Sequential Monte Carlo in Model Comparison: Example in Cellular Dynamics in Systems Biology

C. Mukherjee* M. West†

Abstract

Sequential Monte Carlo analysis of time series provides a direct approach to evaluating approximate model marginal likelihoods for model comparison. We exemplify this in studies of dynamic bacterial communication in systems biology, where a long sequence of state vectors follow a complicated nonlinear dynamic model with several defining biochemical parameters. MCMC methods do not mix well in these contexts, and do not lead easily to reliable estimates of model marginal likelihood. We develop an auxiliary particle filtering algorithm that simultaneously updates latent states and fixed parameters. Our algorithm takes advantage of distributed computing to carry forward a huge number of particles to ensure accuracy of the estimates. Marginal likelihood computation is developed and illustrated in evaluation of relevance of selected model components.

Key Words: Distributed Computing, Dynamic Network Models, Marginal Likelihood, Model Comparison, Particle Filtering, Systems Biology

1. Introduction

Systems biology studies of dynamic cellular networks aim to develop mechanistic models of intra-cellular, biochemical interactions, often via differential equations overlaid on assumed biochemical networks [e.g., Bornholdt, 2005, Wilkinson, 2006]. Interest in such models is tied to developments in single-cell studies, in which data is generated on expression of genes or proteins in individual cells over time using time-lapse fluorescent microscopy [Elowitz et al., 2002, Rosenfeld et al., 2005, Wang et al., 2009]. Such experiments have potential to develop centrally in studies of natural biological systems as well as via synthetic biology – the latter involving engineering of networks with well-defined function, providing opportunity for controlled experimentation and design [Tan et al., 2007].

Formal statistical model assessment is really just beginning in this new field. Single cell time series data is as yet extremely limited – in terms of numbers of genes measured and time resolution – due to experimental limitations. Models of any realistic complexity contain many parameters and current data are unlikely to be informative about many of them. Assessing model forms is therefore challenging, while representing much of excitement and potential of emerging technologies to aid in *identifying* bionetwork structure.

We explore some of these issues a synthetic network model in bacterial cells [Tanouchi et al., 2008]. We develop discrete-time statistical dynamic models inspired by biochemical models of the regulatory gene network. These models incorporate noise "intrinsic" to biological networks as well as approximation and measurement errors, and provide the opportunity to formally evaluate the capacity of single cell data to inform on biochemical parameters and network structure. The primary focus is on sequential Monte Carlo (SMC) methods for model fitting, and Bayesian model assessment based on running multiple candidate network models in parallel. The ability to define effective SMC analyses requires propagation over time of very large numbers of particles in the combined space of *parameters+state variables*, and this is enabled by our development of efficient distributed/parallel implementations of SMC.

*Graduate Student, Department of Statistical Science, Duke University

†The Arts & Sciences Professor of Statistical Science, Department of Statistical Science, Duke University

2. Bayesian Inference for State-Space Models

In this paper we focus on a Markovian, nonlinear, non-Gaussian state-space model with infrequent observations:

$$\begin{array}{ll} \text{Prior:} & \pi(\theta)\pi(x_0|\theta), \\ \text{System Evolution:} & x_t \sim p(x_t|x_{t-1},\theta), \text{ for } t \geq 1, \\ \text{Observations:} & y_t \sim p(y_t|x_t,\theta), \text{ for } t \in t_1, t_2, \ldots, t_K. \end{array}$$

We denote by $x_{0:T} := \{x_0, x_1, \ldots, x_T\}$ the latent states and $y_{t_1,t_2,\ldots,t_K} := \{y_{t_1}, y_{t_2}, \ldots, y_{t_K}\}$ the set of observations made at times $0 < t_1 < t_2 < \ldots < t_K = T$. θ is a set of parameters that governs the model dynamics. The prime objective of our study is to be able to answer questions in terms of the model parameters θ in a Bayesian setting – in particular learning the posterior distribution $p(\theta|y_{t_1,t_2,\ldots,t_K})$ and performing model comparisons.

2.1 Markov Chain Monte Carlo Methods

We briefly discuss here an MCMC algorithm for the state-space model mentioned above. A general strategy is to iterate through the following set of complete conditionals:

$$x_{0:T} \sim p(x_{0:T}|\theta, y_{t_1,t_2,\ldots,t_K}) \quad \text{and} \quad \theta \sim p(\theta|x_{0:T}, y_{t_1,t_2,\ldots,t_K}).$$

For the linear, Gaussian case it is possible to perform block updating of $(x_{0:T}|\theta, y_{t_1,t_2,\ldots,t_K})$ using *Forward Filtering Backward Sampling* [Carter and Kohn, 1994, Frühwirth-Schnatter, 1994]. For a general nonlinear/non-Gaussian model blocked Gibbs updating is not possible; one strategy is to devise an approximate FFBS using local linearization/mixture of Gaussians to sample $x_{0:T}$ and use that as a proposal in a Metropolis-Hastings step. Two main issues with this strategy are the following:

- As the length of the time-series $x_{0:T}$ increases the performance of the approximate FFBS degrades. As a result the Metropolis-Hastings acceptance rate falls quickly.
- For large T, $p(\theta|x_{0:T}, y_{t_1,t_2,\ldots,t_K})$ becomes highly concentrated, therefore the MCMC mixes very poorly.

This introduces high autocorrelation among the MCMC samples and therefore to a poor effective sample size for estimating the parameters. We further illustrate this with our example.

2.2 Sequential Monte Carlo

For a general state-space model with infrequent observations we devise an auxiliary particle filter with simultaneous updating of states x_t and parameters θ. Sequentially updating $p(x_t, \theta_t|D_t)$ in particulate form leads to Monte Carlo samples from the target posterior, here D_t is the information set at time t. The subscript t on θ indicates they are samples from time t posterior, not that they are time varying. The inherent degeneracy in model parameters associated with sequential importance sampling is addressed as follows:

Let $\theta_t = (\phi_t, \psi_t)$ where there exists a conditional sufficient statistic s_t for ϕ_t but where ψ_t does not yield any such structure. We write the time t posterior for the (x_t, θ_t, s_t) as follows:

$$p(x_t, \phi_t, s_t, \psi_t|D_t) = \int p(\phi_t|s_t)\ p(s_t|s_{t-1}, x_t, D_t)\ p(x_t, ds_{t-1}, \psi_t|D_t). \tag{1}$$

Given a sample from $p(x_t, s_{t-1}, \psi_t|D_t)$, s_t is updated in a deterministic way: $s_t = \mathcal{S}(s_{t-1}, x_t, y_t)$ and subsequently ϕ_t is sampled from $p(\phi_t|s_t)$. This provides an exact way of replenishing particles in ϕ space [Storvik, 2002, Fearnhead, 2002, Johannes and Polson, 2007, Carvalho

et al., 2009] . For regeneration of ψ_t we resort to kernel smoothing [Liu and West, 2001]. For $t_{k-1} < t \leq t_k$ Bayes' theorem gives:

$$p(x_t, s_{t-1}, \psi_t | D_t) \propto \int p(y_{t_k} | x_{t_k}, \phi_{t_k-1}, \psi_{t_k})^{\mathbb{1}(t=t_k)} \; p(x_t, d\phi_{t-1}, s_{t-1} | \psi_t, D_{t-1}) \; p(\psi_t | D_{t-1}). \quad (2)$$

Liu and West [2001] suggest fitting a smooth kernel density to $\{\psi_{t-1}^{(i)}\}_{i=1}^M$ with shrinkage in locations and regenerate ψ_t from that density $p(\psi_t | D_{t-1})$. If $\bar{\psi}_{t-1}$ and $\Sigma(\psi_{t-1})$ are respectively the sample mean and covariance matrix of $\{\psi_{t-1}^{(i)}\}_{i=1}^M$, the shrinkage and particle regenerations are given by

$$\mu_{t-1}^{(i)} = a\psi_{t-1}^{(i)} + (1-a)\bar{\psi}_{t-1} \quad \text{and} \quad \psi_t^{(i)} \sim \mathbb{N}(\mu_{t-1}^{(i)}, \kappa^2 \Sigma(\psi_{t-1})), \qquad i = 1, 2, \ldots, M. \quad (3)$$

Smoothness of the kernel density is determined by a discount factor δ; $a = (3\delta - 1)/2\delta$ and $\kappa^2 = 1 - a^2$. Liu and West [2001] further suggests using an auxiliary variable to reduce degeneracy. For $t_{k-1} \leq t < t_k$ we use a strategy of looking ahead to the next available observation y_{t_k} and carry forward particles that are more compatible. If m_{t-1} is a prior point estimate of $(x_{t_k} | x_{t-1}, \phi_{t-1}, \psi_t)$, we can write

$$p(x_t, s_{t-1}, \psi_t | D_t) \propto \int \frac{p(y_{t_k} | x_{t_k}, \phi_{t_k-1}, \psi_{t_k})^{\mathbb{1}(t=t_k)}}{p(y_{t_k} | m_{t-1}, \phi_{t-1}, \mu_{t-1})} \; p(x_t | x_{t-1}, \phi_{t-1}, \psi_t)$$
$$p(y_{t_k} | m_{t-1}, \phi_{t-1}, \mu_{t-1}) \; p(dx_{t-1}, d\phi_{t-1}, s_{t-1} | \psi_t, D_{t-1}) \; p(\psi_t | D_{t-1}). \quad (4)$$

$\{x_{t-1}^{(i)}, \phi_{t-1}^{(i)}, \psi_t^{(i)}\}_{i=1}^M$ are resampled with weights proportional to $p(y_{t_k} | m_{t-1}^{(i)}, \phi_{t-1}^{(i)}, \mu_{t-1}^{(i)})$, and are then evolved with the kernel $p(x_t | x_{t-1}, \phi_{t-1}, \psi_t)$ and finally are associated with weights $p(y_{t_k} | x_{t_k}^{(i)}, \phi_{t_k-1}^{(i)}, \psi_{t_k}^{(i)})^{\mathbb{1}(t=t_k)} / p(y_{t_k} | m_{t-1}^{(i)}, \phi_{t-1}^{(i)}, \mu_{t-1}^{(i)})$. Here we write (1)-(4) in the form of an algorithm.

Algorithm: APF for State-Space Model

1. For each $i = 1, 2, \ldots, M$, sample from the prior $\theta_0^{(i)} \sim \pi(\theta)$ and $x_0^{(i)} \sim \pi(x_0 | \theta_0^{(i)})$.

   For $t = 1, 2, \ldots, T$ perform 2-8 repeatedly

2. For each $i = 1, 2, \ldots, M$ calculate the prior point estimate $m_{t-1}^{(i)}$ and the shrinkage location $\mu_{t-1}^{(i)}$.

3. For each $i = 1, 2, \ldots, M$ select an auxiliary integer variable k_i from $\{1, 2, \ldots, N\}$ with probabilities proportional to

$$g_t^{(l)} = w_{t-1}^{(l)} \; p(y_{t_k} | m_{t-1}^{(l)}, \phi_{t-1}^{(l)}, \mu_{t-1}^{(l)}) \text{ where } t_{k-1} < t \leq t_k, \; l \in \{1, 2, \ldots, N\}.$$

4. Sample ψ_t from the kernel density estimate of $p(\psi_t | D_{t-1})$,

$$\psi_t^{(i)} \sim \mathbb{N}(\mu_{t-1}^{(k_i)}, \kappa^2 \Sigma(\psi_{t-1})), \text{ for } i = 1, 2, \ldots, M.$$

5. Sample current state vector $x_t^{(i)} \sim p(x_t | x_{t-1}^{(k_i)}, \phi_{t-1}^{(k_i)}, \psi_t^{(i)})$, for $i = 1, 2, \ldots, M$.

6. Evaluate the corresponding weight

$$w_t^{(i)} = \left[\frac{p(y_{t_k} | x_{t_k}^{(i)}, \phi_{t_k-1}^{(k_i)}, \psi_{t_k}^{(i)})}{\prod_{\tau=t_{k-1}}^{t_k-1} p(y_{t_k} | m_\tau^{(k_i)}, \phi_\tau^{(k_i)}, \mu_\tau^{(k_i)})} \right]^{\mathbb{1}(t=t_k)}$$

7. Update sufficient statistics: $s_t^{(i)} = \mathcal{S}(s_{t_k-1}^{(k_i)}, x_{t_k}^{(i)}, y_{t_k}) \mathbb{1}(t = t_k) + s_{t-1}^{(k_i)} \mathbb{1}(t \neq t_k)$.

8. Sample $\phi_t^{(i)} \sim p(\phi_t | s_t^{(i)})$, for $i = 1, 2, \ldots, M$.

2.2.1 Marginal Likelihood Calculation

The SMC methods have an added advantage of producing *marginal likelihood* estimates without much further computation. Note that the marginal likelihood has the structure

$$p(y_{t_1}, y_{t_2}, \ldots, y_{t_k}) = p(y_{t_1}, y_{t_2}, \ldots, y_{t_{k-1}})\; p(y_{t_k} | y_{t_1}, y_{t_2}, \ldots, y_{t_{k-1}}).$$

The update is a very simple expression for our auxiliary particle filter, in that

$$p(y_{t_k} | y_{t_1}, y_{t_2}, \ldots, y_{t_{k-1}}) = \prod_{t=t_{k-1}+1}^{t_k} \left\{ \left(\sum_{i=1}^{M} g_t^{(i)} \right) \left(\frac{1}{M} \sum_{i=1}^{M} w_t^{(i)} \right) \right\}.$$

These marginal likelihood values can be used to perform Bayesian model comparison. We elaborate this with our example in the next section.

3. Distributed Computing

Performance of SMC critically depends upon the number of particles used. Theoretical results connecting the required number of particles to the desired accuracy of a particle filter can be found in Del Moral [2004] and in the works of Crisan, Doucet, Jasra and many others. A general rule is that, the higher the dimension of the particles and longer the filter is run, one would require more particles to achieve required accuracy. Fortunately this can be done in a surprisingly efficient manner using distributed computing platforms. Note that every SMC algorithm sequentially performs the following three steps in some order:

(a) Sampling step: Evolving state vector x_{t-1} to x_t and regenerating parameters θ.

(b) Importance step: Computation of particle weights $g_t^{(i)}$ and $w_t^{(i)}$.

(c) Weight normalization and resampling step.

While computations in (a) and (b) can be trivially parallelized on any number of processing units, step (c) requires communication among the processors. Let $N_t^{(i)}$ be the number of times the i^{th} particle is replicated in a resampling step drawing N particles, written as:

$$(N_t^{(1)}, N_t^{(2)}, \ldots, N_t^{(M)}) \sim \text{SRSWR}\left(N; w_t^{(1)}, \ldots, w_t^{(M)}\right).$$

Then on a *"1 master K slave processors"* architecture with each slave evolving M particles, we use the following algorithm to perform weight normalization and resampling in distributed manner:

Algorithm: Weight Normalization and Resampling in a Distributed Environment

1. Every slave computes the processor total weight $W_t^{[k]}$ in parallel.
2. $\{W_t^{[1]}, W_t^{[2]}, \ldots, W_t^{[K]}\}$ are sent to the master.
3. The master processor normalizes the processor total weights and performs *Inter-resampling*, i.e. it computes the replication numbers to individual processors:

$$(N_t^{[1]}, N_t^{[2]}, \ldots, N_t^{[K]}) \sim \text{SRSWR}\left(MK; W_t^{[1]} / \sum_{k=1}^{K} W_t^{[k]}, \ldots, W_t^{[K]} / \sum_{k=1}^{K} W_t^{[k]}\right).$$

4. Master sends the replication number $N_t^{[k]}$ to the k^{th} slave.

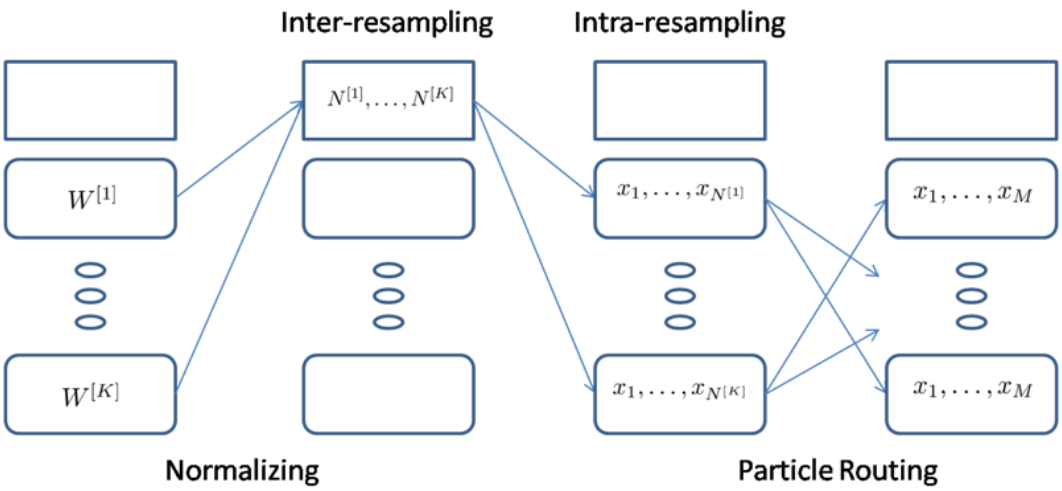

Figure 1: A schematic representation of the weight normalization and resampling step on a 1 master K slave architecture

5. The slave processors perform *Intra-resampling* in parallel, i.e. assigning replication number to individual particles:

$$(N_t^{(1)}, N_t^{(2)}, \ldots, N_t^{(M)}) \sim \text{SRSWR}\left(N_t^{[k]}; w_t^{(1)}/W_t^{[k]}, \ldots, w_t^{(M)}/W_t^{[k]}\right).$$

6. *Particle Routing:* Particles are transferred from surplus processors to deficient processors.

We recognize the following aspects of the distributed resampling step:

- There is no natural concurrency among the processors during the *intra-resampling*.
- Particle routing is often extensive, however this step can not be avoided because otherwise the discrepancy between the computational loads of the processors keep increasing. Depending on the connection module between processors this step can be made more efficient. Discussion of an efficient VLSI implementation of a particle filter can be found in Shabany and Gulak [2006].

Note that the objective of the resampling step is to focus the computational effort in the high density region, so we replicate particles with higher weights. The *multinomial resampling* has complexity of $\mathcal{O}(M^2)$; it often becomes a bottleneck computation in a particle filter with huge number of particles. **?** discussed various $\mathcal{O}(M)$ resampling algorithms; we choose the *Residual Systematic Resampling (RSR)* algorithm for its ease of implementation and utility in a distributed computing context.

4. Example: A Bacterial Communication Model in Systems Biology

Here we use the bacterial communication system investigated in Tanouchi et al. [2008] to illustrate modelling and inferential challenges in the studies of cellular dynamics. In particular we show how some biological questions can be transcribed into statistical learning and model selection problems, and how we can address them.

The bacterial communication system we are considering here is established by quorum sensing (QS); a canonical example is the *lux* system from the marine bacterium *Vibrio fischeri*

(Figure 2). This system consists of two genes encoding proteins LuxI and LuxR. LuxI is an AHL (acyl homoserine lactone) synthase; LuxR is a transcriptional regulator activated by the AHL. The AHL signal is produced inside the cell but freely diffuses across the cell membrane into the environment: therefore, the AHL concentration is low at a low cell density. As the cell density increases, the signal accumulates in the environment and inside the cell. At sufficiently high concentrations, AHL can bind to and activate LuxR, which will then activate downstream genes that are critical for regulating diverse physiological functions, such as bioluminescence, biofilm formation, and bacterial pathogenicity. This mechanism provides an elegant strategy for bacteria to sense their density and to achieve coordinated population behavior.

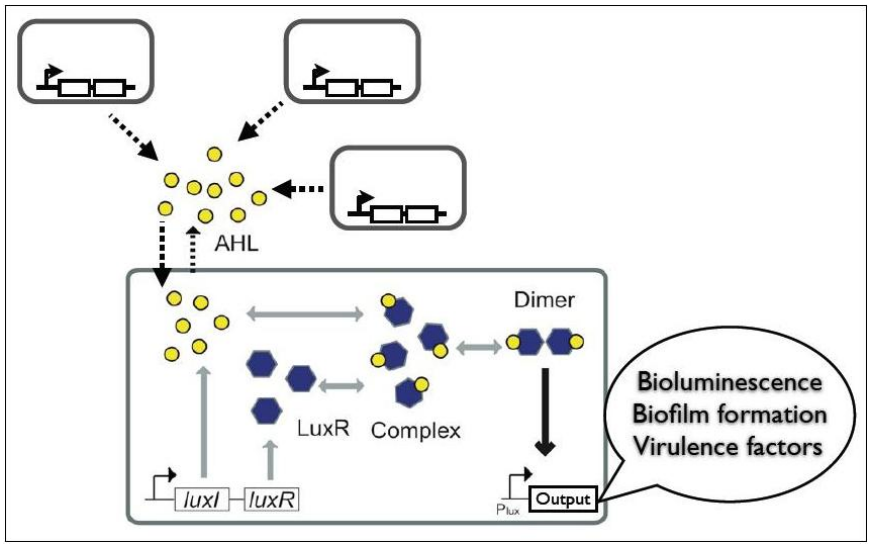

Figure 2: A QS motif

Tanouchi et al. [2008] modelled this system following the ideas of Gillespie [2000] who suggests approximate modelling of a chemical system using Langevin equations. Our QS system is modelled by the following system of *stochastic differential equations*:

$$\begin{aligned}
\frac{dA_i}{dt} &= k_A - \gamma_{A_i} A_i - k_{C_1} A_i R + k_{C_2} C - P(\frac{A_i}{V_i} - \frac{A_e}{V_e}) + \zeta_1 + \zeta_2 + \zeta_3 + \zeta_4 + \zeta_5 + \xi_{A_i}, \\
\frac{dA_e}{dt} &= -\gamma_{A_e} A_e + P(\frac{A_i}{V_i} - \frac{A_e}{V_e}) - \zeta_5 + \zeta_6, \\
\frac{dR}{dt} &= k_R - \gamma_R R - k_{C_1} A_i R + k_{C_2} C + \zeta_3 + \zeta_4 + \zeta_7 + \zeta_8 + \xi_R, \\
\frac{dC}{dt} &= -\gamma_C C + k_{C_1} A_i R - k_{C_2} C - \zeta_3 - \zeta_4 + \zeta_9 + \xi_C,
\end{aligned}$$

where A_i, A_e, R, and C are the amounts of the intracellular AHL, the extracellular AHL, the R protein, and the complex, respectively. $\gamma_{A_i}, \gamma_{A_e}, \gamma_R$, and γ_C are their decay rate constants; k_A and k_R are production rate constants of A_i and R; k_{C_1} and k_{C_2} are the association and dissociation rate of the complex; P is diffusion rate constant of the signal across the cell membrane; V_e is the average extracellular volume per cell; and V_i is an intracellular volume. ζ_i's denote intrinsic noise sources corresponding to different reactions, each having magnitude proportional to the instantaneous rate of the corresponding reaction (Table 1). ξ_{A_i}, ξ_R, and ξ_C denote extrinsic noise sources. Each extrinsic noise source is additive and its magnitude is fixed.

Because the extrinsic noise sources are fluctuations in intracellular machinery that influence the QS system, the equation for A_e does not contain a ξ term. Tanouchi et al. [2008] assumed that extrinsic noise sources are fully correlated with the same magnitude $\xi_{A_i} = \xi_R = \xi_C = \xi$.

Table 1: Reactions and Corresponding Noise Sources

Elementary Reaction	Rate Parameter(s)	Noise Source	Magnitude
AHL production	k_A	ζ_1	k_A
Internal AHL decay	γ_{A_i}	ζ_2	$\gamma_{A_i} A_i$
Complex association	k_{C_1}	ζ_3	$k_{C_1} A_i R$
Complex dissociation	k_{C_2}	ζ_4	$k_{C_2} C$
Diffusion	$\frac{P}{V_i}, \frac{P}{V_e}$	ζ_5	$P\left(\frac{A_i}{V_i} - \frac{A_e}{V_e}\right)$
External AHL decay	γ_{A_e}	ζ_6	$\gamma_{A_e} A_e$
LuxR production	k_R	ζ_7	k_R
LuxR decay	γ_R	ζ_8	$\gamma_R R$
Complex decay	γ_C	ζ_9	$\gamma_C C$
Extrinsic noise		ξ	β^2

Biological questions about this system can be formed in terms of the rate parameters. Learning these parameters from real data in itself an important exercise; moreover, questions about relevance of a reaction in governing the system dynamics can be resolved by performing model comparison. The full model is compared with a sub-model formed by dropping the corresponding reaction term(s) (equivalently, setting that parameter value to zero). In the subsequent discussion we formalize this question in statistical terms and discuss Bayesian solutions.

4.1 Discretized Model

The system of stochastic differential equations provides a way of modelling the dynamics in the QS system in terms of rate parameters. For inferential purpose we discretize them to form the following statistical model:

$$\begin{pmatrix} A_{i,t+h} \\ A_{e,t+h} \\ R_{t+h} \\ C_{t+h} \end{pmatrix} = \begin{pmatrix} A_{i,t} \\ A_{e,t} \\ R_t \\ C_t \end{pmatrix} + h \begin{pmatrix} k_A - \gamma_{A_i} A_{i,t} - k_{C_1} A_{i,t} R_t + k_{C_2} C_t - P(\frac{A_{i,t}}{V_i} - \frac{A_{e,t}}{V_e}) \\ -\gamma_{A_e} A_{e,t} + P(\frac{A_{i,t}}{V_i} - \frac{A_{e,t}}{V_e}) \\ k_R - \gamma_R R_t - k_{C_1} A_{i,t} R_t + k_{C_2} C_t \\ -\gamma_C C_t + k_{C_1} A_{i,t} R_t - k_{C_2} C_t \end{pmatrix}$$

$$+\sqrt{h} \begin{pmatrix} \zeta_{1,t} + \zeta_{2,t} + \zeta_{3,t} + \zeta_{4,t} + \zeta_{5,t} + \xi_t \\ -\zeta_{5,t} + \zeta_{6,t} \\ \zeta_{3,t} + \zeta_{4,t} + \zeta_{7,t} + \zeta_{8,t} + \xi_t \\ -\zeta_{3,t} - \zeta_{4,t} + \zeta_{9,t} + \xi_t \end{pmatrix}$$

where $h = dt$ is the time increment used for discretization. With the following notations: $x_t = (A_{i,t}, A_{e,t}, R_t, C_t)'$, $\theta = (k_A, k_R, k_{C_1}, k_{C_2}, \gamma_{A_i}, \gamma_{A_e}, \gamma_R, \gamma_C, \frac{P}{V_i}, \frac{P}{V_e}, \beta)'$ and

$$\mu(x_t, \theta) = \begin{pmatrix} A_{i,t} \\ A_{e,t} \\ R_t \\ C_t \end{pmatrix} + h \begin{pmatrix} k_A - \gamma_{A_i} A_{i,t} - k_{C_1} A_{i,t} R_t + k_{C_2} C_t - P(\frac{A_{i,t}}{V_i} - \frac{A_{e,t}}{V_e}) \\ -\gamma_{A_e} A_{e,t} + P(\frac{A_{i,t}}{V_i} - \frac{A_{e,t}}{V_e}) \\ k_R - \gamma_R R_t - k_{C_1} A_{i,t} R_t + k_{C_2} C_t \\ -\gamma_C C_t + k_{C_1} A_{i,t} R_t - k_{C_2} C_t \end{pmatrix},$$

our model can be written as

$$x_{t+h} = \mu(x_t, \theta) + w_t, \text{ where } w_t \sim \mathbb{N}_4(0, \Sigma(x_t, \theta)) \tag{5}$$

and

$$\Sigma(x_t,\theta) = h \begin{pmatrix} k_A + \gamma_{A_i} A_{i,t} & & & \\ +k_{C_1} A_{i,t} R_t + k_{C_2} C_t & -|P(\frac{A_{i,t}}{V_i} - \frac{A_{e,t}}{V_e})| & k_{C_1} A_{i,t} R_t & -k_{C_1} A_{i,t} R_t \\ +|P(\frac{A_{i,t}}{V_i} - \frac{A_{e,t}}{V_e})| + \beta^2 & & +k_{C_2} C_t + \beta^2 & -k_{C_2} C_t + \beta^2 \\ & \gamma_{A_e} A_{e,t} & & \\ & +|P(\frac{A_{i,t}}{V_i} - \frac{A_{e,t}}{V_e})| & 0 & 0 \\ & & k_R + \gamma_R R_t & \\ & & +k_{C_1} A_{i,t} R_t & -k_{C_1} A_{i,t} R_t \\ & & +k_{C_2} C_t + \beta^2 & -k_{C_2} C_t + \beta^2 \\ & & & \gamma_C C_t \\ & & & +k_{C_1} A_{i,t} R_t \\ & & & +k_{C_2} C_t + \beta^2 \end{pmatrix}.$$

Note that the parameters P, V_i and V_e are not separately identifiable in this model; we can identify only $\frac{P}{V_i}$ and $\frac{P}{V_e}$. Therefore we consider the ratios $\frac{P}{V_i}$ and $\frac{P}{V_e}$ as model parameters, not individually. The choice of h is a tricky issue - smaller values of h will ensure that the statistical model is a good approximation to the stochastic differential equations model, but comes at a cost – we have to deal with a very long time series. Due to the nature of the experiment observations can not be made very frequently; let $t = t_1, t_2, \ldots, t_K$ be the time points when observations are made. We also assume a simple additive Gaussian structure for observational noise:

$$y_t = x_t + v_t, \text{ where } v_t \sim N(0, V). \tag{6}$$

Combining (5) and (6), we have the state-space model:

$$\begin{array}{lrcll} \text{System Evolution:} & x_{t+h} & \sim & p(x_{t+h}|x_t,\theta) & = \mathbb{N}(\mu(x_t,\theta), \Sigma(x_t,\theta)), \quad \text{for } t > 0, \\ \text{Observations:} & y_t & \sim & p(y_t|x_t,\theta) & = \mathbb{N}(x_t, V), \quad \text{for } t \in t_1, t_2, \ldots, t_K. \end{array}$$

Note that this system evolution is nonlinear, and though the noise terms are Gaussian, its variance depends on the states and the parameters.

4.2 Experiments with Synthetic Data

We use a synthetic data set that corresponds to an experiment run for 10 minutes and observations are recorded at each 0.1 minute. θ parameter values used for simulation are as suggested in Tanouchi et al. [2008] – we use these values as prior means (Table 2). We choose $h = 0.01$ minute that makes $T = 1000$ with observations available at $t_1 = 10, t_2 = 20, \ldots, t_K = 1000$. For simplicity we use independent observational noise for A_i, A_e, R and C with signal-to-noise ratio roughly 1 for each term.

4.2.1 Prior Distributions

We use truncated normal distribution as independent priors for the model parameters θ_j and the initial states $x_{0,k}$ since they represent rates of some reactions and number of molecules of some biochemical species respectively:

$$\pi(\theta_j) = \mathbb{N}_+(m_{\theta_j}, s^2_{\theta_j}) \text{ for } j = 1, \ldots, 11 \text{ and } \pi(x_{0,k}) = \mathbb{N}_+(m_{x_{0,k}}, s^2_{x_{0,k}}) \text{ for } k = 1, 2, 3, 4.$$

We use the following values for prior means based on the prior information gathered in Tanouchi et al. [2008]. Priors are relatively uninformative based on coefficients of variation set to 0.5. For the observational variance we use $\pi(V) = \textit{Inverse-Wishart}\ (s, S)$ with a diagonal S corresponding to the independent observational noise terms that were used for generating the synthetic data. We set the degrees of freedom to $s = 10$.

Table 2: Prior Specification

Mean	Value	Std	Value
m_{k_A}	842 molecules/min	s_{k_A}	$m_{k_A}/2$
$m_{\gamma_{A_i}}$	0.023/min	$s_{\gamma_{A_i}}$	$m_{\gamma_{A_i}}/2$
$m_{k_{C_1}}$	0.1 molecules/min	$s_{k_{C_1}}$	$m_{k_{C_1}}/2$
$m_{k_{C_2}}$	1/min	$s_{k_{C_2}}$	$m_{k_{C_2}}/2$
m_{P/V_i}	1250/min	s_{P/V_i}	$m_{P/V_i}/2$
m_{P/V_e}	2.5031e-4/min	s_{P/V_e}	$m_{P/V_e}/2$
$m_{\gamma_{A_e}}$	0.0018/min	$s_{\gamma_{A_e}}$	$m_{\gamma_{A_e}}/2$
m_{k_R}	20 molecules/min	s_{k_R}	$m_{k_R}/2$
m_{γ_R}	0.2/min	s_{γ_R}	$m_{\gamma_R}/2$
m_{γ_C}	0.02/min	s_{γ_C}	$m_{\gamma_C}/2$
m_β	7	s_β	$m_\beta/2$
$m_{A_{i,0}}$	10 molecules	$s_{A_{i,0}}$	$m_{A_{i,0}}/2$
$m_{A_{e,0}}$	293477 molecules	$s_{A_{e,0}}$	$m_{A_{e,0}}/2$
m_{R_0}	90 molecules	s_{R_0}	$m_{R_0}/2$
m_{C_0}	88 molecules	s_{C_0}	$m_{C_0}/2$

4.2.2 MCMC

Updating $(V|x_{0:T}, \theta, y_{t_1,t_2,\ldots,t_K})$ can be done with a Gibbs step as the complete conditional takes a conjugate inverse-Wishart form. θ is linear in the mean $\mu(x_t, \theta)$ but it appears in the likelihood also through the covariance matrices $\Sigma(x_t, \theta)$ – therefore sampling from the conditional $p(\theta|x_{0:T}, V, y_{t_1,t_2,\ldots,t_K})$ is not straightforward. We update each $(\theta_j|\theta_{(-j)}, x_{0:T}, V, y_{t_1,t_2,\ldots,t_K})$ using random-walk Metropolis-Hastings steps.

For updating the states we note that $\mathbb{E}[(A_{i,t}, A_{e,t}, C_t)|(A_{i,t-1}, A_{e,t-1}, C_{t-1}), R_{0:T}, \theta]$ is linear in $(A_{i,t-1}, A_{e,t-1}, C_{t-1})$ as is $\mathbb{E}[R_t|R_{t-1}, (A_{i,0:T}, A_{e,0:T}, C_{0:T}), \theta]$. We sequentially update

$$(A_{i,0:T}, A_{e,0:T}, C_{0:T}|R_{0:T}, \theta, V, y_{t_1,t_2,\ldots,t_K}) \quad \text{and} \quad (R_{0:T}|A_{i,0:T}, A_{e,0:T}, C_{0:T}, \theta, V, y_{t_1,t_2,\ldots,t_K})$$

with Metropolis-Hastings steps that use approximate FFBS proposals. Linearity makes forward filtering trivial – backward sampling needs some form of approximation to

$$\mathbb{V}[(A_{i,t}, A_{e,t}, C_t)|(A_{i,t-1}, A_{e,t-1}, C_{t-1}), R_{0:T}, \theta] \quad \text{and} \quad \mathbb{V}[R_t|R_{t-1}, (A_{i,0:T}, A_{e,0:T}, C_{0:T}), \theta].$$

Even though this state updating scheme performs better than the many other approximate FFBS methods, we observe that the approximation degrades quickly as the length of the underlying latent time series $x_{0:T}$ increases (Figure 3).

Table 3: Effective Sample Size for Estimation of Parameters

Parameters	k_A	k_R	k_{C_1}	k_{C_2}	γ_{A_i}	γ_{A_e}	γ_R	γ_C	$\frac{P}{V_i}$	$\frac{P}{V_e}$	β
Effective Sample Size	414	22135	1146	1347	4960	344	1208	1776	248	267	274

Table 3 shows the effective sample sizes for estimating parameter θ_j based on the MCMC chain run for $100,000$ iterations after a burn-in of $10,000$ steps when $T = 1000$. Note that for some parameters the effective sample size is as small as 248. Clearly MCMC takes too long to produce a representative sample from the posterior. Therefore marginal likelihood estimates based on MCMC samples are unreliable because they are based on such poor effective sample size.

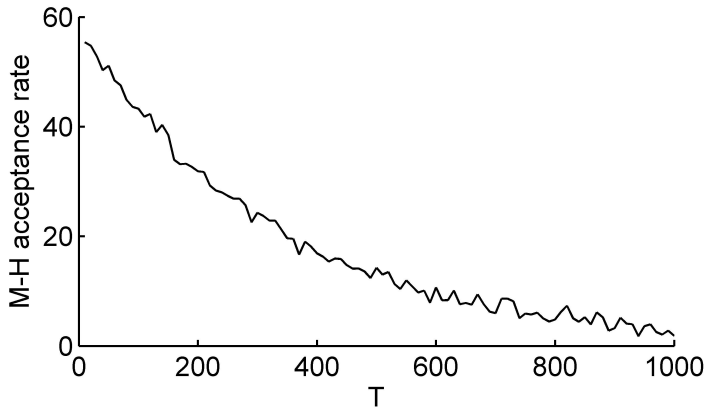

Figure 3: Metropolis-Hastings acceptance rates for updating $A_{i,0:T}, A_{e,0:T}, C_{0:T}$ given $R_{0:T}, \theta, V, y_{t_1,t_2,\ldots,t_K}$. For each T we run a MCMC sampler for 100,000 iterations after a burn in of 10,000 steps.

4.2.3 SMC

We ran the particle filter with 100 million particles on 100 processors on the *Shared Cluster Resource* of Duke University. For the distributed implementation we use MPI with C++. Figure 4 shows the effective sample size calculated using $ESS_t = \left[\sum_{i=1}^{M} {W_t^{(i)}}^2\right]^{-1}$ where $W_t^{(i)}$ is the normalized weight of particle i at time t. We note that we can achieve a much higher effective sample size with the particle filter compared to the MCMC algorithm. Figure 5 displays medians, upper and lower 95% quantiles from the SMC samples for each of the 11 parameters.

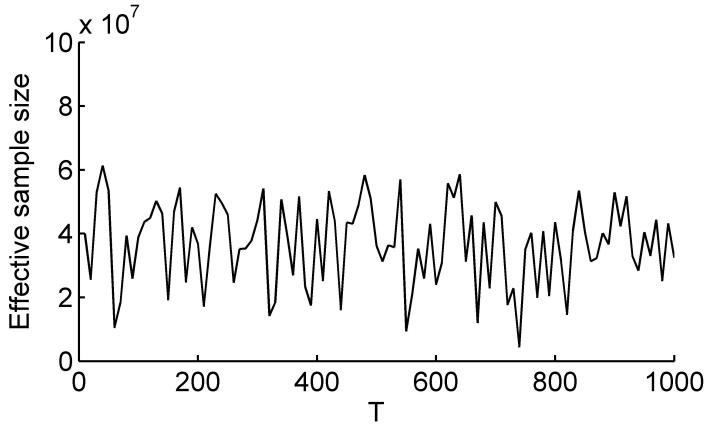

Figure 4: Effective sample size of the particle filter for the QS Model with 100 million particles

Now focus on the model comparison question. We are interested in evaluating the relevance of individual reactions in governing the QS model dynamics. Consider the full QS Model (M_0) and 11 sub-models ($M_1, M_2, \ldots, M_{11}$) formed by setting $\theta_j = 0$ for $j = 1, 2, \ldots, 11$ respectively in the full model. Given the data $y_{t_1,t_2,\ldots,t_k}$ we perform model comparison using

$$p(M_j|y_{t_1,t_2,\ldots,t_k}) = \frac{\pi(M_j)\ p_{M_j}(y_{t_1,t_2,\ldots,t_k})}{\sum_{l=1}^{11} \pi(M_l)\ p_{M_l}(y_{t_1,t_2,\ldots,t_k})} \text{ for } j = 0, 1, \ldots, 11.$$

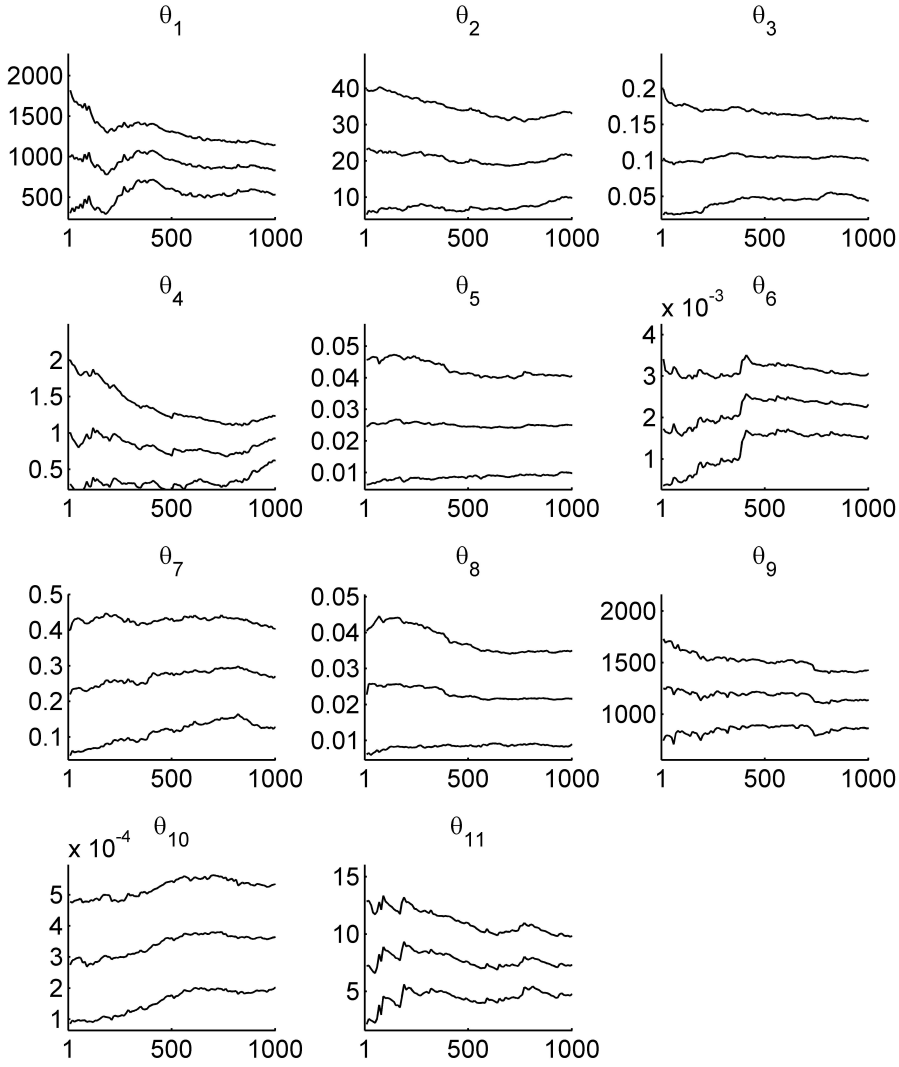

Figure 5: Plots of 95% upper and lower posterior quantiles and median against T for each parameter θ_j of the QS model.

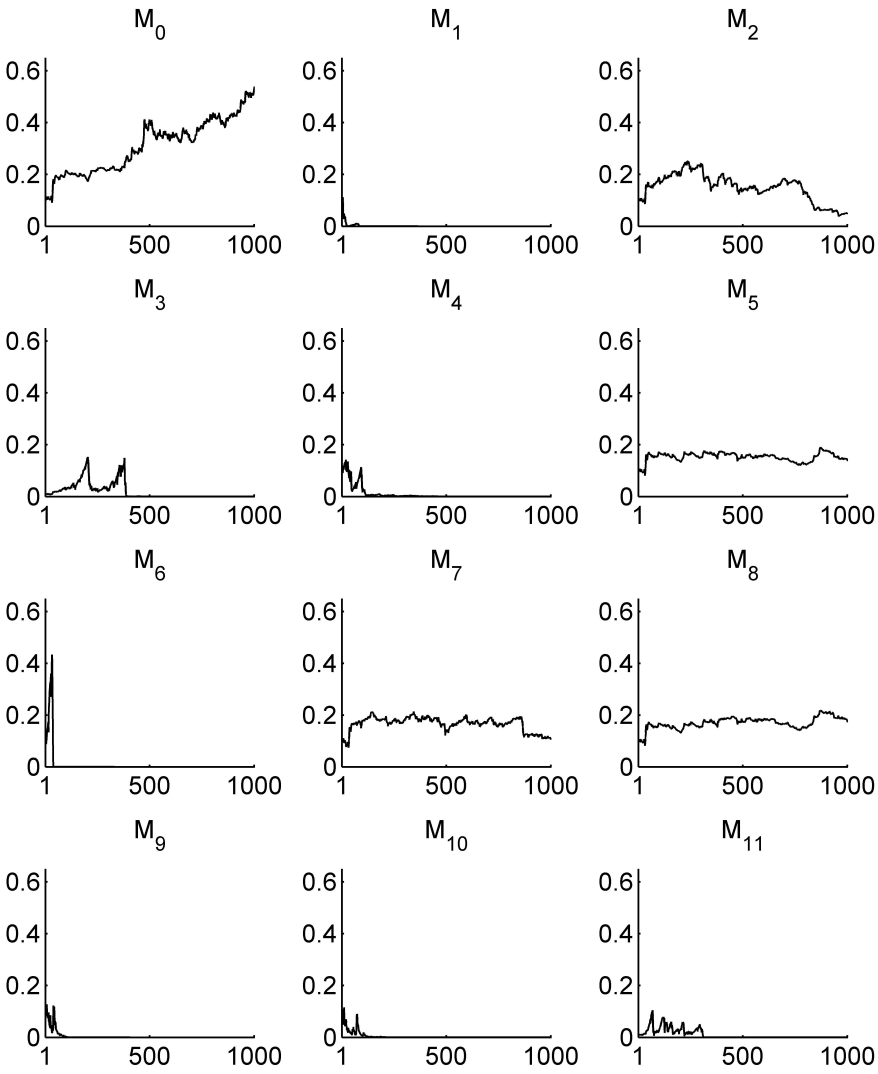

Figure 6: Posterior model probabilities for $M_0, M_1, M_2, \ldots, M_{11}$

Here $\pi(M_j)$ are the prior probabilities assigned to model M_j and $p_{M_j}(y_{t_1,t_2,\ldots,t_k})$ the *marginal likelihood* of the data under model M_j. We start with equal priorities for each of these 12 models (i.e. $\pi(M_j) = 1/12$ for $j = 0, 1, 2, \ldots, 11$) and run a separate particle filter for each model. Figure 6 shows the posterior model probabilities based on the easy marginal likelihood estimates we obtain from the particle filter.

Note that the posterior model probabilities of M_1, M_4, M_6, M_9 and M_{10} become negligible very quickly. With as few as 20 observations the full model is preferred compared to these sub-models, which means the data contains strong information about the relevance of non-zero values of the corresponding parameters and hence support the inclusion in the network model of the corresponding reaction terms. For models M_3 and M_{11} the posterior probabilities take longer to vanish, and for the model M_2 we observe a decreasing trend though it remains significant. So, a larger amount of data enables learning about aspects of the relevance of some of the network model components represented, and included in the model, by non-zero parameters. This example highlights the use and ability of SMC-based computational Bayesian methods to directly yield formal model assessment focussed on "modules" of network structure represented by a dynamic mechanistic model. As single cell experimental technologies advance rapidly in the coming few years, the role and relevance of formal statistical methods – and model assessment and comparison in particular – will increase due to access to increasingly rich data, and we expect such approaches to play critical roles in advancing systems biology.

5. Acknowledgements

We are grateful to Lingchong You and Yu Tanouchi of Duke University for discussions of models in systems biology and provision of the quorum sensing model and simulations, and to Jarad Niemi for discussions of statistical and computational issues in dynamic models. Research was partially supported by grants to Duke University from the NSF (DMS-0342172) and the National Institutes of Health (grants P50-GM081883-01 and NCI U54-CA-112952-01). Aspects of the research were also partially supported by the NSF grant DMS-0635449 to the Statistical and Applied Mathematical Sciences Institute. Any opinions, findings and conclusions or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of the NSF or NIH.

References

S. Bornholdt. Systems biology: Less is more in modeling large genetic networks. *Science*, 310: 449–451, 2005.

C. K. Carter and R. Kohn. On Gibbs sampling for state-space models. *Biometrika*, 82:339–350, 1994.

C. M. Carvalho, M. Johannes, H. F. Lopes, and N. Polson. Particle learning and smoothing. Technical report, Department of Statistical Science, Duke University, 2009.

P. Del Moral. *Feynman-Kac formulae: Genealogical and Interacting Particle Systems with Applications*. New York: Springer-Verlag, 2004.

M.B. Elowitz, A.J. Levine, E.D. Siggia, and P.S. Swain. Stochastic gene expression in a single cell. *Science*, 297:1183–1186, 2002.

P. Fearnhead. MCMC, sufficient statistics, and particle filters. *Journal of Computational and Graphical Statistics*, 11:848–862, 2002.

S. Frühwirth-Schnatter. Applied state-space modelling of non-gaussian time series using integration-based Kalman filtering. *Statistics and Computing*, 4:259–269, 1994.

D. T. Gillespie. The chemical Langevin equation. *Journal of Chemical Physics*, 113:297–306, 2000.

M. Johannes and N. Polson. Particle filtering and parameter learning. *Social Science Research Network*, 2007.

J. Liu and M. West. Combined parameter and state estimation in simulation-based filtering. In A. Doucet, J.F.G. De Freitas, and N.J. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, pages 197–217. New York: Springer-Verlag, 2001.

N. Rosenfeld, J.W. Young, U. Alon, P.S. Swain, and M.B. Elowitz. Gene regulation at the single-cell level. *Science*, 307:1962–1965, 2005.

M. Shabany and P.G. Gulak. An efficient architecture for distributed resampling for high-speed particle filtering. *IEEE International Symposium on Circuits and Systems*, 2006.

G. Storvik. Particle filters in state-space models with the presence of unknown static parameters. *IEEE. Trans. of Signal Processing*, 50:281–289, 2002.

C.M. Tan, H. Song, J. Niemi, and L. You. A synthetic biology challenge: making cells compute. *Molecular BioSystems*, 3(5):343–353, 2007. doi: 10.1039/b618473c.

Y. Tanouchi, D. Tu, J. Kim, and L. You. Noise reduction by diffusional dissipation in a minimal quorum sensing motif. *PLoS Computational Biology*, 4(8):e1000167, August 2008.

Q. Wang, J. Niemi, C.M. Tan, L. You, and M. West. Image segmentation and dynamic lineage analysis in single-cell flourescence microscopy. *Cytometry A*, 2009. To appear.

D. J. Wilkinson. *Stochastic Modelling for Systems Biology*. London: Taylor & Francis/Chapman & Hall/CRC, 2006.