt$ is the probability that R_μ occurs in the state $\mathbf{x}$ within Ω in the next infinitesimal time interval $[t, t+dt)$.

A Bimolecular Reaction Channel

- A typical *bimolecular reaction channel* R_μ has form:

$$S_1 + S_2 \xrightarrow{c_\mu} S_3 + \ldots$$

- c_μ is probability that a S_1 molecule and S_2 molecule collide and react within next dt time units.
- Assume molecules hard spheres with masses m_i and radii r_i.
- Thermal equilibrium means that a selected S_i can be found uniformly distributed within Ω.
- Also means avg. relative speed in which S_2 sees S_1 moving is:

$$\overline{v}_{12} = \sqrt{8k_B T/\pi m_{12}}$$

where k_B is Boltzmann's constant and $m_{12} = m_1 m_2/(m_1 + m_2)$.

A Bimolecular Reaction Channel (cont)

- In next dt, S_2 molecule sweeps a collision cylinder relative to S_1 which has a height $\overline{v}_{12}dt$ and base area $\pi(r_1 + r_2)^2$.
- Probability that S_1 is within the collision cylinder is ratio of the cylinder's volume to Ω, so c_μ is:

$$c_\mu = \Omega^{-1}\pi(r_1 + r_2)^2\overline{v}_{21}p_\mu$$

  where p_μ is probability that S_1 and S_2 react when they collide.
- If we assume that S_1 and S_2 react only when their kinetic energy exceeds the activation energy, ε_μ, then c_μ is:

$$c_\mu = \Omega^{-1}\pi(r_1 + r_2)^2\left(\frac{8k_BT}{\pi m^*}\right)^{1/2} exp(-\varepsilon_\mu/k_BT).$$

A Bimolecular Reaction Channel (cont)

- Number of combinations of S_1 and S_2 molecules is $x_1 x_2$, so propensity function for R_μ is $a_\mu(\mathbf{x}) = c_\mu x_1 x_2$.
- If $S_1 = S_2$ then number of combinations is $x_1(x_1 - 1)/2$, and $a_\mu(\mathbf{x}) = c_\mu x_1(x_1 - 1)/2$.

Monomolecular Reactions

- *Monomolecular* reactions are of this form:

$$S_1 \xrightarrow{c_\mu} S_2 + \ldots$$

- S_1 makes a spontaneous change in its internal structure.
- c_μ must be found from quantum mechanical considerations.
- Propensity function is simply $a_\mu(\mathbf{x}) = c_\mu x_1$.
- If it is actually an enzymatic reaction of the form:

$$E + S_1 \xrightarrow{c_\mu} E + S_2 + \ldots$$

where E is an enzyme, should be considered as a bimolecular reaction.

Trimolecular Reactions

- *Trimolecular* reactions are of this form:

$$S_1 + S_2 + S_3 \xrightarrow{c_\mu} S_4 + \ldots$$

- Probability is very small, so typically used as approximation for:

$$S_1 + S_2 \underset{c_2}{\overset{c_1}{\rightleftarrows}} S^* \text{ and } S^* + S_3 \xrightarrow{c_3} S_4 + \ldots$$

- This approximation is reasonable when the lifetime of S^* is very short (i.e., $1/c_2$ is very small).
- The probability that a molecule of S^* reacts with a randomly chosen molecule of S_3 is approximately $c_3(1/c_2)$.

Trimolecular Reactions (cont)

- Consider a small but finite time interval Δt which is still much larger than the lifetime of S^* (i.e., $\Delta t >> 1/c_2$).
- If Δt is sufficiently small, then the probability that S_1 and S_2 react in that time interval to form S^* is $c_1 \Delta t$.
- Probability of both reactions occuring in Δt is $(c_1 c_3/c_2)\Delta t$, so c_μ for the trimolecular reaction approximation is:

$$c_\mu \quad = \quad c_1 c_3 / c_2$$

- Approximation because Δt is not a true infinitesimal.
- Propensity function for this reaction is $a_\mu(\mathbf{x}) = c_\mu x_1 x_2 x_3$.

Relationship Between c_μ and k_μ

- For bimolecular reactions, c_μ is proportional to Ω^{-1}.
- For monomolecular reactions, it is independent of volume.
- For trimolecular reactions, it is proportional to Ω^{-2}.
- In general, if m is the number of reactant molecules in R_μ:

$$c_\mu \quad \propto \quad \Omega^{-(m-1)}$$

- Key to understanding relationship between c_μ and k_μ.
- For monomolecular reactions, c_μ is equal to k_μ.
- For bimolecular reactions, c_μ is equal to k_μ/Ω if the reactants are different species and $2k_\mu/\Omega$ if the same species.

Jump Markov Processes

- Stochastic model is a *jump Markov process*.
- A Markov process is one where the next state is only dependent on the present state and not the past history.
- A jump Markov process is one in which the state updates occur in discrete amounts.

Time Evolution of Probability

- Not possible to know the exact state $\mathbf{X}(t)$.
- Only can know probability of being in a given state at time t starting from a state $\mathbf{X}(t_0) = \mathbf{x}_0$ (i.e., $\mathcal{P}(\mathbf{x}, t|\mathbf{x}_0, t_0)$).
- Probability using a time-evolution of step dt is shown below:

$$\begin{aligned}\mathcal{P}(\mathbf{x}, t+dt|\mathbf{x}_0, t_0) &= \mathcal{P}(\mathbf{x}, t|\mathbf{x}_0, t_0) \times \left[1 - \sum_{j=1}^{m} (a_j(\mathbf{x})dt)\right] \\ &+ \sum_{j=1}^{m} \mathcal{P}(\mathbf{x} - v_j, t|\mathbf{x}_0, t_0) \times (a_j(\mathbf{x} - v_j)dt)\,.\end{aligned}$$

- dt is small enough that at most one reaction occurs during dt.

Chemical Master Equation

- *Chemical master equation* (CME) defines time evolution of state probabilities, $\mathcal{P}(\mathbf{x}, t|\mathbf{x}_0, t_0)$:

$$\begin{aligned}\frac{\partial \mathcal{P}(\mathbf{x}, t|\mathbf{x}_0, t_0)}{dt} &= \lim_{dt \to 0} \frac{\mathcal{P}(\mathbf{x}, t+dt|\mathbf{x}_0, t_0) - \mathcal{P}(\mathbf{x}, t|\mathbf{x}_0, t_0)}{dt} \\ &= \sum_{j=1}^{m} [a_j(\mathbf{x} - v_j)\mathcal{P}(\mathbf{x} - v_j, t|\mathbf{x}_0, t_0) \\ &\quad - a_j(\mathbf{x})\mathcal{P}(\mathbf{x}, t|\mathbf{x}_0, t_0)]\end{aligned}$$

- Typically cannot be solved analytically since it represents a set of equations as large as the number of molecules in the system.

Stochastic Simulation

- Trajectories for $\mathbf{X}(t)$ can be generated using *stochastic simulation*.
- Could pick a small time step *dt* and at each step update the system state by selecting a reaction to occur or doing nothing.
- For a sufficiently small *dt*, however, the vast majority of time steps result in no reaction.

Gillespie's Stochastic Simulation Algorithm

- *Gillespie's stochastic simulation algorithm* (SSA) improves the efficiency of simulation by stepping over useless time steps.
- Not based directly on CME, but equivalent form that uses $p(\tau, \mu | \mathbf{x}, t)$.
- Defined such that $p(\tau, \mu | \mathbf{x}, t) d\tau$ is probability that the next reaction is R_μ which occurs in $[t+\tau, t+\tau+d\tau]$ assuming current state is $\mathbf{X}(t) = \mathbf{x}$.
- This is a joint PDF for two random variables, τ and μ given that the system is in state $\mathbf{x}$ at time t.
- Simulation advances from one reaction to the next skipping over time points in which no reaction occurs.

Derivation of Gillespie's SSA

- Introduce $\mathcal{P}_0(\tau|\mathbf{x},t)$ that represents probability that there is no reaction in the time interval $[t,t+\tau]$.
- $p(\tau,\mu|\mathbf{x},t)$ defined as follows:

$$p(\tau,\mu|\mathbf{x},t)d\tau \quad = \quad \mathcal{P}_0(\tau|\mathbf{x},t) \times (a_\mu(\mathbf{x})d\tau)\,. \tag{1}$$

- No reactions occur in the interval $[t,t+\tau)$ and the R_μ reaction occurs in the interval $[t+\tau,t+\tau+d\tau]$.

Derivation of Gillespie's SSA (cont)

- The function $\mathcal{P}_0(\tau|\mathbf{x},t)$ must satisfy the following:

$$\mathcal{P}_0(\tau+d\tau|\mathbf{x},t) \quad = \quad \mathcal{P}_0(\tau|\mathbf{x},t) \times \left[1-\sum_{j=1}^{m}(a_j(\mathbf{x})d\tau)\right].$$

- Using this formula, get following differential equation:

$$\frac{d\mathcal{P}_0(\tau,\mathbf{x},t)}{d\tau} = -a_0(\mathbf{x})\mathcal{P}_0(\tau|\mathbf{x},t) \quad \text{where} \quad a_0(\mathbf{x}) = \sum_{j=1}^{m} a_j(\mathbf{x}).$$

- With $\mathcal{P}_0(\tau=0|\mathbf{x},t)=1$, has following solution:

$$\mathcal{P}_0(\tau|\mathbf{x},t) \quad = \quad exp(-a_0(\mathbf{x})\tau). \tag{2}$$

Derivation of Gillespie's SSA (cont)

- Inserting Equation 2 into Equation 1 and canceling $d\tau$ yields:

$$p(\tau,\mu|\mathbf{x},t) \quad = \quad exp(-a_0(\mathbf{x})\tau) \times a_\mu(\mathbf{x}),$$

  which can be rewritten as:

$$p(\tau,\mu|\mathbf{x},t) \quad = \quad a_0(\mathbf{x})exp(-a_0(\mathbf{x})\tau) \times \frac{a_\mu(\mathbf{x})}{a_0(\mathbf{x})}.$$

- $p(\tau,\mu|\mathbf{x},t)$ can be divided into PDFs for τ and μ.
- τ is exponential random variable with mean and std dev of $\frac{1}{a_0(\mathbf{x})}$.
- μ is integer random variable with point probabilities $\frac{a_\mu(\mathbf{x})}{a_0(\mathbf{x})}$.

Gillespie's SSA (Direct Method)

1. Initialize: $t = t_0$ and $\mathbf{x} = \mathbf{x}_0$.
2. Evaluate $a_j(\mathbf{x})$ and $a_0(\mathbf{x}) = \sum_{j=1}^{m} a_j(\mathbf{x})$.
3. Draw two unit uniform random numbers, r_1 and r_2.
4. Determine the time, τ, until the next reaction:

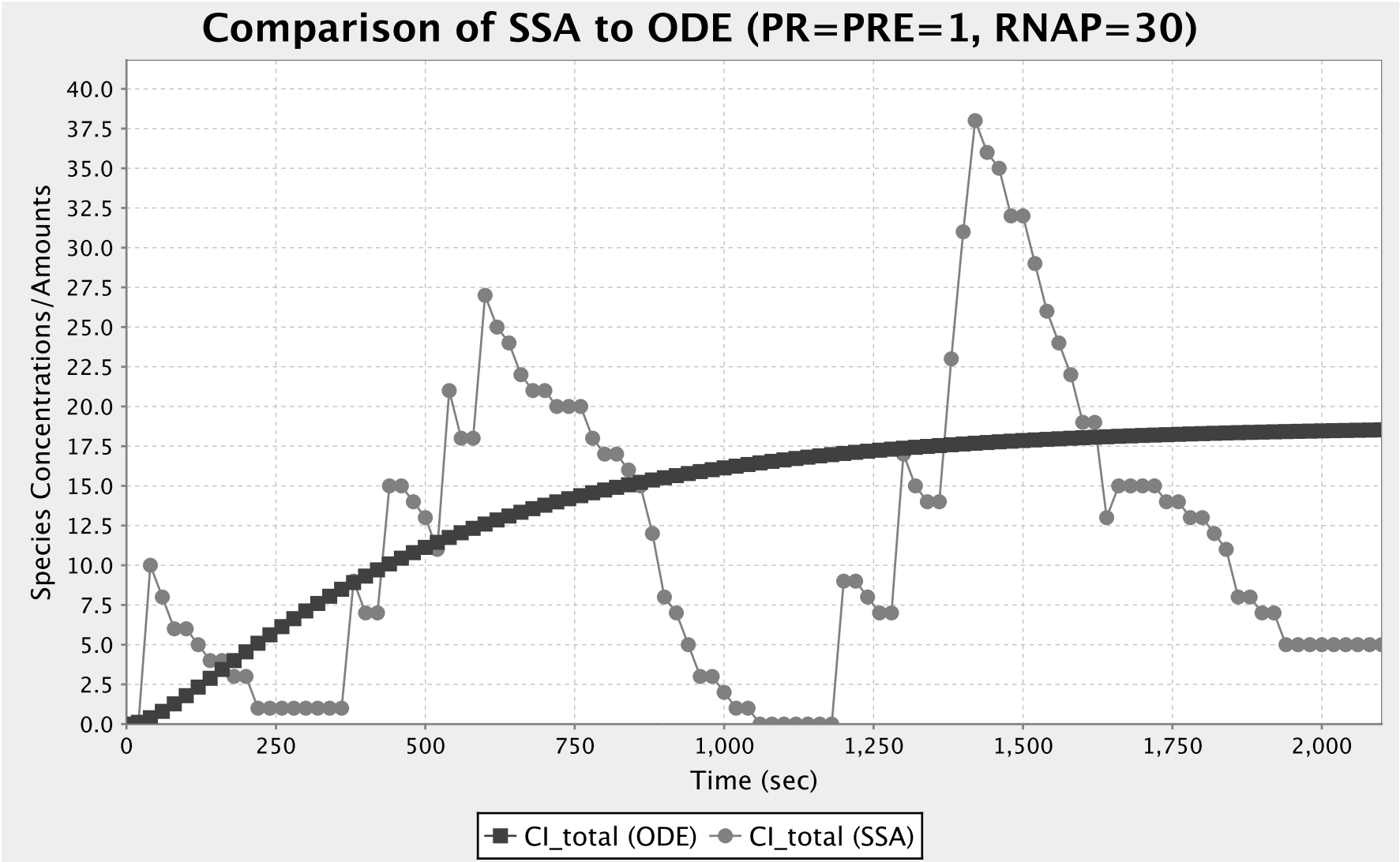
$$\tau = \frac{1}{a_0(\mathbf{x})} \ln\left(\frac{1}{r_1}\right).$$

5. Determine the next reaction, R_μ:

$$\mu = \text{ the smallest integer satisfying } \sum_{j=1}^{\mu} a_j(\mathbf{x}) > r_2 a_0(\mathbf{x}).$$

6. Determine the new state: $t = t + \tau$ and $\mathbf{x} = \mathbf{x} + v_\mu$.
7. If t is greater than the desired simulation time then halt.
8. Record $(\mathbf{x}, t)$ and goto step 2.

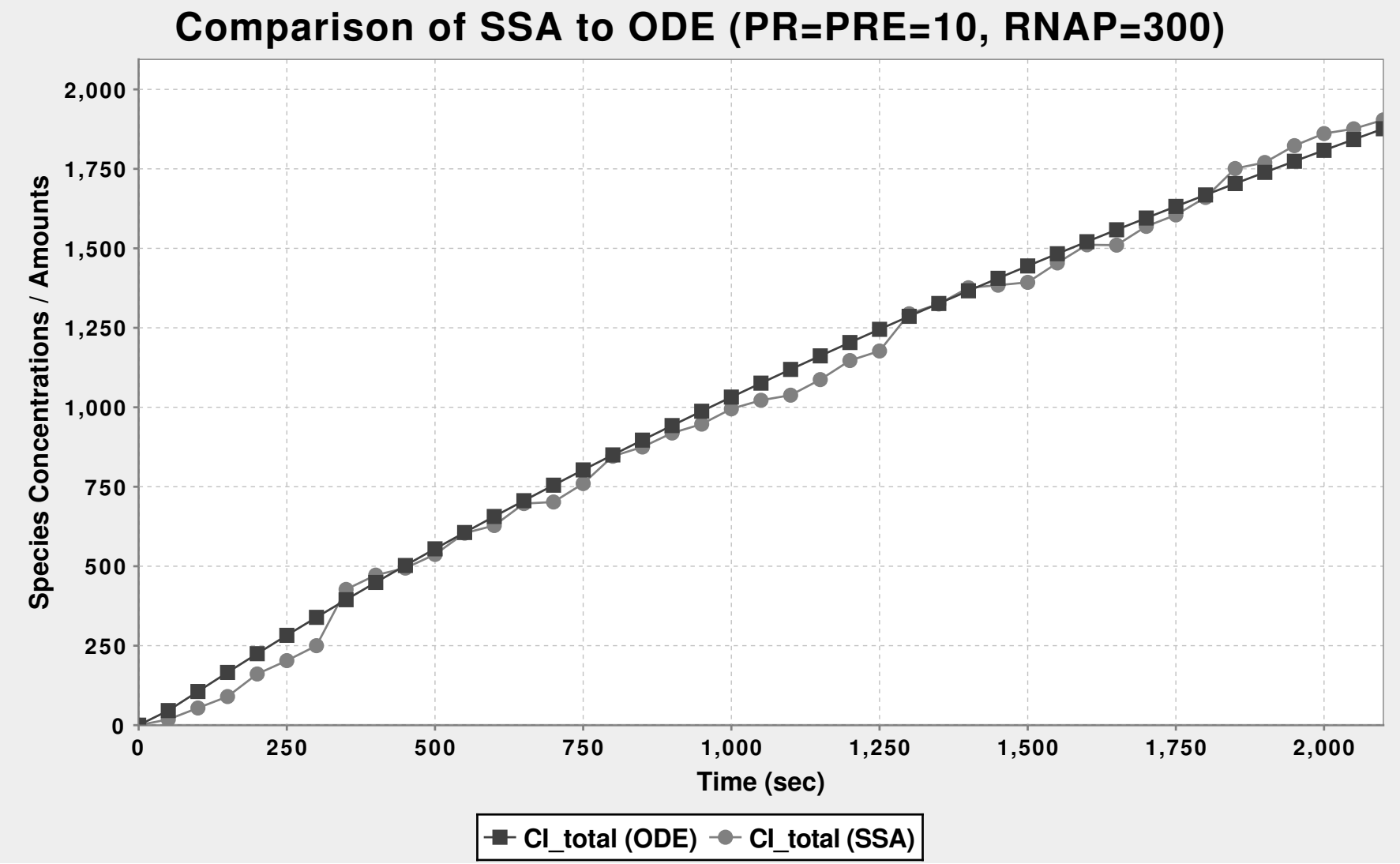
Simulation of P_{RE} and O_R Promoters



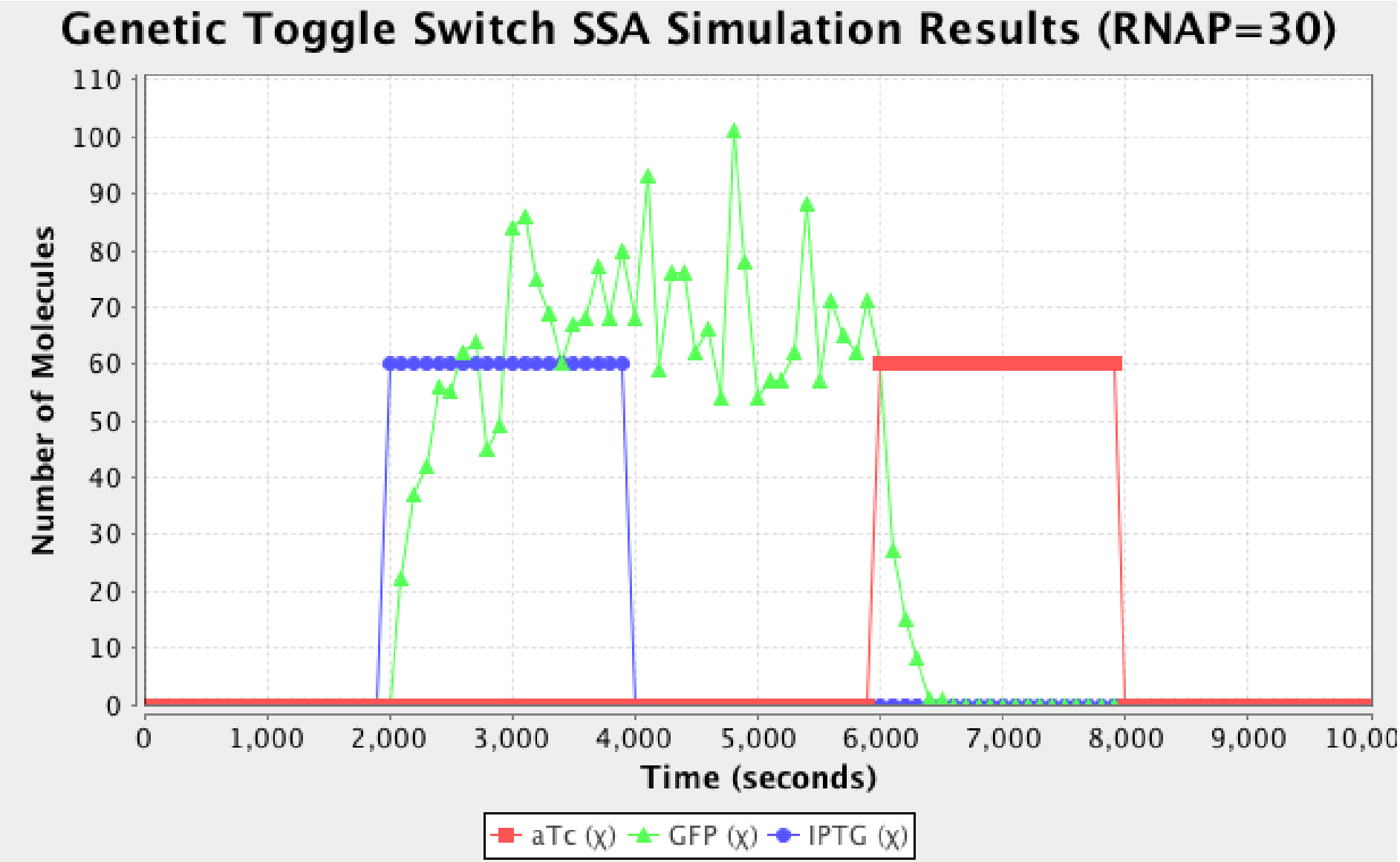
Discussion

- Using SSA to compute a single trajectory is no more complex than numerical simulation of reaction rate equations.
- Provides a closer approximation of molecular reality for systems with small molecule counts such as genetic circuits.
- Unfortunately, SSA has a substantial computational cost:
  - Must be run many times (1000s) to produce reasonable statistics while simulations of reaction rate equations only run once.
  - Very slow since τ is equal to $1/a_0(\mathbf{x})$ and can be very large when any molecule counts become large.
- When molecule counts increase, relative difference between deterministic and stochastic trajectories decrease.

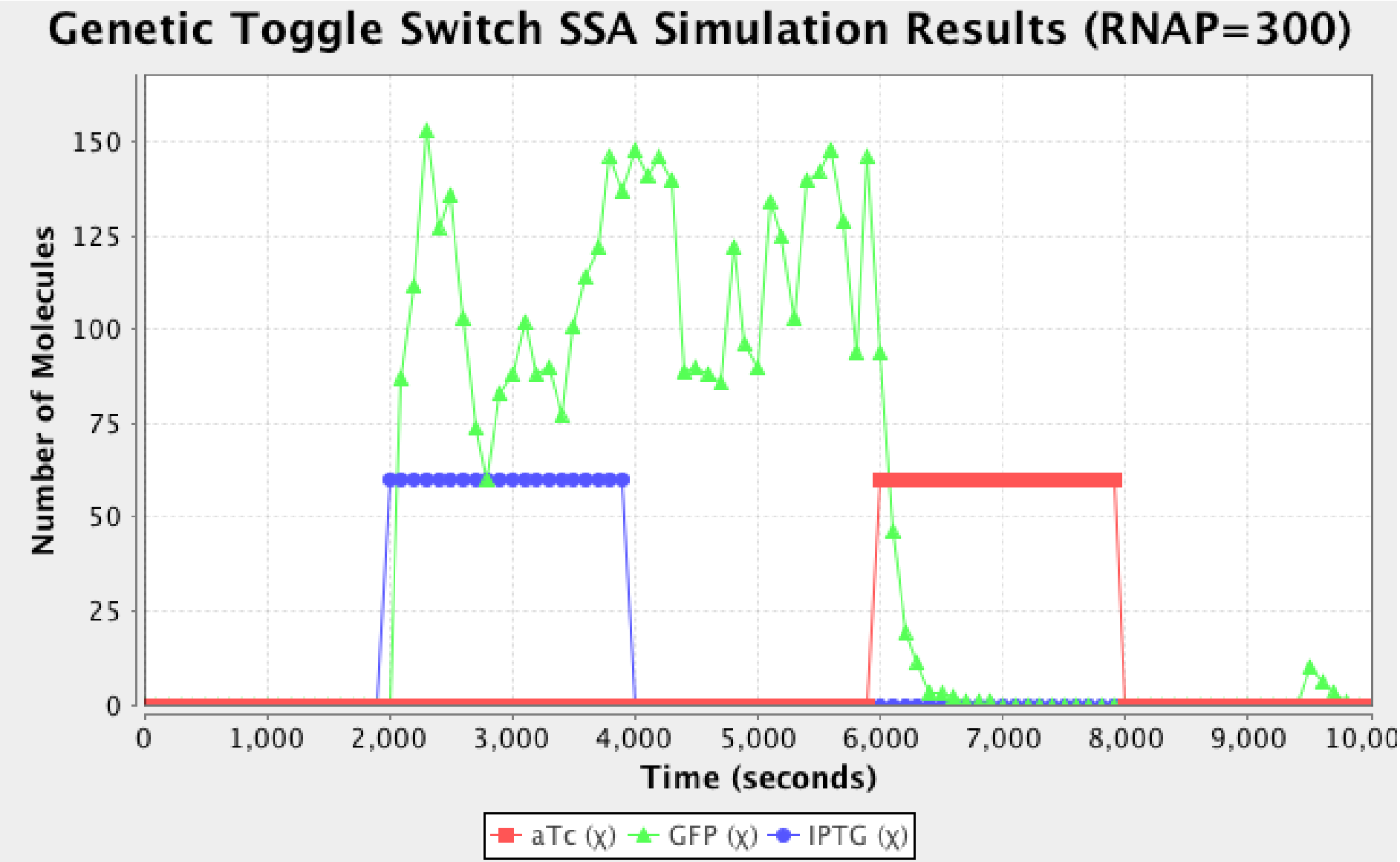
Simulation of P_{RE} and O_R Promoters



SSA Simulation of the Genetic Toggle Switch



SSA Simulation of the Genetic Toggle Switch



Assignment #6

- Perform SSA simulation on your genetic toggle switch model.
  - Set the initial value of LacI (in all models) to 60 molecules.
  - Add events that set IPTG to 60 at 2000s, IPTG to 0 at 4000s, aTc to 60 at 6000s, and aTc to 0 at 8000s.
  - Perform 100 SSA simulation runs for 10000s and compare with the expected toggle simulation results.
  - Graph the inputs IPTG and aTc, and for a single run and the average of all runs the output GFP.
  - Upload an archive of your project to `https://synbiohub.utah.edu` and provide a share link.
- Perform SSA simulation on your paper's genetic circuit model
  - Set initial conditions/parameters and add events to test your genetic circuit model and update your model as needed to get the results you expect.
  - Create a graph that demonstrates your model's behavior.
  - Upload an archive of your project to `https://synbiohub.utah.edu` and provide a share link.

Assignment #6 (cont)

- Simulate the reactions shown below using SSA. Create three trajectories with five reaction firings each.
- You may either do this simulation by hand, using a spreadsheet, or writing a simple program.
- Submit all your work.

$$
\begin{aligned}
\text{LacI} &\xrightarrow{k_d} () \\
\text{TetR} &\xrightarrow{k_d} () \\
pLac &\xrightarrow{k_o} np\ \text{TetR} + pLac \\
pTet &\xrightarrow{k_o} np\ \text{LacI} + pTet \\
pLac + nc\ \text{LacI} &\xrightarrow{k_{rf}} pLacLacI \\
pLacLacI &\xrightarrow{k_{rr}} pLac + nc\ \text{LacI} \\
pTet + nc\ \text{TetR} &\xrightarrow{k_{rf}} pTetTetR \\
pTetTetR &\xrightarrow{k_{rr}} pTet + nc\ \text{TetR}
\end{aligned}
$$

Constant	*Value*
$K_r = k_{rf}/k_{rr}$	$(0.1/1.0)\ M^{-nc}$
k_o	$0.1\ \text{sec}^{-1}$
k_d	$0.1\ \text{sec}^{-1}$
nc	2
np	10
$pLac$	1
$pTet$	1

Gillespie's First Reaction Method

1. Initialize: $t = t_0$ and $\mathbf{x} = \mathbf{x}_0$.
2. Evaluate propensity functions $a_j(\mathbf{x})$ at state $\mathbf{x}$.
3. For each j, determine the time, τ_j, until the next R_j reaction:

$$\tau_j = \frac{1}{a_j(\mathbf{x})} \ln\left(\frac{1}{r_j}\right).$$

   where each r_j is a unit uniform random number.
4. Let μ be the reaction whose τ_μ is the smallest.
5. Let τ equal τ_μ.
6. Determine the new state: $t = t + \tau$ and $\mathbf{x} = \mathbf{x} + v_\mu$.
7. If t is greater than the desired simulation time then halt.
8. Record $(\mathbf{x}, t)$ and goto step 2.

Observations

- First reaction method requires m random variables per simulation!
- OBSERVATION: not all propensities change after a reaction.
- Following three steps are taken during every iteration and take a time proportional to the number of reactions, m.
  1. Update all m propensity functions, $a_j(\mathbf{x})$.
  2. Generate m random numbers and next reaction times.
  3. Find the smallest reaction time, τ_μ.
- Must eliminate each of these performance bottlenecks.

Gibson/Bruck's Improvements

- τ_j and $a_j(\mathbf{x})$ stored for use in future iterations.
- τ_j uses absolute time to make it useful for multiple iterations.
- *Dependency graph* used to indicate relations between reactions.
  - Has vertex for each R_j and edge from R_j to other reaction that has as a reactant either a reactant or a product of R_j.
- Reuse every τ_j except the one for τ_μ, renormalizing τ_j when its propensity has changed as indicated by the dependency graph.
- *Indexed priority queue* used to organize $a_j(\mathbf{x})$ and τ_j data to make easy to update and to find smallest entry.
  - An indexed priority queue is a tree structure in which the parent always has a lower τ_j value than both its children.
  - This means the top node always has the smallest τ_j value.
  - Can be updated in $O(log(m))$ time.

Gibson/Bruck's Next Reaction Method

1. Initialize:
   1. $t = t_0$ and $\mathbf{x} = \mathbf{x}_0$.
   2. Generate a dependency graph, G.
   3. Evaluate propensity functions $a_j(\mathbf{x})$ at state $\mathbf{x}$.
   4. For each j, determine time, τ_j, until next R_j reaction:

$$\tau_j \quad = \quad t + \frac{1}{a_j(\mathbf{x})} \ln\left(\frac{1}{r_j}\right).$$

      where each r_j is a unit uniform random number.
   5. Store the τ_j values in an indexed priority queue Q.

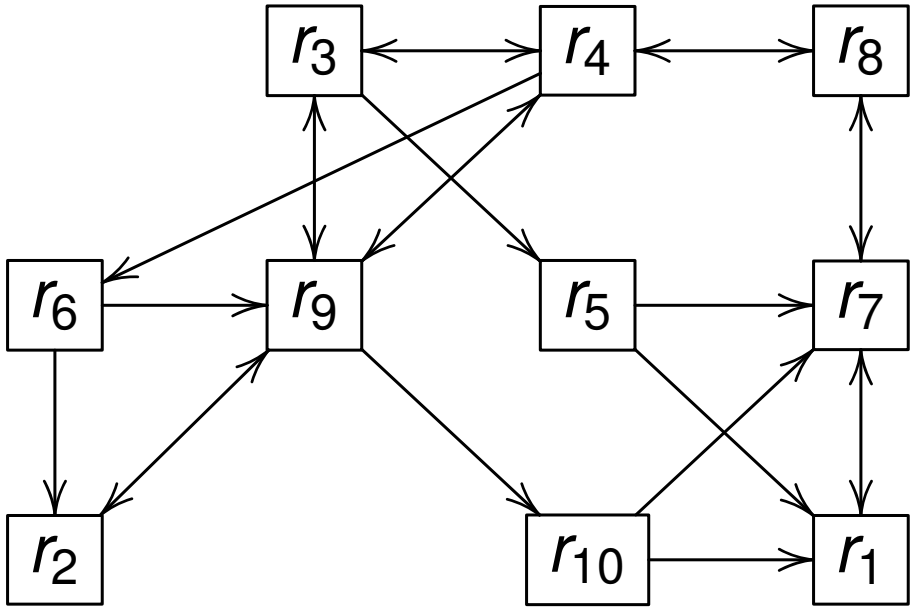
Gibson/Bruck's Next Reaction Method (cont)

2. Let R_μ be the reaction whose τ_μ is the smallest stored in Q.
3. Let τ equal τ_μ.
4. Determine the new state: $t = \tau$ and $\mathbf{x} = \mathbf{x} + v_\mu$.
5. For each edge (μ, α) in the dependency graph G,
   1. Set $a_{\alpha,old} = a_\alpha$ and update a_α.
   2. If $\alpha \neq \mu$, set $\tau_\alpha = (a_{\alpha,old}/a_\alpha)(\tau_\alpha - t) + t$
   3. If $\alpha = \mu$, generate a random number, r_μ, and

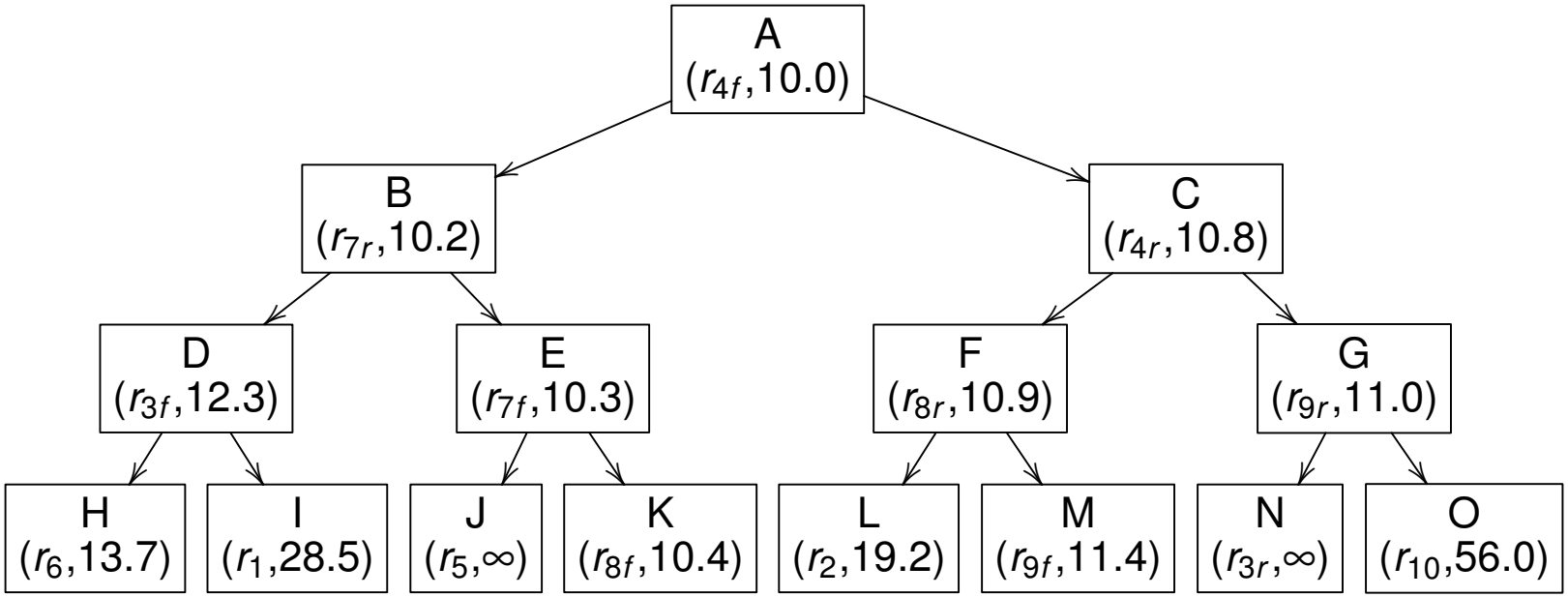
$$\tau_\mu \quad = \quad t + \frac{1}{a_\mu(\mathbf{x})} \ln\left(\frac{1}{r_\mu}\right).$$

6. If t is greater than the desired simulation time then halt.
7. Record $(\mathbf{x}, t)$ and goto step 2.

Dependency Graph



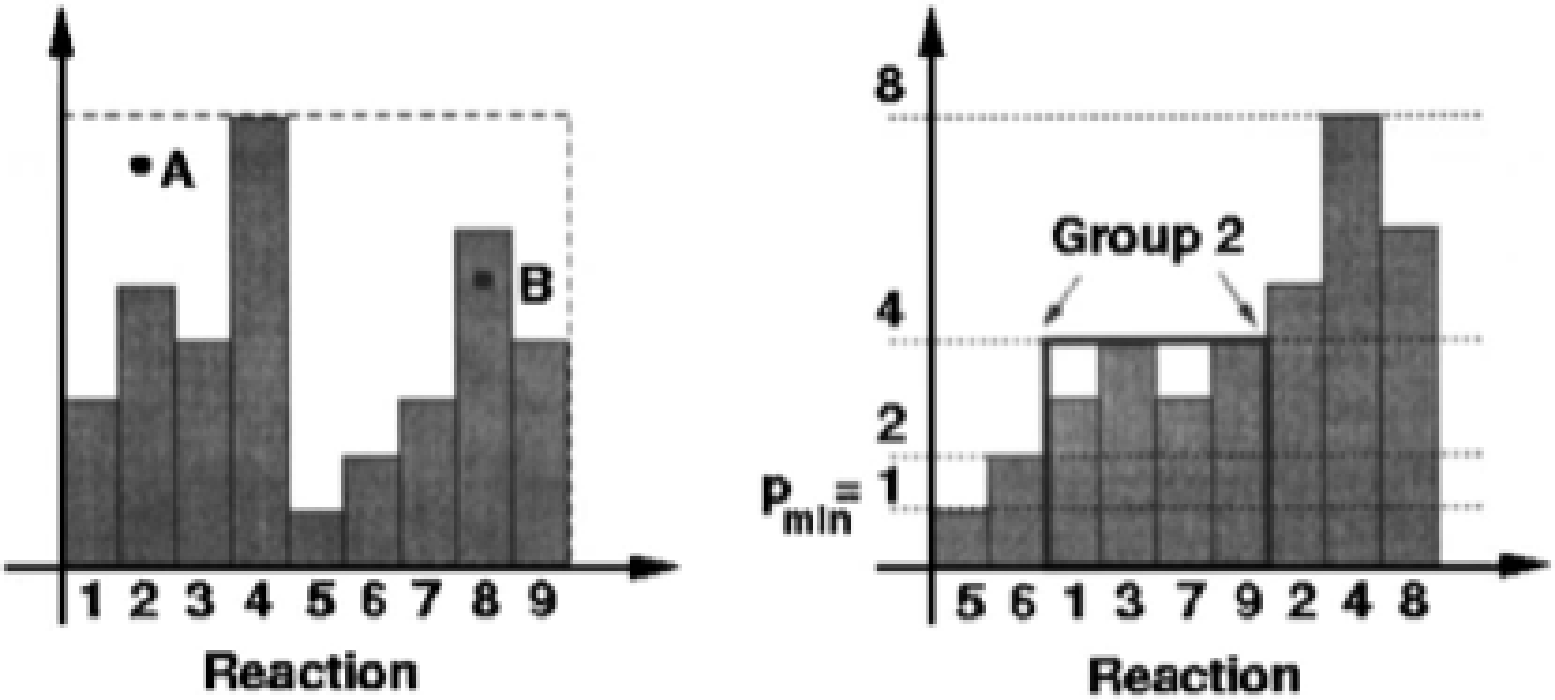
Indexed Priority Queue



r_1	r_2	r_{3f}	r_{3r}	r_{4f}	r_{4r}	r_5	r_6	r_{7f}	r_{7r}	r_{8f}	r_{8r}	r_{9f}	r_{9r}	r_{10}
I	*L*	*D*	*N*	*A*	*C*	*J*	*H*	*E*	*B*	*K*	*F*	*M*	*G*	*O*

Composition and Rejection

- Using *composition* and *rejection*, can construct a constant-time algorithm.



Composition and Rejection Algorithm (SSA-CR)

1. Initialize: $t = t_0$ and $\mathbf{x} = \mathbf{x}_0$.
2. Evaluate propensities $a_j(\mathbf{x})$ and assign to G groups.
3. Compute group propensities, $a^i(\mathbf{x})$, and total $a_0(\mathbf{x}) = \sum_{i=1}^{G} a^i(\mathbf{x})$.
4. Draw four unit uniform random numbers, r_1, r_2, r_3, and r_4.
5. Determine the time, τ, until the next reaction:

$$\tau = \frac{1}{a_0(\mathbf{x})} \ln\left(\frac{1}{r_1}\right).$$

6. Use r_2 to select a group of reactions (composition).
7. Use r_3 and r_4 to select reaction R_μ within the group (rejection).
8. Determine the new state: $t = t + \tau$ and $\mathbf{x} = \mathbf{x} + v_\mu$.
9. Compute $a_j(\mathbf{x})$ of affected reactions.
10. Assign affected reactions to groups, yielding new $a^i(\mathbf{x})$, and compute new total $a_0(\mathbf{x}) = \sum_{i=1}^{G} a^i(\mathbf{x})$.
11. If t is greater than the desired simulation time then halt.
12. Record $(\mathbf{x}, t)$ and goto step 3.

Tau Leaping

- Next reaction method still simulates every reaction event one at a time which is not practical for many interesting systems.
- *Tau-leaping* gives up exactness to improve simulation speed.
- Many reactions are fired at once in the time interval $[t, t+\tau]$.
- Introduce m random functions, $K_j(\tau; \mathbf{x}, t)$, where each returns the number of times that R_j fires in $[t, t+\tau]$ in state $\mathbf{X}(t) = \mathbf{x}$.
- New state after τ-leap is:

$$\mathbf{X}(t+\tau) \quad = \quad \mathbf{x} + \sum_{j=1}^{m} K_j(\tau, \mathbf{x}, t) v_j.$$

Tau Leaping (cont)

- Unfortunately, these functions are dependent on each other.
- Number of R_j reactions depends on $a_j(\mathbf{x})$ which depends on $\mathbf{x}$ which depends on number of all other reactions.
- Even if joint PDF can be computed, likely as expensive as full simulation.

Leap Condition

- States that τ be chosen to be small enough such that no propensity function changes by a significant amount.
- If satisfied, $K_j(\tau, \mathbf{x}, t)$ can be approximated to be a statistically independent *Poisson random variable*:

$$K_j(\tau, \mathbf{x}, t) \quad \approx \quad \mathcal{P}_j(a_j(\mathbf{x}), \tau) \;\; (j = 1, \ldots, m)$$

where $\mathcal{P}_j(a_j(\mathbf{x}), \tau)$ returns the number of events k in the interval $[t, t+\tau]$ such that:

$$\mathcal{P}[k \text{ events}] \quad = \quad \frac{e^{-a_j(\mathbf{x})\tau}(a_j(\mathbf{x})\tau)^k}{k!}$$

- τ must be small enough to satisfy leap condition, but large enough to fire enough events to speedup simulation.
- How to find τ small enough to satisfy leap condition, but large enough to fire enough events to speedup simulation?

Determining Tau

- One method for selecting τ uses the following equation:

$$\tau = \min_{i \in I_{rs}} \left\{ \frac{max\{\varepsilon_i x_i, 1\}}{|\sum_{j \in J_{ncr}} v_{ij} a_j(\mathbf{x})|}, \frac{max\{\varepsilon_i x_i, 1\}^2}{\sum_{j \in J_{ncr}} v_{ij}^2 a_j(\mathbf{x})} \right\} \quad (3)$$

  where I_{rs} are the species that appear as reactants in reactions and J_{ncr} are the *non-critical reactions*.
- A reaction is non-critical if it can be fired n_c times (between 5 and 30) without causing a reactant to become negative.
- The goal is that no propensity function changes by more than $\varepsilon a_j(\mathbf{x})$, where ε is an *accuracy control parameter* satisfying $0 < \varepsilon << 1$.
- The value of ε_i is ε for species that only appear in unimolecular reactions, $\varepsilon/2$ if it only appears in bimolecular reactions with different species, and $\varepsilon/(2 + (x_i - 1)^{-1})$ if it appears in bimolecular reactions with two molecules of the same species.

Explicit Tau-Leaping Simulation Algorithm

1. Initialize: $t = t_0$ and $\mathbf{x} = \mathbf{x}_0$.
2. Evaluate propensity functions $a_j(\mathbf{x})$ at state $\mathbf{x}$.
3. Determine J_{ncr}.
4. If $J_{ncr} = \emptyset$ then $\tau' = \infty$ else determine value for τ' using Equation 3.
5. If J_{ncr} includes all reactions then $\tau'' = \infty$ else use SSA to compute τ'' and j_c, the next critical reaction.
6. $\tau = min(\tau', \tau'')$ and $t = t + \tau$.
7. $\mathbf{x} = \mathbf{x} + \sum_{j \in J_{ncr}} \mathcal{P}_j\left(a_j(\mathbf{x})\tau\right)\mathbf{v}_j$.
8. If $\tau'' \leq \tau'$ then $\mathbf{x} = \mathbf{x} + \mathbf{v}_{j_c}$.
9. If t is greater than the desired simulation time then halt.
10. Record $(\mathbf{x}, t)$ and go to step 2.

Discussion

- ε provides means of trading off accuracy for runtime.
- For large ε, a significant runtime improvement can be achieved at the cost of some accuracy.
- Care has be taken though as large jumps can cause bad things such as species counts being made negative.
- As ε is made smaller, tau-leaping gradually reduces to the SSA.
- For a very small ε, not as efficient as SSA as it takes many τ leaps that produce no events.
- If τ much less than a few multiples of $1/a_0(\mathbf{x})$, revert to SSA.

The Chemical Langevin Equation

- If $\Delta t = \tau$ is small enough that no $a_j(\mathbf{x})$ changes significantly,

$$\mathbf{X}(t+\Delta t) \approx \mathbf{x} + \sum_{j=1}^{m} \mathcal{P}_j(a_j(\mathbf{x}), \Delta t) v_j.$$

- If Δt is large enough that there are many firings of each R_j, Poisson can be approximated with a Normal random variable:

$$\begin{aligned} \mathbf{X}(t+\Delta t) &\approx \mathbf{x} + \sum_{j=1}^{m} \mathcal{N}_j(a_j(\mathbf{x})\Delta t, a_j(\mathbf{x})\Delta t) v_j \\ &= \mathbf{x} + \sum_{j=1}^{m} v_j a_j(\mathbf{x})\Delta t + \sum_{j=1}^{m} v_j \sqrt{a_j(\mathrm{x})}\mathcal{N}_j(0,1)\sqrt{\Delta t} \end{aligned}$$

using $\mathcal{N}(m, \sigma^2) = m + \sigma\mathcal{N}(0,1)$.

The Chemical Langevin Equation (cont)

- If Δt is a *macroscopically infinitesimal* time increment dt,

$$\mathbf{X}(t+dt) \approx \mathbf{X}(t) + \sum_{j=1}^{m} v_j a_j(\mathbf{X}(t))dt + \sum_{j=1}^{m} v_j \sqrt{a_j(\mathrm{X}(t))} N_j(t)\sqrt{dt}$$

  where $N_j(t)$ are m statistically independent and temporally uncorrelated Normal random variables with mean 0 and variance 1.

- This equation is the *chemical Langevin equation*.

The Chemical Langevin Equation (cont)

$$\mathbf{X}(t+dt) \approx \mathbf{X}(t) + \sum_{j=1}^{m} v_j a_j(\mathbf{X}(t))dt + \sum_{j=1}^{m} v_j \sqrt{a_j(\mathrm{X}(t))} N_j(t)\sqrt{dt}$$

- Chemical Langevin Equation has two parts:
  - A deterministic part that grows linearly with $a_j(\mathbf{x})$.
  - A stochastic part that grows proportional to $\sqrt{a_j(\mathbf{x})}$.
- $a_j(\mathbf{x})$ grow in direct proportion system size.
- Stochastic part scales relative to deterministic part as the inverse square root of the system size.

The Reaction Rate Equation

- As size increases, magnitude of fluctuations diminish until chemical Langevin equation can be reduced to:

$$\mathbf{X}(t+dt) \approx \mathbf{X}(t) + \sum_{j=1}^{m} v_j a_j(\mathbf{X}(t))dt$$

- Rearranging this equation results in the following:

$$\frac{\mathbf{X}(t+dt) - \mathbf{X}(t)}{dt} = \frac{d\mathbf{X}(t)}{dt} = \sum_{j=1}^{m} \mathbf{v}_j a_j(\mathbf{X}(t)) \tag{4}$$

- This is simply the reaction rate equation, but it has been derived from stochastic chemical kinetics.
- Reaction rate equations valid when system large enough that no propensity changes significantly in dt and every R_j fires many times in dt.

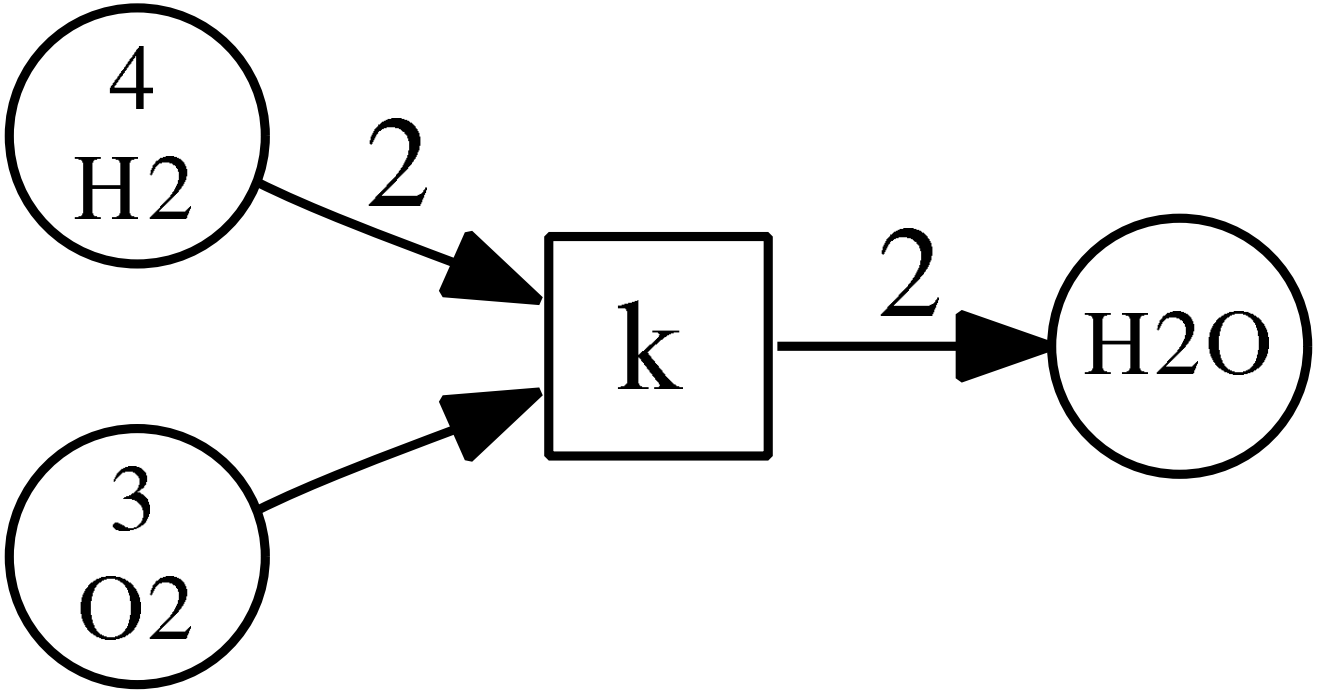
Stochastic Petri Nets

- One interesting model that has been applied to biological systems is *stochastic Petri nets* (SPNs).
- SPNs are a graphical representation which is quite similar to representations used in biochemistry.
- SPNs are also isomorphic to jump Markov processes.
- Several analysis tools have been developed for SPNs.

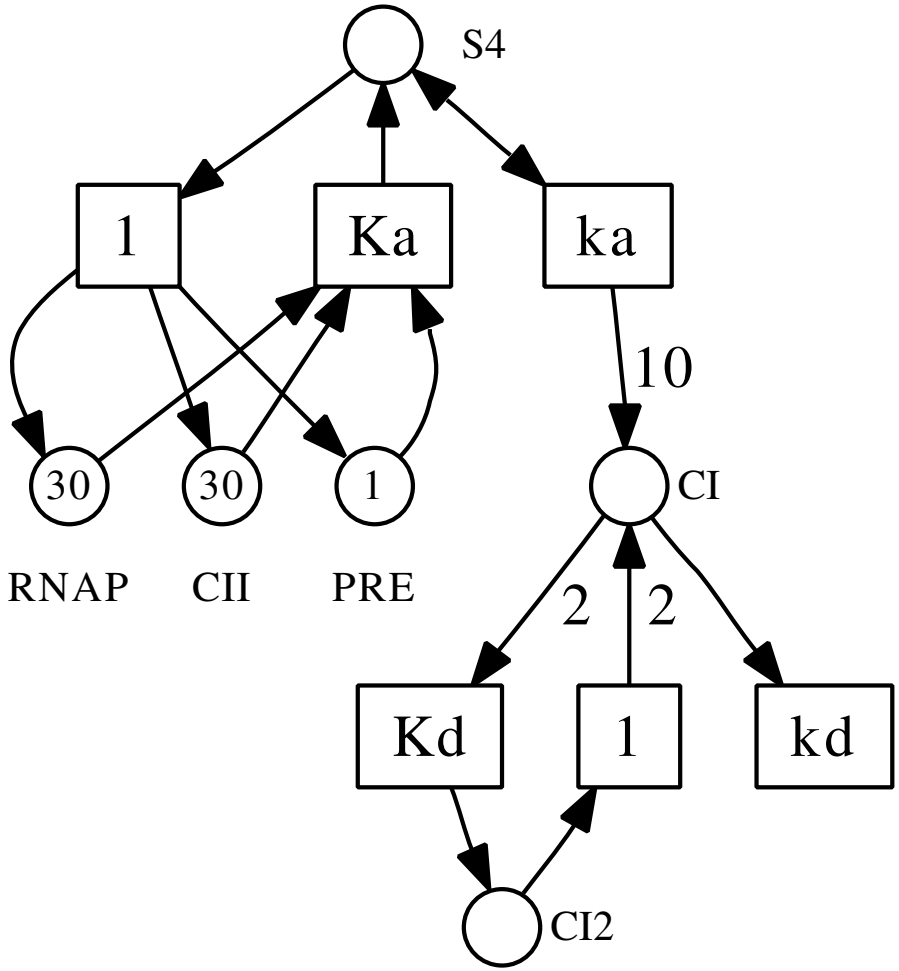
Stochastic Petri Net Model

- Composed of:
  - A set of *places* P (molecular species),
  - A set of *transitions* T (reactions),
  - An *input function* I (stoichiometry of reactants),
  - An *output function* O (stoichiometry of products),
  - A *weight function* W (rate of reaction), and
  - An *initial marking* M_0 (initial molecule counts).
- For elementary reactions, transition labeled with rate constant and assumed that rate function includes product of reactants.
- The state of an SPN is its *marking* which is an assignment of a number of tokens to each place in the net.
- Corresponds to current molecule counts.

Simple Stochastic Petri Net



SPN for Part of the Phage λ Model



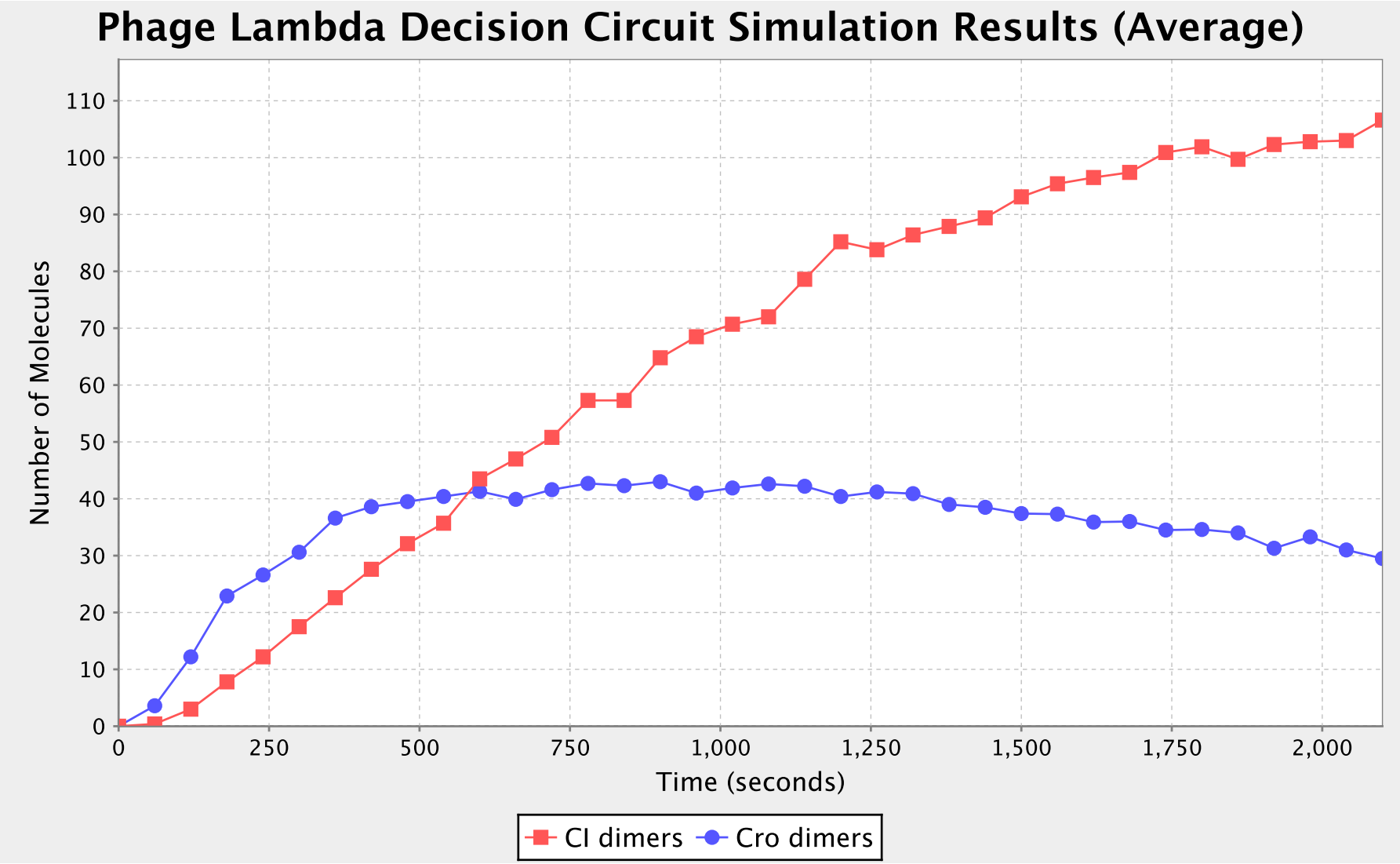
Phage λ Model

- Phage λ has two developmental pathways to replicate its DNA.
- The decision appears to be stochastic.
- Controled by two independently produced proteins competing for a switch.
- Resulting switch behavior is nondeterministic.
- Initially homogeneous population can follow different pathways.
- Two *E. Coli* in same environment and infected with the same number of phages, one may lyse while other is lysogenized.
- A deterministic model always results in exactly one possible outcome unless the parameters or initial conditions are changed.
- Therefore, stochastic analysis necessary to predict the probability that a cell heads down the lysis or lysogeny pathway after infection.

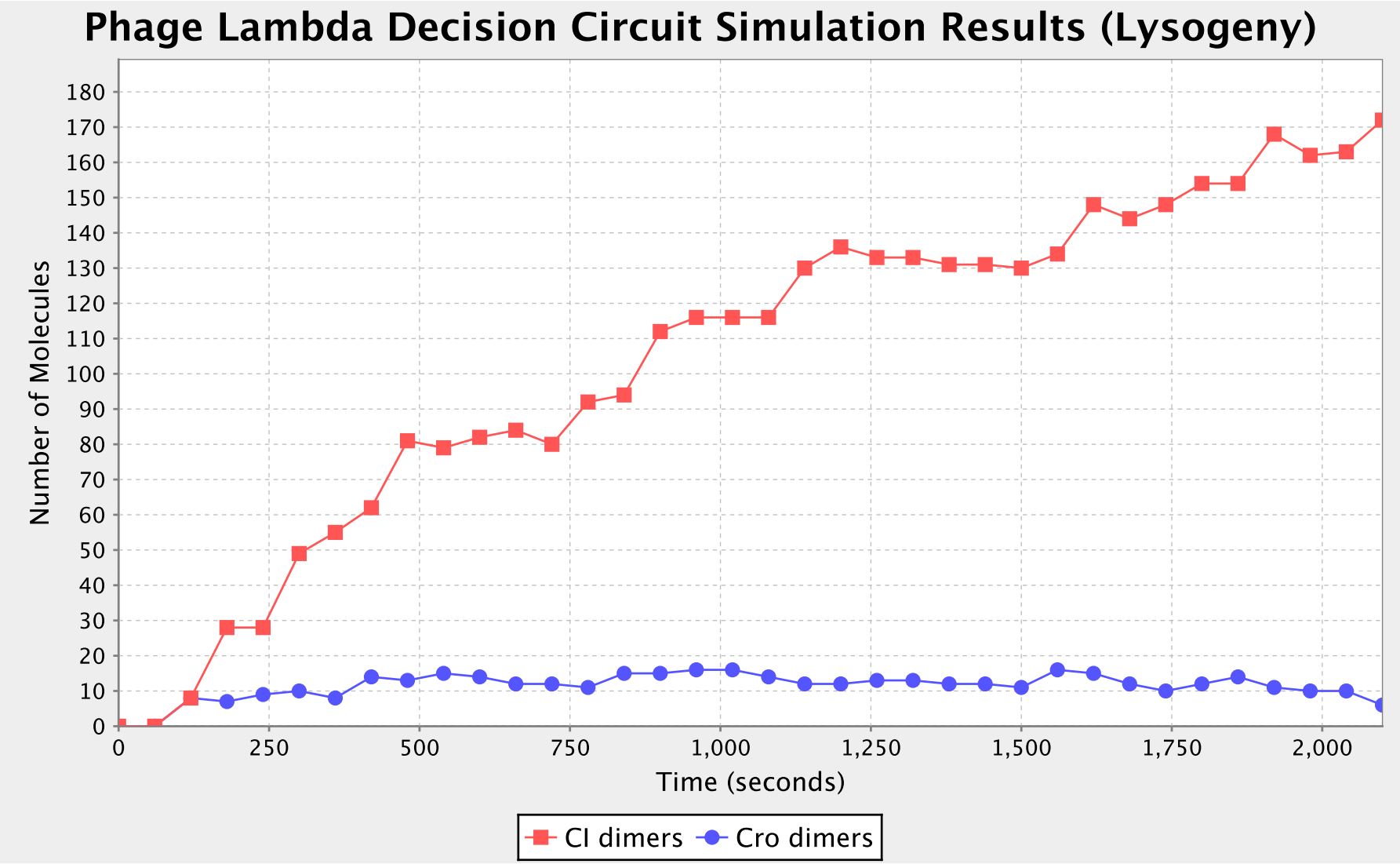
Lysis versus Lysogeny

- Goal of analysis is to predict probability of lysogeny.
- Shown experimentally to depend on *multiplicity of infection* (MOI), and on nutritional state of the cell.
- Well-fed cells tend to go into lysis.
  - Higher Hfl-related proteolytic activity shortens lifetimes of CII and CIII.
- Cells with higher MOI tend to lysogeny.
  - Hfl concentration is constant in MOI.
  - *cII* and *cIII* genes are proportional to MOI.
- Decision between lysis and lysogeny is essentially determined by a race between the buildup of the proteins Cro_2 and CI_2.
  - If Cro_2 reaches critical level first, result is lysis.
  - If CI_2 reaches critical level first, result is lysogeny.

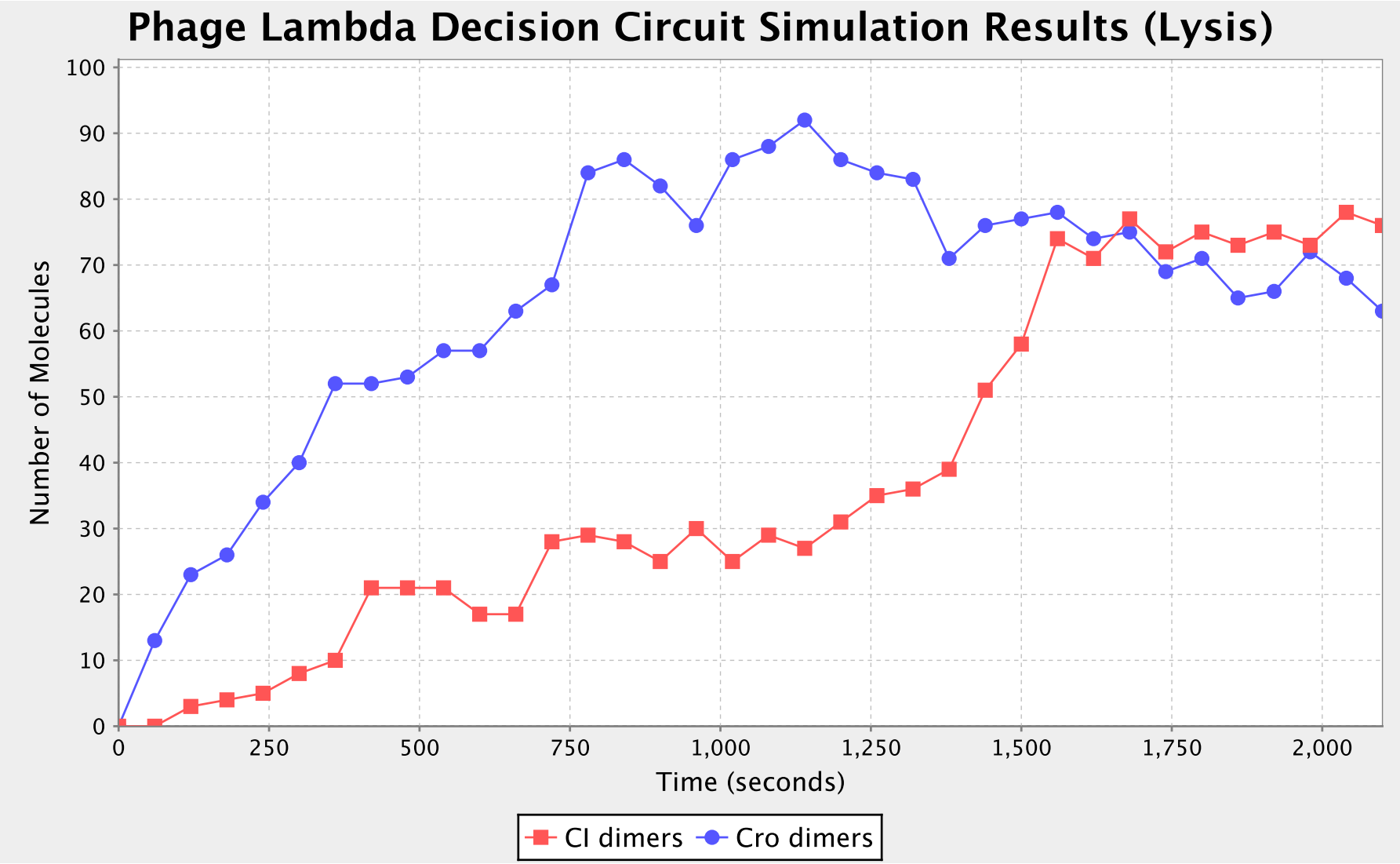
Time Courses (Average)



Time Courses (Lysogeny)



Time Courses (Lysis)

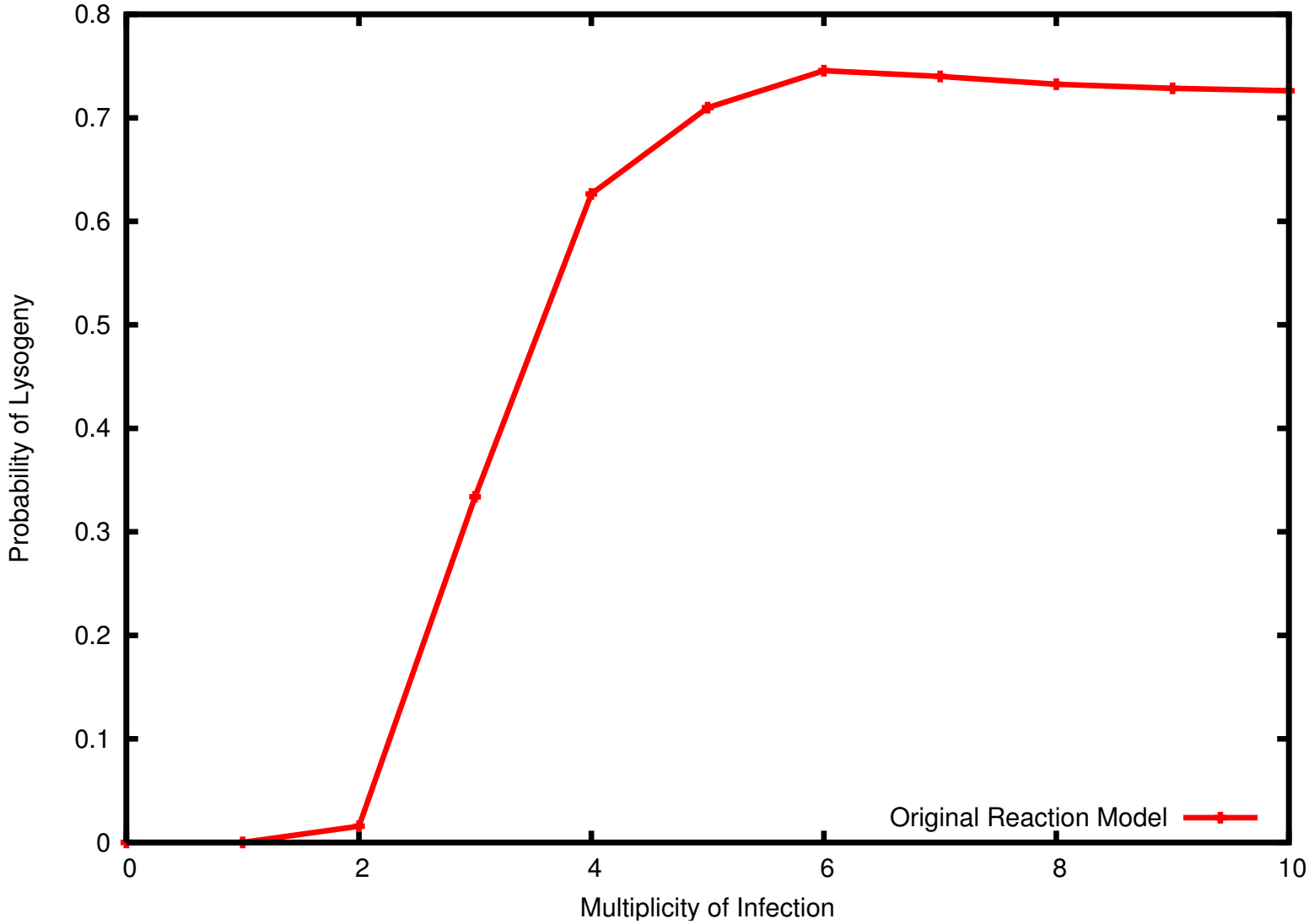


Kourilsky's Measurements

- Measured lysogenic fraction vs. API.
- Included measurements of O^- and P^- strains incapable of phage chromosome replication.
- Experiments performed on both starved and well-fed cells.
- Stochastic analysis is only practical for the starved data as the number of simulation runs goes like $1/f$ where f is fraction of lysogens.
- Determined probability of lysogeny, $F(M)$, using 10,000 runs of the SSA.
- Use Poisson distribution to map $F(M)$ to API data:

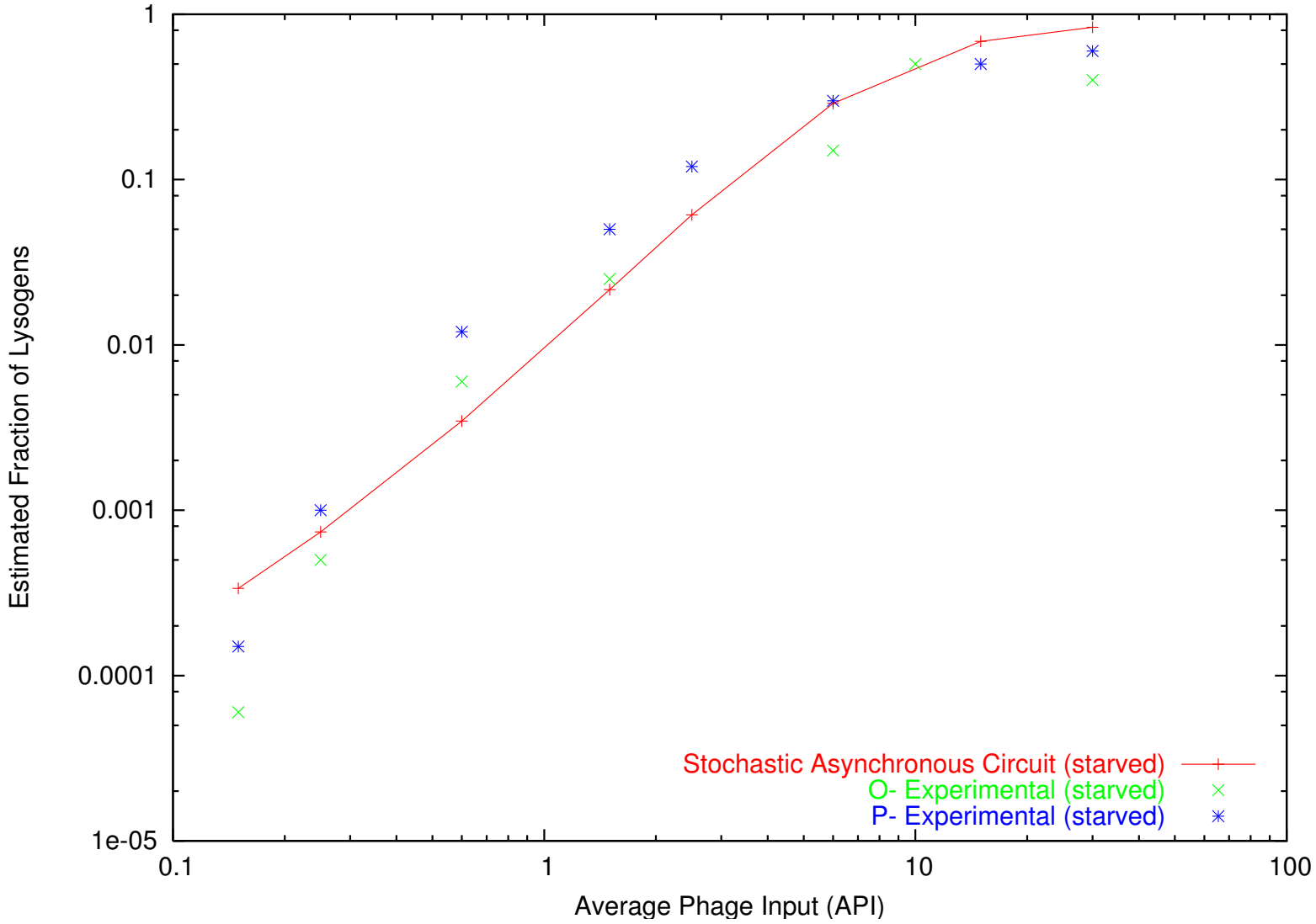
$$\begin{aligned} \mathcal{P}(M,A) &= \frac{A^M}{M!}e^{-A} \\ F_{\text{lysogen}}(A) &= \sum_M \mathcal{P}(M,A)\cdot F(M) \end{aligned}$$

Probability of Lysogeny



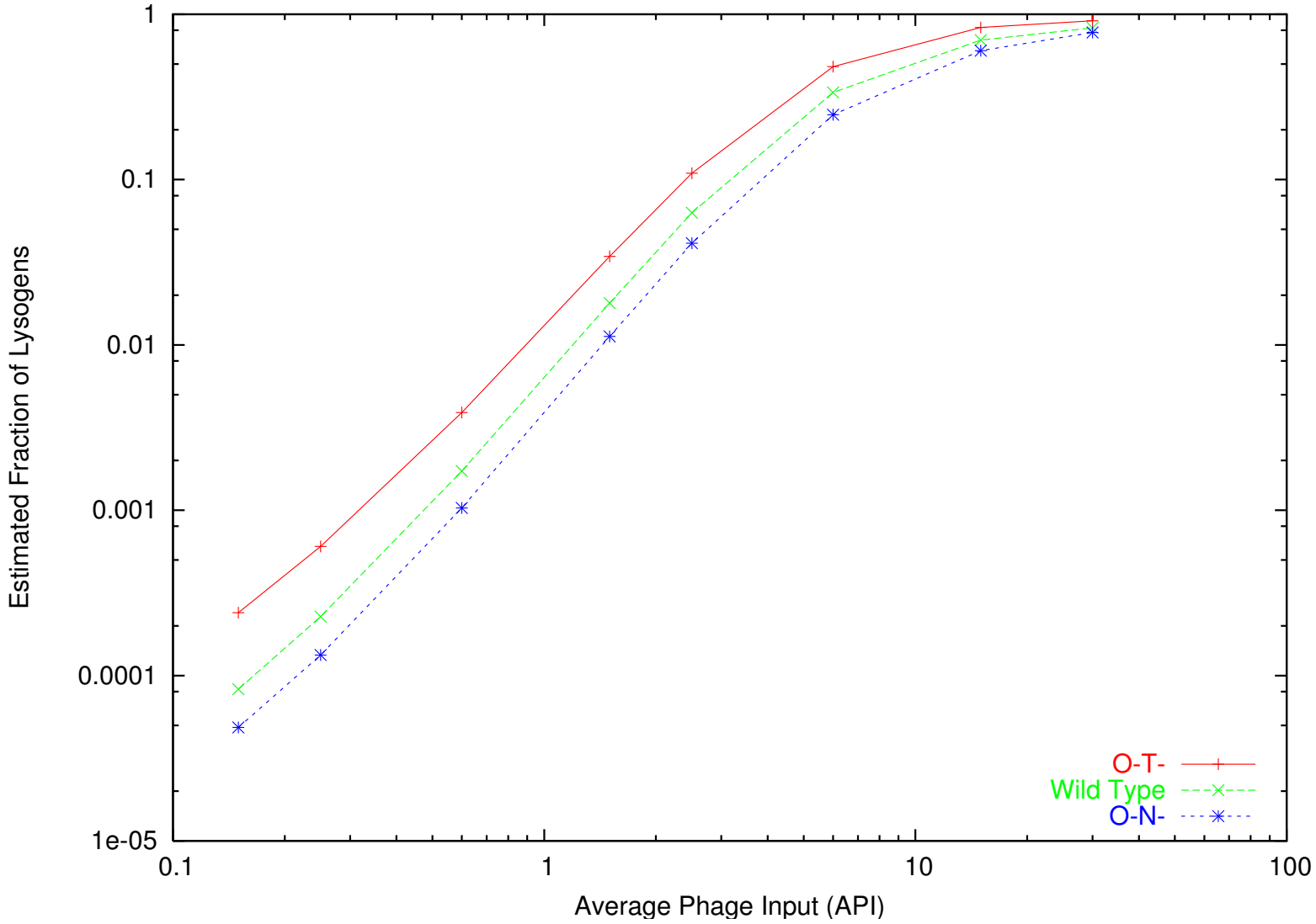
(courtesy of Kuwahara et al. (2006))

Lysogenic Fraction



(courtesy of Kuwahara et al. (2006))

Mutants

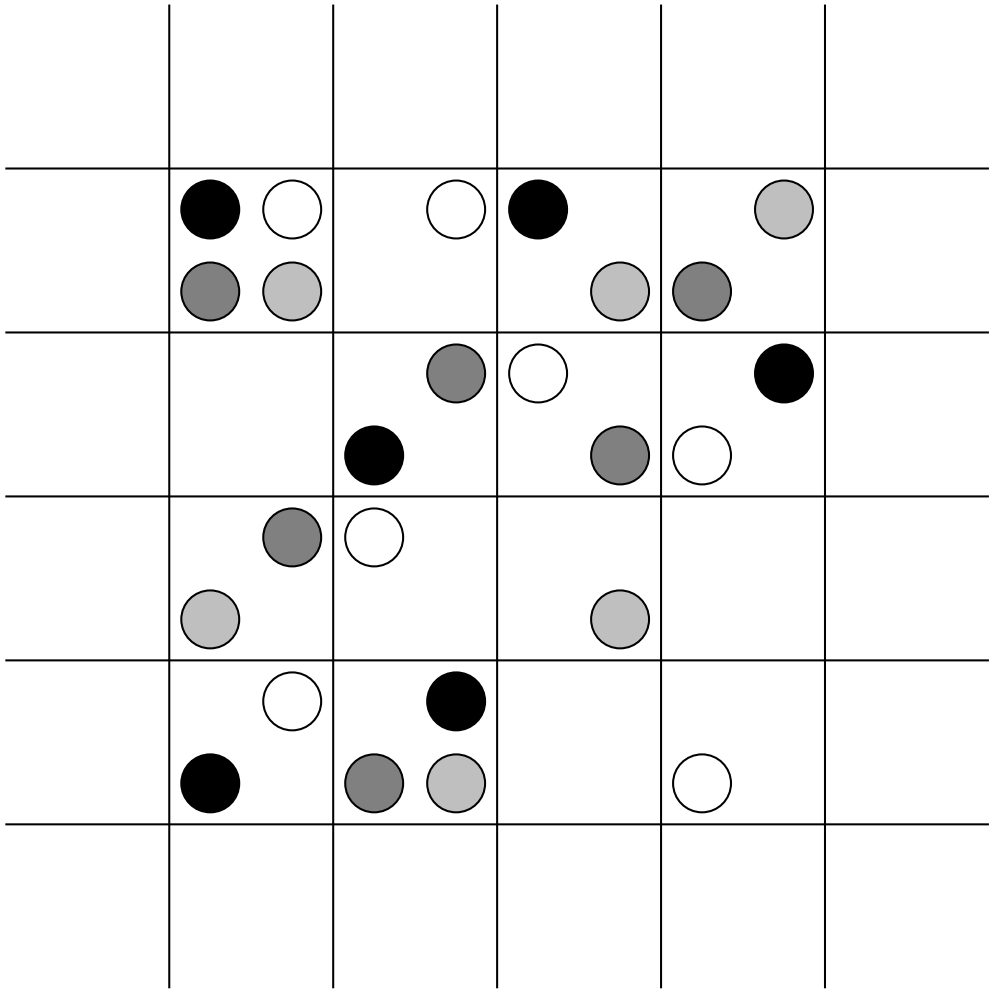


(courtesy of Kuwahara et al. (2006))

Spatial Gillespie

- As the volume increases, well-stirred assumption is less valid.
- Several methods proposed to add spatial considerations.
- Stundzia and Lumsden proposed spatial Gillespie method.
- System divided into several discrete subvolumes.
- Size of subvolumes selected such that within them well-stirred assumption is reasonable.
- Diffusion within subvolume faster than rate of reactions.

Discrete Subvolumes used by Spatial Gillespie



Spatial Gillespie (cont)

- State is now number of each species within each subvolume.
- During simulation cycle, molecule either reacts with others within the subvolume or diffuses to adjacent subvolume.
- Beginning with $(S_1, \ldots, S_n)$, spatial considerations added as follows assuming volume divided into $p \times q \times r$ subvolumes:

$$(S_1^{(i,j,k)}, \ldots, S_n^{(i,j,k)})$$

where $i = 1, \ldots, p$, $j = 1, \ldots, q$, and $k = 1, \ldots, r$.

$$\begin{aligned} S_\mu^{(i,j,k)} &\overset{K}{\leftrightarrow} S_\mu^{(i+1,j,k)} \\ S_\mu^{(i,j,k)} &\overset{K}{\leftrightarrow} S_\mu^{(i,j+1,k)} \\ S_\mu^{(i,j,k)} &\overset{K}{\leftrightarrow} S_\mu^{(i,j,k+1)}. \end{aligned}$$

- Now any stochastic simulation algorithm can be used.

Sources

- Stochastic simulation:
  - SSA - Gillespie (1977), Gillespie (1992), and Gillespie (2005).
  - Next reaction method - Gibson and Bruck (2000).
  - SSA-CR - Slepoy et al. (2008).
  - Tau-leaping - Gillespie and Petzold (2003), Cao et al. (2006), and Rathinam et al. (2003).
- Stochastic Petri nets:
  - SPNs - Molloy (1982) and Marsan et al. (1984).
  - Applied to modeling biological systems - Goss and Peccoud (1998).
- Stochastic analysis of phage λ:
  - Arkin et al. (1998).
- Spatial methods:
  - Survey of various methods - Takahashi et al. (2005).
  - Spatial Gillepsie method - Stundzia and Lumsden (1996).
- Chapter 4 of Engineering Genetic Circuits - Myers (2009).

Project Proposal

- Email your project idea to me ASAP.
- Project will be a mini-iGEM project.
  - Experimental track - design a genetic circuit *in silico*.
  - Software track - design a software tool for genetic design.
- Submit a one page project proposal by Friday October 19th.
- Figures and references do not count towards the page limit, and they should DEFINITELY be included.

Project Requirements

- Project updates due November 2rd and November 16th.
  - One page project update describing your progress.
  - Must show concrete evidence of your work to date on the project.
  - Experimental track: links to SynBioHub collections.
  - Software track: links to github (or similar) source code repositories.
- Project presentations on December 6th (approximately 10 minutes each).
- Project final report due on December 14th.
  - 4 pages including figures and references (appendices excluded) in IEEE Transactions format.
  - Experimental track: all design files submitted in a well-organized, documented SynBioHub collection.
  - Software track: all code files submitted on github (or similar), including documentation for running the software and example files to test with.