STOCHASTIC SIMULATION AND PARAMETER ESTIMATION OF GENE NETWORKS
OUTLINE

- Review of the stochastic models
  - Chemical master equation
  - Chemical Langevin equation

- Stochastic simulation algorithms (SSA)
  - Exact SSA
  - Approximate SSA
  - Multiscale SSA

- Parameter estimation-Particle filtering solution
CHARACTERISTICS OF GENE NETWORKS

- Gene networks model components
  - N molecular species, M reaction channels \( R_1 \) to \( R_M \)
  - State vector \( X(t) = [X_1(t), \ldots, X_N(t)]^T \)
  - State change vector \( \nu_m = [\nu_{1m}, \ldots, \nu_{im}, \ldots, \nu_{Nm}]^T \)
  - Rate constants \( c = [c_1, \ldots, c_M] \)
  - Propensity \( a_i(x) = c_i h_i(x), i = 1, \ldots, M \)
- Example: viral infection network

\[
\begin{align*}
R_1 : RNA & \xrightarrow{c_1} DNA, & R_2 : DNA & \xrightarrow{c_2} RNA + DNA, \\
R_3 : RNA & \xrightarrow{c_3} P + RNA, & R_4 : RNA & \xrightarrow{c_4} \emptyset, \\
R_5 : P & \xrightarrow{c_5} \emptyset, & R_6 : DNA + P & \xrightarrow{c_6} V,
\end{align*}
\]
**The Chemical Master Equation (CME)**

- Fundamental premise of dynamics:
  Given $X(t) = x$, the probability that one reaction $R_m$ will occur in the next infinitesimal time interval $[t, t+dt]$ is $a_m(x)dt$

- $X(t)$ is a discrete state Markov process

- The time evolution of its state probability is governed by CME

$$
\frac{\partial P(x, t)}{\partial t} = \sum_{m=1}^{M} [a_m(x - v_m)P(x - v_m, t) - a_m(x)P(x, t)]
$$

- Use to simulate the evolution
THE CHEMICAL LANGEVIN EQUATION

- Stochastic differential equation (SDE)
  \[ dX(t) = \sum_{m=1}^{M} v_m a_m(X(t)) \, dt + \sum_{m=1}^{M} v_m \sqrt{a_m(X(t))} \, dt \, N_m(t) \]

- Approximate \( X(t) \) as a continuous state Markov process

- Valid under some mild conditions

- Used in the parameter estimation, easier to formulate the objective
STOCHASTIC SIMULATION ALGORITHMS (SSAs)

- Question: given structure, how does $X(t)$ change?
  - solve CME to get the $P(x,t)$, difficult
  - use SSA to simulate realizations of $X(t)$, $0 < t < T$

- Exact SSA
  - Direct Method, First reaction method
  - Next reaction method
  - Optimized direct method, sorting direct method

- Approximate SSA
  - $\tau$-leap method
  - K-leap method

- Multiscale SSA
**Exact SSA**

- Generate the occurrence time of every reaction
- Correspond to a probability model identical to CME
- Types of exact SSA
  - Direct Method, First reaction method
  - Next reaction method
  - Optimized direct method, sorting direct method
DIRECT METHOD (DM)

- Purpose: Give $X(t) = x$, find out
  - the occurrence time $\tau$ of next reaction
  - which reaction will occur? Index $\mu$
  - based on fundamental premise: given $X(t) = x$, the probability that one reaction $R_m$ will occur in the next infinitesimal time interval $[t, t+dt]$ is $a_m(x)dt$

- $\tau$ and $\mu$ are independent random variables
  - $p(\tau) = a_0(x) \exp(-a_0(x)\tau)$, $a_0(x) = \sum_{m=1}^{M} a_m(x)$
  - $p(\mu) = a_{\mu}(x) / a_0(x)$, $\mu = 1, \ldots, M$
DIRECT METHOD PROCEDURE

- Initialization: set the initial number of molecules $X(0)$, set $t=0$
- Calculate the propensity functions, $a_m(x), m=1,\ldots, M$
- Generate $\tau$ and $\mu$
- Set $t \leftarrow t + \tau$, update state vector $X(t) \leftarrow X(t) + \nu_\mu$
- Go back to second step, or stop
SIMULATION RESULTS OF DM

- Viral infection network
  
  \[ R_1 : RNA \xrightarrow{c_1} DNA, \quad R_2 : DNA \xrightarrow{c_2} RNA + DNA, \]
  
  \[ R_3 : RNA \xrightarrow{c_3} P + RNA, \quad R_4 : RNA \xrightarrow{c_4} \emptyset, \]
  
  \[ R_5 : P \xrightarrow{c_5} \emptyset, \quad R_6 : DNA + P \xrightarrow{c_6} V, \]

- Simulation conditions
  
  - initial X(0) = [20, 200, 1000, 0]
  
  - c = [1 0.025 100 1 1.99 11.25 \times 10^{-6}] \text{ day}^{-1}
RESULTS OF DM

- Change of the amount of protein molecules (one realization)

<table>
<thead>
<tr>
<th>No</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steps</td>
<td>35697</td>
<td>35791</td>
<td>35302</td>
<td>33702</td>
<td>37414</td>
</tr>
<tr>
<td>Time(s)</td>
<td>13.2</td>
<td>10.9</td>
<td>12.8</td>
<td>10.7</td>
<td>13.5</td>
</tr>
</tbody>
</table>
FIRST REACTION METHOD (FRM)

- Procedure: (generate $\tau$ and $\mu$)
  - Calculate the propensity functions, $a_m(x), m=1,...,M$
  - Independently generate the occurrence time of each reaction channel $\tau_m, m=1,...,M$
  according to $p(\tau_m) = a_m(x) \exp(-a_m(x)\tau_m)$
  - Take
  $$\tau = \min\{\tau_1,...,\tau_M\}, \mu = \arg\min_m\{\tau_1,...,\tau_M\}$$

- Equivalent to DM
- Not efficient, need to generate more random variables
Next reaction method (NRM)

- Improve the efficiency of FRM using data structures
- Two observations
  1. Each $a_m(x)$ is only affected by a few reactions
     - Dependency graph: tell which $a_m(x)$ should be updated
  2. $\tau_m$ can be reused in the next step
     - Indexed priority queue
- More efficient than DM for loosely coupled systems

M. A. Gibson and J. Bruck, “Exact stochastic simulation of chemical systems with many species and many channels”
OTHER exact SSAs

- Optimized direct method (ODM)
  - incorporate dependency graph into the DM
  - preferable to NRM in systems which have multiscale nature

- Sorting direct method (variant of ODM), logarithmic direct method (use binary search tree)
  - more efficient under particular condition

- Exact SSA is slow for large systems
**Approximate SSAs**

- Speed up the simulation while giving up some exactness
- Instead of a single reaction, simulate a number of reactions in each simulation step
- Realistic in practical large systems
  - Heat shock response model: 61 reactions, 28 species
- Types of approximate SSAs: $\tau$-leap, K-leap, etc.
  
  Ref: X. Cai and X. Wang, “Stochastic Modeling and Simulation of Gene Networks”
τ-LEAP METHOD

- In each simulation step, advance the system by a preselected time \( \tau \)
- Difference: multiple reactions happen in \([t, t+ \tau]\)
- Choose \( \tau \) satisfying the leap condition (propensity function cannot change largely in \([t, t+ \tau]\))
  \[
  |\Delta a_m(\tau, x)| = |a_m(X(t+\tau)) - a_m(x)| \leq \varepsilon a_0(x) \quad \forall m=1,...,M
  \]
- Use the first order Taylor expansion of \( \Delta a_m(\tau, x) \)
- bound the mean and variance of \( \Delta a_m(\tau, x) \), get

\[
\tau = \min_m \left\{ \frac{\varepsilon a_0(x)}{|\eta_m(x)|}, \frac{\varepsilon^2 a_0^2(x)}{\sigma_m^2(x)} \right\}
\]
**τ-LEAP METHOD**

- **Procedure**
  - Initialization: set initial number $X(0), t=0$
  - Calculate propensity functions
  - Calculate $\tau$
  - If $\tau$ is too small (less than small multiple of $1/a_0(x)$), reject it and instead run a moderate number (say 100) of exact SSA, otherwise go to next step
  - For each reaction channel, generate the number of firing according to Poisson distribution with mean $a_m(x)\tau$
  - update $X(t)$, set $t \leftarrow t + \tau$
**BINOMIAL $\tau$-LEAP METHOD**

- Poisson random variable can be very large. The number of reactions in one time step may be too large so that the number of molecules may be negative at the end of the step.

- Use binomial distribution instead of Poisson distribution

- Avoid negative numbers, improve accuracy
SIMULATION RESULTS OF $\tau$ -LEAP METHOD

- Number of Protein Molecules (one realization)

<table>
<thead>
<tr>
<th>No</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steps</td>
<td>3537</td>
<td>4946</td>
<td>3736</td>
<td>4483</td>
<td>4783</td>
</tr>
<tr>
<td>Time</td>
<td>2.12</td>
<td>2.24</td>
<td>2.51</td>
<td>2.04</td>
<td>2.32</td>
</tr>
</tbody>
</table>
A unacceptable result with original $\tau$-leap
- negative number of proteins
**K-leap method**

- Instead of determining the leap time $\tau$ for each step, determine the total number of reactions-$K$ in each step
  - The number of reactions can be bounded. Hence the number of molecules and the propensity function will not change significantly, more accurate than $\tau$-leap.

- Procedure
  - Determine $K$ (various methods are developed)
  - Generate step time from the conditional distribution given $K$ (Gamma distribution)
  - Generate $Km$ from a multinomial pdf given $K$
**Multiscale SSA**

- Multiscale property: certain reaction channels fire much more frequently than others, almost always in gene networks.

- System can be partitioned into fast and slow reactions.

- Use a quasi-steady-state assumption to get the stationary pdf of fast reactions.

- Only need to simulate the slow reactions (it’s slow, number of firing is small).
SUMMARY OF SSA

- **Exact SSA:** exact to CME, slow, many improved methods

- **Approximate SSA:** faster than exact SSA, should carefully choose leap conditions

- **Multiscale SSA:** significantly speed up the simulation in multiscale systems
PARAMETER ESTIMATION

- Purpose: estimate rate constants \( c \) from noisy observations of the state vector \( X(t) \)
  - collect unknown parameters \( \theta = [c_1 \ldots c_M] \)
- Convert CLE to SDE form
  \[
  dX(t) = \sum_{m=1}^{M} v_m a_m(X(t))dt + \sum_{m=1}^{M} v_m \sqrt{a_m(X(t))} dt N_m(t)
  \]
  \[
  X(t + dt) - X(t) = Sa(X(t), \theta) dt + (SA(X(t), \theta) S^T)^{1/2} dW
  \]
- \( S = [v_1 \ldots v_M] \)
  \[
  a(X(t), \theta) = [a_1(X(t), c_1) \ldots a_M(X(t), c_M)]^T
  \]
  \[
  A(X(t), \theta) = \text{diag}\{a(X(t), \theta)\} \]
**Estimation model**

- Convert SDE to standard form

\[ X(t + dt) - X(t) = \mu(X(t), \theta) dt + \sigma(X(t), \theta) dW \]

- drift \( \mu(X(t), \theta) = S\alpha(X(t), \theta) \)

- diffusion \( \sigma(X(t), \theta) = (S\alpha(X(t), \theta) S^T)^{1/2} \)

- \( dW \) denotes Wiener process

- Observations

\[ y_i = y(i\Delta) = X(i\Delta) + \epsilon_i, \quad i = 1, \ldots, L \]
PARTICLE FILTERING ESTIMATION

Ref: X. Shen and H. Vikalo, “Inferring Parameters of Gene Regulatory Networks via Particle Filtering”

- Optimization model:
  \[ \hat{\theta} = \text{maximize}_{\theta} \ p(\theta \mid y_{1:L}) \]

- \[ p(\theta \mid y_{1:L}) = \int p(x_{1:L}, \theta \mid y_{1:L}) \ dx_{1:L} \]

- Use Bayes rule:
  \[ p(x_{1:L}, \theta \mid y_{1:L}) \propto p(y_{1:L} \mid x_{1:L}, \theta) \ p(x_{1:L} \mid \theta) \ p(\theta) \]
Particle Filtering Estimation

- $p(x_{1:L}, \theta \mid y_{1:L}) \propto p(y_{1:L} \mid x_{1:L}, \theta) p(x_{1:L} \mid \theta) p(\theta)$

- $p(y_{1:L} \mid x_{1:L}, \theta)$ is Gaussian according to observation model

- $p(\theta)$ is prior distribution

- Use Markov property
  
  $p(x_{1:L} \mid \theta) = p(x_L \mid x_{L-1}, \theta) \ldots p(x_2 \mid x_1, \theta) p(x_1 \mid \theta)$
 PARTICLE FILTERING ESTIMATION

- After transformation, the optimization model can be solved by particle filter if we can know transition density

\[ p(x_{n+1} \mid x_n, \theta) = p(X((n+1)\Delta) \mid X(n\Delta), \theta) \]

- For the SDE, the closed form transition density doesn’t exist.

- Euler Approximation

\[ x_{n+1} = x_n + \mu(x_n, \theta)\Delta + \sigma(x_n, \theta)\delta W, \ \delta W \sim N(0, \Delta I) \]
PARTICLE FILTERING ESTIMATION

- Euler approximation valid when $\Delta$ is small, not hold in experiments.

- Introduce missing values between observed data points. $\{z_1, \ldots z_{m-1}\}$

- Calculate transition as
  \[
  p(x_{n+1} | x_n) = p(x_{n+1} | z_{m-1}) p(z_{m-1} | z_{m-2}) \ldots p(z_2 | z_1) p(z_1 | x_n)
  \]

- Time interval is reduced
  \[
  z_{i+1} = z_i + \mu(z_i, \theta) \frac{\Delta}{m} + \sigma(z_i, \theta) \delta W, \quad \delta W \sim N(0, \frac{\Delta}{m} I)
  \]
**Verification of the Method**

- For exactness, have to use CME based exact SSA to generate the data

- Use a MCMC move step to improve the performance of particle filter

**Result**

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.08</td>
<td>0.82</td>
<td>0.09</td>
<td>0.9</td>
<td>0.25</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>0.0707</td>
<td>0.8219</td>
<td>0.0597</td>
<td>0.5625</td>
<td>0.3283</td>
<td>0.1195</td>
<td></td>
</tr>
<tr>
<td>0.35</td>
<td>0.3</td>
<td>0.1</td>
<td>0.1</td>
<td>0.12</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>0.2875</td>
<td>0.4167</td>
<td>0.1197</td>
<td>0.1432</td>
<td>0.1178</td>
<td>0.1384</td>
<td></td>
</tr>
</tbody>
</table>
The End