Pacific Symposium on Biocomputing 14:227-238 (2009)

PARAMETER ESTIMATION OF *IN SILICO* BIOLOGICAL PATHWAYS WITH PARTICLE FILTERING TOWARDS A PETASCALE COMPUTING

KAZUYUKI NAKAMURA[1,*], RYO YOSHIDA[1], MASAO NAGASAKI[2], SATORU MIYANO[2] AND TOMOYUKI HIGUCHI[1]

[1] *The Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-ku, Tokyo 1068569, Japan*

[2] *The Institute of Medical Science, The University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 1088639, Japan*

The aim of this paper is to demonstrate the potential power of large-scale particle filtering for the parameter estimations of *in silico* biological pathways where time course measurements of biochemical reactions are observable. The method of particle filtering has been a popular technique in the field of statistical science, which approximates posterior distributions of model parameters of dynamic system by using sequentially-generated Monte Carlo samples. In order to apply the particle filtering to system identifications of biological pathways, it is often needed to explore the posterior distributions which are defined over an exceedingly high-dimensional parameter space. It is then essential to use a fairly large amount of Monte Carlo samples to obtain an approximation with a high-degree of accuracy. In this paper, we address some implementation issues on large-scale particle filtering, and then, indicate the importance of large-scale computing for parameter learning of *in silico* biological pathways. We have tested the ability of the particle filtering with 10^8 Monte Carlo samples on the transcription circuit of circadian clock that contains 45 unknown kinetic parameters. The proposed approach could reveal clearly the shape of the posterior distributions over the 45 dimensional parameter space.

1. Introduction

Mathematical modeling of biological pathways has broad utility to understanding of complex networks of biochemical reactions in living cells. In systems biology, *in silico* modeling of biological pathways has proven to be

*to whom correspondence should be addressed: nakakazu@ism.ac.jp

Pacific Symposium on Biocomputing 14:227-238 (2009)

a promising approach with successful applications.[1–4] Usually, a pathway model is comprised of a set of differential[1,5,6] or difference[2–4,7,8] equations that describe kinetic mechanism between biochemical reactants, e.g. mRNA and protein. In order to conduct simulation experiments, it is essential to find a set of model parameters and initial concentrations of reactants such that the solution fits observations. For example, one may find effective values of rate constant parameters for which time course microarray data are available.

More recently, several approaches have been proposed as a technique of estimating parameters of *in silico* pathway, e.g. genetic algorithm,[8] unscented Kalman filter,[6] and gradient methods.[5] We also have been working on this topic based on the method of particle filtering,[10–13] which has been a popular technique for the problem of system identifications in the field of statistical science and signal processing. While these existing methods could cope with the estimation of relatively small number of parameters, their ability is limited by the case that model contains much larger number of unknown parameters.

As an illustration, consider the use of particle filtering that approximates the posterior distributions of model parameters based on a set of Monte Carlo samples. The size of random numbers usually lies in 10^3 to 10^5.[13] As the number of unknown parameters becomes large, it is required to generate an exceedingly large number of Monte Carlo samples in order to cover the overall parameter space. Such a problem, often called *curse of dimensionality*, is also common in the existing methods.[5,6,8] For example, genetic algorithm explores values of parameters by alternating cross-over and mutation of initially-generated seeds for unknown parameters.[8] In its applications, it is a key to success whether or not an initial set of candidate parameters lies in the region close to the optimal values.

To address such a limitation, we adopt the direct way that uses an exceedingly large amount of Monte Carlo samples, e.g. of order 10^8 or more, for improving ability of particle filtering. In our previous works,[14,15] we applied the particle filtering with $10,000$ Monte Carlo samples to the parameter estimation of transcription network of circadian clock. Unfortunately, the number of parameters that were estimated was limited to only a few. However, as will be demonstrated, by increasing the number of samples to 10^8 particles, the ability of the particle filtering has been extended greatly so that 45 unknown parameters could be estimated with a high degree of accuracy. Particularly, the large-scale particle filtering could reveal clearly the shape of the posterior distributions over the 45 parameters. In

Pacific Symposium on Biocomputing 14:227-238 (2009)

doing so, we aim to obtain an overall view of high-dimensional parameter space of *in silico* biological pathways, e.g. multi-modality and higher-order moments.

One more attractive feature of the particle filtering is a high degree of compatibility with parallel computing. This favorable feature increases the potential of particle filtering such that we can use a larger number of parameters to deal with higher dimensional parameter space, e.g. of order 10^2, by exploiting the parallel computation schemes and computational resources which will be available in the near future.

In this paper, we first give a brief explanation about the method of the particle filtering, and then, address some computational problems that arise from the implementation of large-scale particle filtering. The development of the 10 petaflops supercomputer has been going on as a national project. It mainly targets high performance computing of life science simulators, including molecular dynamics, biomechanics and biochemical simulations. This work is a pilot study towards the estimation of large-scale *in silico* models by the aid of petascale computing.

In Section 2, we show the method of particle filtering after introducing the state space model that relates *in silico* pathway models to time course measurements. We also discuss some computational issues on the large-scale particle filtering. Section 3 shows the numerical experiments which test the applicability of the proposed approach on the analysis of transcription network of circadian clock. Finally, discussion and conclusion are given in Section 4.

2. Method

2.1. *Nonlinear State Space Model*

Suppose that we model successive change in concentration levels of biochemical reactants by a set of rate equations

$$\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_{t-1}, \boldsymbol{\theta}), \tag{1}$$

where the p dimensional vector $\boldsymbol{x}_t$ contains concentration levels of all variables, e.g. mRNAs and proteins, at time step t and $\boldsymbol{\theta}$ denotes a vector of model parameters. The state variable $\boldsymbol{x}_t$ which is assumed to be unobservable evolves during successive discrete time points from $t-1$ to t according to the p dimensional vector-valued function $\boldsymbol{f}$. For ordinary differential equations, several methods are available to be associated with (1), for example, discretization in time. [9]

To deal with an inevitable modeling error, we may rather use the stochastic difference equations, referred to as system model,

$$\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) + \boldsymbol{v}_t \,, \tag{2}$$

where the noise component $\boldsymbol{v}_t$ is added to (1). In what follows, we use the general form (2) which contains (1) as a special case.

The measurement model, referred to as observation model, associates the simulation model to time course data:

$$\boldsymbol{y}_t = \boldsymbol{h}(\boldsymbol{x}_{t-1}) + \boldsymbol{w}_t \,. \tag{3}$$

Here, $\boldsymbol{h}(\cdot)$ represents the function of measurement mechanism, which defines the conversion from unobserved concentration level $\boldsymbol{x}_t$ to observed time course data $\boldsymbol{y}_t$. $\boldsymbol{w}_t$ denotes additive measurement noise. Initial state variables $\boldsymbol{x}_0$ are assumed to follow a certain distribution $p(\boldsymbol{x}_0)$. The probabilistic model consisting of (2) and (3) is referred to as the nonlinear state space model.[10–12]

Of interest is then to find values of parameters, $\boldsymbol{\theta}$, and unobserved concentrations of reactants, $\boldsymbol{x}_t$, based on time course data $\boldsymbol{y}_t, t = 1, \cdots, T$. This paper focuses on the Bayesian estimations of model parameters, which are based on the joint posterior distribution of $\boldsymbol{x}_t$, $t = 0, \cdots, T$ and $\boldsymbol{\theta}$ conditional on the observed data. However, the Bayesian estimations usually require us to approximate the posterior distributions because their closed forms are usually unavailable due to nonlinearity of models $\boldsymbol{f}$.[13,16]

2.2. *Particle Filtering*

The particle filtering is a sequential Monte Carlo technique [10–12] which approximates a joint posterior distribution of state vectors and parameters

$$p(\boldsymbol{x}_{0:T}, \boldsymbol{\theta} | \boldsymbol{y}_{1:T}) \propto p(\boldsymbol{y}_{1:T} | \boldsymbol{x}_{1:T}) p(\boldsymbol{x}_{1:T} | \boldsymbol{\theta}, \boldsymbol{x}_0) p(\boldsymbol{x}_0) p(\boldsymbol{\theta}) \,, \tag{4}$$

where $p(\boldsymbol{x}_0)$ and $p(\boldsymbol{\theta})$ denote the prior distributions of initial concentrations and model parameters, respectively. Here, sets of state vectors and observations up to time t are denoted respectively by $\boldsymbol{x}_{0:t} = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$ and $\boldsymbol{y}_{1:t} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_t\}$.

Denote N Monte Carlo samples from the conditional distribution $p(\boldsymbol{x}_t, \boldsymbol{\theta} | \boldsymbol{y}_{1:k})$ by $(\boldsymbol{x}_{t|k}^{(i)}, \boldsymbol{\theta}_k^{(i)})$, $i = 1, \cdots, N$. Then, the procedure of the particle filtering is summarized as follows:

- Set $t \leftarrow 1$. Generate N Monte Carlo samples $\{\boldsymbol{x}_{0|0}^{(i)}, \boldsymbol{\theta}_0^{(i)}\}_{i=1}^N$ from the prior distributions of initial state variables $p(\boldsymbol{x}_0)$ and parameter vector $p(\boldsymbol{\theta})$.

Pacific Symposium on Biocomputing 14:227-238 (2009)

- ($\sharp$) Perform the following procedure:
  - (Prediction step) After drawing $\{\boldsymbol{v}_t^{(i)}\}_{i=1}^N$, generate N prediction samples of state variables $\{\boldsymbol{x}_{t|t-1}^{(i)}, \boldsymbol{\theta}_{t-1}^{(i)}\}_{i=1}^N$ according to $\boldsymbol{x}_{t|t-1}^{(i)} = \boldsymbol{f}(\boldsymbol{x}_{t-1|t-1}^{(i)}, \boldsymbol{\theta}_{t-1}^{(i)}) + \boldsymbol{v}_t^{(i)}$ for $i = 1, \cdots, N$.
  - ($\star$) (Filtering step) Execute the following procedure:
    (i) Compute likelihood function of each particle $p(\boldsymbol{y}_t|\boldsymbol{x}_{t|t-1}^{(i)})$, $i = 1, \cdots, N$.
    (ii) Calculate the normalizing weights $q_t^{(i)} \propto p(\boldsymbol{y}_t|\boldsymbol{x}_{t|t-1}^{(i)})$ for $i = 1, \cdots, N$ such that $\sum_{i=1}^N q_t^{(i)} = 1$.
    (iii) Generate $\{\boldsymbol{x}_{t|t}^{(i)}, \boldsymbol{\theta}_t^{(i)}\}_{i=1}^N$ by the resampling of $\{\boldsymbol{x}_{t|t-1}^{(i)}, \boldsymbol{\theta}_{t-1}^{(i)}\}_{i=1}^N$ with the probabilities $\{q_t^{(i)}\}_{i=1}^N$.
- If t is equal to T, stop the algorithm. Otherwise, set $t \leftarrow t+1$ and go to ($\sharp$).

Here, the likelihood function $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ is defined by the probability density function, e.g. normal distribution, corresponding to the observation model (3). The final draws of $\{\boldsymbol{\theta}_T^{(i)}\}_{i=1}^N$ generated from this procedure are approximately distributed according to the augmented posterior distribution $p(\boldsymbol{x}_{0:T}, \boldsymbol{\theta}|\boldsymbol{y}_{1:T})$. Indeed, it can be proved that the empirical distributions given by N realization $\{\boldsymbol{\theta}_T^{(i)}\}_{i=1}^N$ approach to the true distribution $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$ with probability one as N goes to infinity.[13]

As the number of unknown parameters becomes large, the method requires to generate an exceedingly large number of Monte Carlo samples in order to cover the overall parameter space. For example, the transcription regulation model of circadian clock that will be shown in Section 3 contains 45 unknown parameters in total. Particularly, the problem of "*particle degeneracy*" is caused by a small number of Monte Carlo samples (Chapter 14 of Ref. 13). This is due to the fact that during the repeated resampling of initially generated $\{\boldsymbol{\theta}_0^{(i)}\}_{i=1}^N$ drawn from $p(\boldsymbol{\theta})$, the number of distinct values in the resampled particles can diminish drastically as the time step increases. Hence, in order to estimate a large number of parameters, it is essential to use a fairly large amount of Monte Carlo samples.

Both the time complexity and space complexity of the prediction step are linear order of the number of particles. On the other hand, the time complexity of the filtering step is $O(N \log N)$ because the resampling algorithm requires $O(\log N)$ evaluations of weight table to yield one filtered particle (see Appendix A for details). We employ this type of resampling procedure in the numerical experiment. Rather than performing the ran-

Pacific Symposium on Biocomputing 14:227-238 (2009)

dom resampling algorithms, the filtering step can be replaced by the resampling algorithm with the proportional-allocation that yields the filtered samples by increasing and decreasing the number of prediction samples proportional to their weights. In that case, the time complexity can be reduced to linear order.

Here, we briefly discuss the low coupling property of the particle filtering. In the prediction step, each predictive sample is updated independently. In the likelihood calculation (i) at the filtering step, each likelihood can be also computed independently. As a result, the tasks in each step can be divided into parallel processors. On the other hand, the processes of normalization (ii) and resampling (iii) require additional parallel computing technique in order to manage the communication among N Monte Carlo samples. In the step (ii), it is necessary that the likelihood values of N samples are collected into a root CPU. To implement a parallel computing for such interacted tasks, we can use the divide and conquer algorithm.[17] By the aid of it, time complexity of normalization and resampling can be reduced up to $O((N/M)\log M)$ where M CPU processors are available, which indicates scalability of the particle filtering in parallel computing.

3. Numerical Experiments

3.1. *Implementation*

Circadian clocks of living cells are controlled by oscillating feedback loop in the transcription circuits of clock-control genes.[2,19] The model that we consider was constructed by incorporating the regulatory mechanisms among the five mRNAs (per, cry, rev/erb, clock, bmal), the translated proteins (PER, CRY, REV/ERB, CLOCK, BMAL) and the two protein complexes (PER/CRY, CLOCK/BMAL), which were reported by Ref. 18. It consists of the 12 first-order stochastic difference equations as,

$$
\begin{aligned}
x_{1,t+1} &= x_{1,t} - k_{d_1}x_{1,t} + k_{tc_1}I(x_{5,t} < s_{1,5})I(x_{12,t} > s_{1,12}) + k_{c_1} + v_{1,t+1},\\
x_{2,t+1} &= x_{2,t} - k_{d_2}x_{2,t} + k_{tl_2}x_{1,t} - k_{b_{2,4}}x_{2,t}x_{4,t} + v_{2,t+1},\\
x_{3,t+1} &= x_{3,t} - k_{d_3}x_{3,t} + k_{tc_3}I(x_{5,t} < s_{3,5})I(x_{12,t} > s_{3,12}) + k_{c_3} + v_{3,t+1},\\
x_{4,t+1} &= x_{4,t} - k_{d_4}x_{4,t} + k_{tl_4}x_{3,t} - k_{b_{2,4}}x_{2,t}x_{4,t} + v_{4,t+1},\\
x_{5,t+1} &= x_{5,t} - k_{d_5}x_{5,t} + k_{b_{2,4}}x_{2,t}x_{4,t} + v_{5,t+1},\\
x_{6,t+1} &= x_{6,t} - k_{d_6}x_{6,t} + k_{tc_6}I(x_{5,t} < s_{6,5}) + k_{c_6} + v_{6,t+1},\\
x_{7,t+1} &= x_{7,t} - k_{d_7}x_{7,t} + k_{tl_7}x_{6,t} + v_{7,t+1},\\
x_{8,t+1} &= x_{8,t} - k_{d_8}x_{8,t} + k_{c_8} + v_{8,t+1},
\end{aligned}
$$

$$
\begin{aligned}
x_{9,t+1} &= x_{9,t} - k_{d_9} x_{9,t} + k_{tl_9} x_{8,t} - k_{b_{9,11}} x_{9,t} x_{11,t} + v_{9,t+1}, \\
x_{10,t+1} &= x_{10,t} - k_{d_{10}} x_{10,t+1} + k_{tc_{10}} I(x_{5,t} > s_{10,5}) I(x_{7,t} < s_{10,7}) \\
&\quad + k_{c_{10}} + v_{10,t+1}, \\
x_{11,t+1} &= x_{11,t} - k_{d_{11}} x_{11,t} + k_{tl_{11}} x_{10,t} - k_{b_{9,11}} x_{9,t} x_{11,t} + v_{11,t+1}, \\
x_{12,t+1} &= x_{12,t} - k_{d_{12}} x_{12,t} + k_{b_{9,11}} x_{9,t} x_{11,t} + v_{12,t+1}, \qquad (5)
\end{aligned}
$$

in order corresponding to per, PER, cry, CRY, PER/CRY, rev/erb, REV/ERB, clock, CLOCK, bmal, BMAL, and CLOCK/BMAL, respectively. Here, $v_{i,t}$, $i = 1, \cdots, 12$, stand for the Gaussian noises with mean 0 and variance 2.0×10^{-7}. Of interest is then to estimate the 26 rate constant parameters $k_{\cdot}$, the 7 threshold parameters $s_{\cdot}$, and the initial concentrations of 12 reactants $x_{i,0}$, $i = 1, \cdots, 12$. Note that we did not estimate values of k_{d_8} and k_{c_8} for the reason shown below.

Ueda *et al.*[19] studied the transcription regulations of mammalian circadian clocks based on the time course gene expression data which were generated with Affymetrix mouse high-density oligonucleotide probe array. Time course data were measured at equally-spaced 12 time points during 44 hours (about two days). We extracted gene expression measurements of the five mRNAs (per, cry, rev/erb, clock, bmal) from Figure 1a and the supplementary data (`http://sirius.cdb.riken.jp/MouseLiver/MouseLiverCCG(020707).html` [19]), where the set of 12 time points were allocated to the 700 simulation time steps by the conduction of time alignment. Note that in the current model (5), there are no regulatory mechanisms which yield oscillating concentrations of clock mRNA. As a possible solution, we decided to treat the measurements of clock mRNA concentrations as an input sequence. To this end, we interpolated the observed 12 data points of clock by using spline functions, and then, the interpolated values were substituted into the corresponding state variables which remained to be fixed during the parameter estimation process.

Figure 1 shows the observed time course of gene expression profiles and the result of simulation which was generated according to (5) with the predetermined parameters without inclusion of system noises $v_{i,t}$s. This figure shows inconsistency in periods, phases and amplitude among the simulation paths and observed profiles. The objective of this numerical experiment is to capture the observed oscillating patterns of circadian rhythm by the application of the particle filtering where the currently obtained parameters have the bias in the cyclic behavior.

Before proceeding to the particle filtering, the observation noise $\boldsymbol{w}_t$ was set as the normal distribution $N(0, 0.64)$ for per mRNA and $N(0, 0.25)$

Pacific Symposium on Biocomputing 14:227-238 (2009)

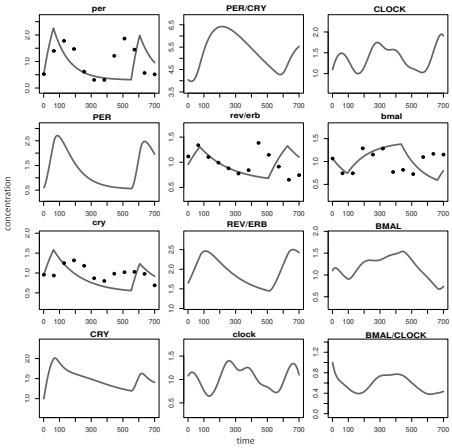

Figure 1. Result of initial simulation which was generated according to (5) with the pre-determined parameters without inclusion of system noises $v_{i,t}$s. The observed 12 data points are indicated by dots.

for the other reactants. We adopted the values that are employed in the simulation shown in Figure 1 as the prior mean of the parameter $\boldsymbol{\theta}$ and the initial state $\boldsymbol{x}_0$. Because the parameters and the states should be non-negative, the prior distributions were set to truncated normal distributions. The prior variances of initial concentration $x_{\cdot,0}$, coefficients $k_{\cdot}$ and threshold values $s_{\cdot}$ were set to 0.01, 0.001 and 0.05, respectively.

In order to realize the implementation of the particle filtering with 10^8 samples, all particles were divided into 20 sets of 5×10^6 particles, and allocated into 20 computational units. After repeating the particle filtering with 5×10^6 samples for each computational unit, which yields $\{\boldsymbol{\theta}_T^{(i,k)}\}_{i=1}^{10^6}$ for $k = 1, \cdots, 20$, we can compute the estimator by $\hat{\boldsymbol{\theta}} = \frac{1}{10^8} \sum_{k=1}^{20} \sum_{i=1}^{5\times 10^6} \boldsymbol{\theta}_T^{(i,k)}$. Due to the independence of the computational units, the integrated estimator $\hat{\boldsymbol{\theta}}$ still preserves the statistical consistency.

3.2. *Results*

Figure 2 shows the histograms of Monte Carlo samples drawn from the posterior distribution of the parameters k_{d_i} $(i = 1, \ldots, 12)$, where the sizes of Monte Carlo samples were set to 10^8 and 10^5, respectively. It is observed from the right panel of Figure 2 that the number of distinct values declines drastically for the estimation with 10^5 particles. Obviously, we would not obtain any efficient estimates by using such degenerated particles. On the

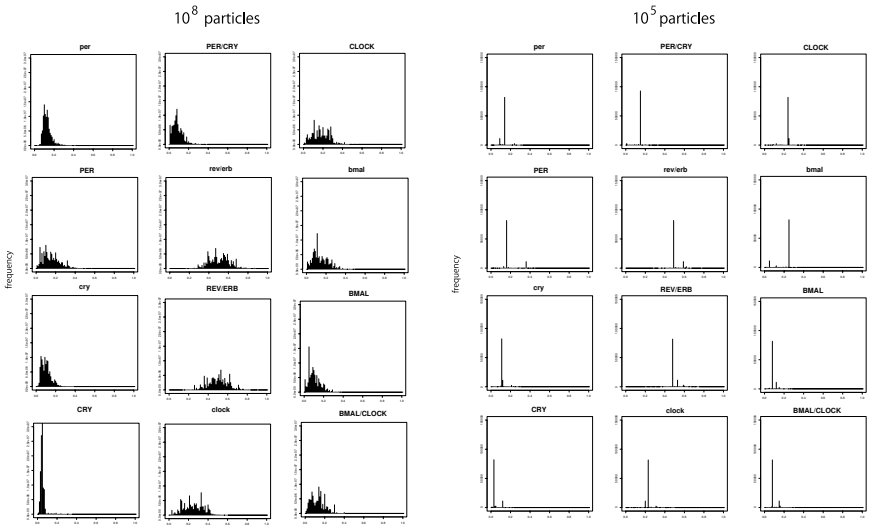

Figure 2. Approximated posterior distributions of the 12 "degradation rate" $k_{d.}$, which were generated by using 10^8 (left) and 10^5 (right) particles.

other hand, the ability of the particle filtering was improved greatly by the use of 10^8 particles as the histograms of the Monte Carlo samples reveal the shapes of posterior distributions clearly as shown in the left panel of Figure 2.

Figure 3 shows the results of simulations based on the estimated parameters and initial state variables, which were computed as the averages of 10^8 and 10^5 particles, respectively. In the case of 10^8 particles, simulation paths were generated from the estimated parameters in each of the 20 computational units, and then, the generated 20 paths were averaged in order to compute the final estimates shown in Figure 3. The observed data points are also shown in the Figure 3. The simulated concentrations corresponding to 10^5 particles preceded the observed periodicity of circadian rhythm whereas the difference in periodicity has been reduced greatly by using 10^8 particles compared with the initially-chosen parameters (prior means).

3.3. *Estimation of computational time*

We evaluate computational time for execution of large-scale particle filtering on upcoming peta-scale supercomputers.

We measured and estimated CPU times in the following three cases:

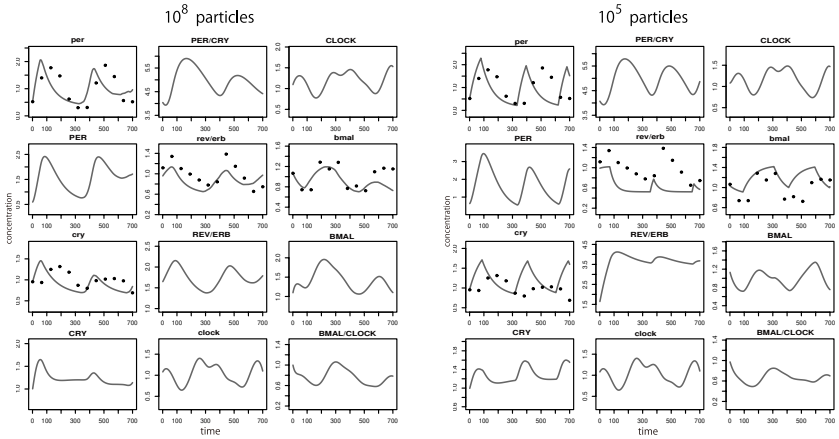

Figure 3. State transition estimated by using 10^8 (left) and 10^5 (right) particles.

(a) 10^8 particles are divided into 20 sets of 5 million particles, and then, repeat the particle filtering 20 times on Opteron 2200 (about 5 gigaflops).
(b) Particle filtering with 10^8(100 million) samples is performed on one petaflops machine.
(c) Particle filtering with 10^9(one billion) samples is performed on one petaflops machine.

For the evaluations of (b) and (c), the CPU times were estimated based on the observed CPU time of the experiment (a). The results of the tests are summarized in Table 1. The current implementation that divides 10^8 particles into 20 computational subunits requires about eight days on average, which can be reduced to 90 seconds by the aid of one petaflops computer. Even with one billion particles, the estimated CPU time can be within 20 minutes.

Table 1. Observed and estimated CPU times (second). The left column shows the measured CPU time on Opteron 2200 and 20 sets of 5 million particles. The middle and right columns show the estimated CPU times for 100 million and one billion particles.

	Opteron, 5mil. $\times$ 20	1 petaflops, 100 mil.	1 petaflops, 1 billion
Ave.	6.8×10^5	90	1.0×10^3
Min.	3.7×10^5	50	5.6×10^2
Max.	2.0×10^5	2.7×10^2	3.0×10^3

Pacific Symposium on Biocomputing 14:227-238 (2009)

4. Conclusion

This paper has shown the promising approach based on large scale particle filtering towards exhaustive explorations for kinetic parameters of *in silico* biological pathways. The approach could offer an overall view of posterior distributions defined over the 45 unknown parameters within a reasonable computing time. To our knowledge, the use of one hundred million particles is the first attempt not only in systems biology but also in the other scientific fields. Because the idea is too obvious, such a direct and straightforward approach may have been overlooked so far.

A limitation of our approach is the sensitivity of the estimation results to the choice of the prior distributions, that is, if the prior distributions are fairly inappropriate, we would not obtain reasonable posterior estimations. In order to overcome such a difficulty, we have to develop a methodology for incorporating the existing knowledge on biochemistry and systems biology.

According to the evaluation of computational time, it has been found that we can execute the particle filtering even with one billion particles within 20 minutes by the aid of one petaflops computer that will be accessible in Japan. The results of the experiments indicate that the large scale particle filtering can be a promising approach to parameter estimation of *in silico* biological pathways in the near future.

Appendix A. Resampling Algorithm

The filtering step ($\star$) of the algorithm that has $O(N \log N)$ time complexity is achieved as follows:

- Compute likelihood function of each particle $p(\boldsymbol{y}_t|\boldsymbol{x}^{(i)}_{t|t-1})$, $i = 1, \cdots, N$. Time complexity of this step is given by $O(N)$.
- Calculate the normalizing weights $q^{(i)}_t = p(\boldsymbol{y}_t|\boldsymbol{x}^{(i)}_{t|t-1})$ for $i = 1, \cdots, N$ such that $\sum_{i=1}^{N} q^{(i)}_t = 1$. Time complexity of this step is given by $O(N)$.
- Compute cumulative weight table $\hat{q}^{(i)}_t$ for $i = 0, \cdots, N$ by $\sum_{j=1}^{i} \hat{q}^{(j)}_t$ ($\hat{q}^{(0)}_t = 0$). Time complexity is given by $O(N)$.
- For each $j = 1, \cdots, N$
  - Generate random number u from uniform distribution $U(0, 1]$. Time complexity is $O(N)$.
  - Search the number k where $\hat{q}^{(k-1)}_t < u \leq \hat{q}^{(k)}_t$ by binary search. Time complexity is $O(N \log N)$.
  - Set $(\boldsymbol{x}^{(j)}_{t|t}, \boldsymbol{\theta}^{(j)}_t) \leftarrow (\boldsymbol{x}^{(k)}_{t|t-1}, \boldsymbol{\theta}^{(k)}_{t-1})$, which requires $O(N)$.

Only the binary searching requires $O(N \log N)$. In order to deal with much larger number of particles, we can replace the binary search algorithm by the resampling with the proportional-allocation that yields the filtered samples by increasing and decreasing the number of prediction samples proportional to their weights.

References

1. K. C. Chen, L. Calzone, A. Csikasz-Nagy, F. R. Cross, B. Novak and J. J. Tyson, *Mol Biol Cell.*, **15**, 3841-3862 (2004).
2. H. Matsuno, R. Murakami, R. Yamane, N. Yamasaki, S. Fujita, H. Yoshimori and S. Miyano, *Pac Symp Biocomput.*, **8**, 152-163 (2003).
3. A. Doi, S. Fujita, H. Matsuno, M. Nagasaki and S. Miyano, *In Silico Biol.*, **4**, 271-291 (2004).
4. A. Doi, M. Nagasaki, H. Matsuno and S. Miyano, *In Silico Biol.*, **6**, 1-13 (2006).
5. K. Fujarewicz, M. Kimmel, T. Lipniacki and A. Swierniak, *IEEE/ACM Transaction on Computational Biology and Bioinformatics*, **4**, 322-335 (2007).
6. M. Quach, N. Brunel and F. d'Alché-Buc, *Bioinformatics*, **23**, 3209-3216 (2007).
7. I. Nachman, A. Regev and N. Friedman, *Bioinformatics*, **20**, i248-i256 (2004).
8. G. Koh, H. F. C. Teong, M-V ment, D. Hsu and P.S. Thiagarajan, *Bioinformatics*, **22**, e271-e280 (2006).
9. B.-E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, F. d'Alché-Buc, *Bioinformatics*, **19**, ii386-ii349 (2003).
10. G. Kitagawa, *J. Computational and Graphical Stat.* **5**, 1-25 (1996).
11. N. J. Gordon, D. J. Salmond and A. F. M. Smith *IEE Proceedings-F,* **140**, 107-113 (1993).
12. T. Higuchi and G. Kitagawa, *IEICE Tras. Inf. Syst.*, **E83-D**, 36-43 (2000).
13. A. Doucet and N. De Freitas, Springer-Verlag (2001).
14. M. Nagasaki, R. Yamaguchi, R. Yoshida, S. Imoto, A.Doi, Y.Tamada, H. Matsuno, S. Miyano and T. Higuchi, *Genome Inform.* **17(1)**, 46-61 (2006).
15. S. Tasaki, M. Nagasaki, M. Oyama, H. Hata, K. Ueno, R. Yoshida, T. Higuchi, S. Sugano, S. Miyano, *Genome Inform.* **17(2)**, 226-238 (2006).
16. G. Kitagawa, *J. Amer. Stat. Assoc.*, **93**, 1203-1215 (1998).
17. Peter Pacheco, *Parallel Programming with MPI*, Morgan Kaufmann (1996).
18. Y. Fujii, Y. Okitsu, H. Matsuno, S. Miyano and S. T. Inoue, A new regulatory interactions suggested by simulations for circadian genetic control mechanism in mammals, *Genome Inform.*, available at `http://www.jsbi.org/journal/GIW0`, (2004).
19. H. R. Ueda, W. Chen, A. Adachi, H. Wakamatsu, S. Hayashi, T. Takasugi, M. Nagano, K. Nakahama, Y. Suzuki, S. Sugano, M. Iino, Y. Shigeyoshi and S. Hashimoto, *Nature*, **418**, 534-539 (2002).