Wu et al. BMC Bioinformatics 2014, 15(Suppl 12):S3
http://www.biomedcentral.com/1471-2105/15/S12/S3

BMC Bioinformatics

RESEARCH Open Access

Approximate Bayesian computation schemes for parameter inference of discrete stochastic models using simulated likelihood density

Qianqian Wu, Kate Smith-Miles, Tianhai Tian*

From IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2013)
Shanghai, China. 18-21 December 2013

Abstract

Background: Mathematical modeling is an important tool in systems biology to study the dynamic property of complex biological systems. However, one of the major challenges in systems biology is how to infer unknown parameters in mathematical models based on the experimental data sets, in particular, when the data are sparse and the regulatory network is stochastic.

Results: To address this issue, this work proposed a new algorithm to estimate parameters in stochastic models using simulated likelihood density in the framework of approximate Bayesian computation. Two stochastic models were used to demonstrate the efficiency and effectiveness of the proposed method. In addition, we designed another algorithm based on a novel objective function to measure the accuracy of stochastic simulations.

Conclusions: Simulation results suggest that the usage of simulated likelihood density improves the accuracy of estimates substantially. When the error is measured at each observation time point individually, the estimated parameters have better accuracy than those obtained by a published method in which the error is measured using simulations over the entire observation time period.

Background

In recent years, quantitative methods have become increasingly important for studying complex biological systems. To build a mathematical model of a complex system, two main procedures are commonly conducted [1]. The first step is to determine the elements of the network and regulatory relationships between the elements. In the second step, we need to infer the model parameters according to experimental data. Since biological experiments are time-consuming and expensive, normally experimental data are often scarce and incomplete compared with the number of unknown model parameters. In addition, the likelihood surfaces of large models are complex. The calibration of these unknown parameters within a model structure is one of the key issues in systems biology [2]. The analysis of such dynamical systems therefore requires new, effective and sophisticated inference methods.

During the last decade, several approaches have been developed for estimating unknown parameters: namely, optimization methods and Bayesian inference methods. Aiming at minimizing an objective function, optimization methods start with an initial guess, and then search in a directed manner within the parameter space [3,4]. The objective function is usually defined by the discrepancy between the simulated outputs of the model and sets of experimental data. Recently, the objective function has been extended to a continuous approach by considering simulation over the whole time period [5] and a multi-scale approach by including multiple types of experimental information [6]. Several types of optimization methods can be found in the literature, among which two major types are called gradient-based optimization methods and evolutionary-based optimization methods. Based on these two basic approaches, various techniques such as simulated annealing

* Correspondence: tianhai.tian@monash.edu
School of Mathematical Sciences, Monash University, Melbourne, Australia

[7]. linear and non-linear least-squares fitting [8], genetic algorithms [9] and evolutionary computation [10,11] have been attempted to build computational biology models. Using optimization methods, the inferred set of parameters produces the best fit between simulations and experimental data [12,13]. which have been successfully applied for biological systems, however, there are still some limitations with these methods such as the problem of high computational cost when significant noise exists in the system. To address these issues, deterministic and stochastic global optimization methods have been explored [14].

When modeling biological systems where molecular species are present in low copy numbers, measurement noise and intrinsic noise play a substantial role [15], which is a major obstacle for modeling. Bayesian inference methods have been used to tackle such difficulties by extracting useful information from noise data [16]. The main advantage of Bayesian inference is that it is able to infer the whole probability distributions of parameters by updating probability estimates using Bayes' Rule, rather than just a point estimate from optimization methods. Also. Bayesian methods are more robust than using other methods when they are applied to estimate stochastic systems, which is not that obvious for modeling of deterministic systems [17]. Developments have taken place during the last 20 years and recent advances in Bayesian computation including Markov chain Monte Carlo (MCMC) techniques and sequential Monte Carlo (SMC) methods have been successfully applied to biological systems [18,19].

For the case of parameter estimation when likelihoods are analytically or computationally intractable, approximate Bayesian computation (ABC) methods have been applied successfully [20,21]. ABC algorithms provide stable parameter estimates and are also relatively computationally efficient, therefore, they have been treated as substantial techniques for solving inference problems of various types of models that were intractable only a few years ago [19]. In ABC. the evaluation of the likelihood is replaced by a simulation-based procedure using the comparison between the observed data and simulated data [22]. Recently, a semi-automatic method has been proposed to construct the summary statistics for ABC [23]. These methods have been applied in a diverse range of fields such as molecular genetics, epidemiology and evolutionary biology etc. [24-26].

Despite substantial progress in the application of ABC to deterministic models, the development of inference methods for stochastic models is still at the very early stage. Compared with deterministic models, there are a number of open problems in the inference of stochastic models. For example, recent work proposed ABC to infer unknown parameters in stochastic differential equation models [27]. Our recent computational tests [28] showed the advantages and disadvantages of a published ABC algorithm for stochastic chemical reaction systems in [17]. In this work, we propose two novel algorithms to improve the performance of ABC algorithms using the simulated likelihood density.

Results and discussion

The first test system with four reactions

We first examine the accuracy of our proposed methods using a simple model of four chemical reactions [29]. The first reaction is the decay of molecule S_1. Then two molecules S_1 form a dimer S_2 in the second reaction; and this dimerization process is reversible, which is represented by the third reaction. The last reaction in the system is a conversion reaction from molecule S_2 to its product S_3. All these four reactions are given by

$$\begin{aligned} S_1 &\xrightarrow{c_1} (), \\ S_1 + S_1 &\xrightarrow{c_2} S_2, \\ S_2 &\xrightarrow{c_3} S_1 + S_1, \\ S_2 &\xrightarrow{c_4} S_3. \end{aligned}$$

We start with an initial condition with $\mathbf{S}$ = (10000, 0,0) and rate constants of $\mathbf{c}$ = (0.1,0.002,0.5,0.04), which is termed as the exact rate constants in this test. The stochastic simulation algorithm (SSA) was used to simulate the stochastic system [30]. A single trajectory for this model during a period of T = 30 in a step size of Δt = 3 is presented in Figure 1.

When applying the algorithms in the Method section to estimate model parameters, we assumed the prior distribution for each estimated parameter follows a uniform distribution $\pi(\theta) \sim U(0, A)$. For rate constants $c_1 \sim c_4$, the values of A are (0.5, 0.005, 1, 0.1). Figure 2 shows probabilistic distributions of the estimated rate constant of c_1 over iterations (2 ~ 5). In this test, we have the step size

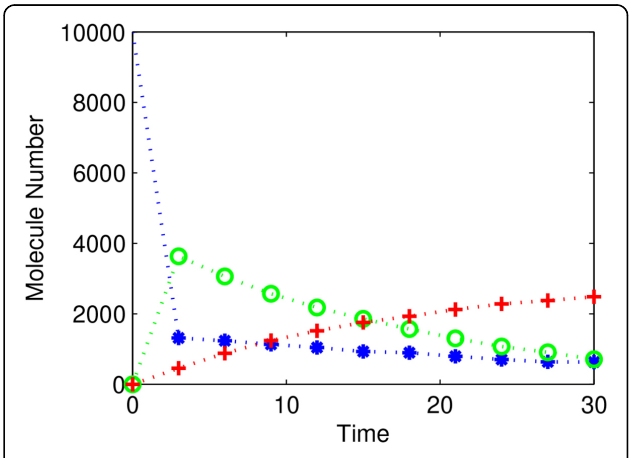

Figure 1 Simulated experimental data for system dynamics in a time length of 30 with step size Δt of 3. Blue star for S_1, green circle for S_2, and red cross for S_3.

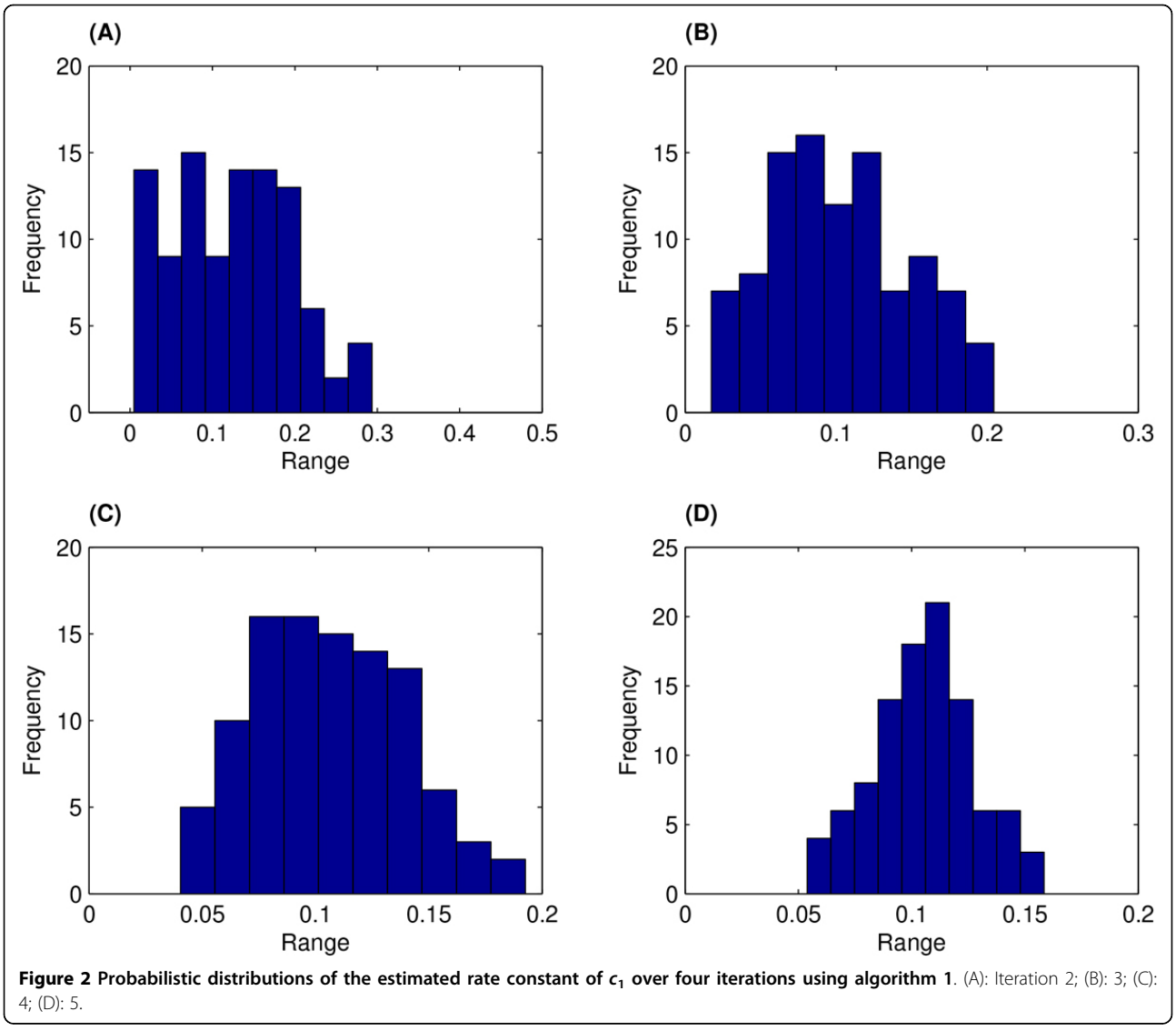

Figure 2 Probabilistic distributions of the estimated rate constant of c_1 over four iterations using algorithm 1. (A): Iteration 2; (B): 3; (C): 4; (D): 5.

Δt = 3 and simulation number B_k = 10. Figure 2 suggests that the probabilistic distribution starts from nearly a uniform distribution in the second iteration (Figure 2A) and gradually converges to a normalized-like distribution with a mean value that is close to the exact rate constant.

There are two tolerance values in the proposed algorithms, namely α for the discrepancy in step 2.c and $\in_k$ for the fitness error in step 2.d. In the following tests, we considered two strategies: the value of a is a constant [31] or its value varies over iterations. To examine the factors that influence the convergence rate of particles over iterations, we calculated the mean count number for each iteration, which is the averaged number of counts for accepting all simulated estimation of parameter sets. The averaged error is defined by the sum of relative errors of each rate constant for each iteration. Table 1 displays the performances of the tests under three schemes which used fixed discrepancy tolerance α = 0.1, 0.05 or varying values of α. In each case, we used the same values of $\in_k$ for the fitness tolerance. The value of α in the varying α strategy equals the value of $\in_k$, namely $\alpha_k = \in_k$.

In these performances, we used $\in_k$ = (0.07, 0.06, 0.055, 0.05, 0.045) and (0.05, 0.045, 0.04, 0.035, 0.03) for algorithm 1 with step sizes Δt = 3 and 5, respectively. For algorithm 2, these values are $\in_k$ = (0.095, 0.08, 065, 0.05, 0.04) and (0.059, 0.055, 0.05, 0.045, 0.04). An interesting observation is that the values of mean count number are very large in the first iteration, then decrease sharply and stay within a value stably. We have a detailed test of using different values of the fitness tolerance $\in_k$ and found that when using step size of Δt = 3, mean count number stays at one if $\in_k \geq 0.1$; but it starts to increase sharply to a large number if $\in_k < 0.1$.

Table 1 Comparison of averaged error and mean count number for estimated rate constants over five iterations using algorithms 1 and 2 with simulation number of 10 for system 1

Δt	$\alpha \backslash k$		1	2	3	4	5
Algorithm 1							
3	0.1	MN	15.41	7.21	7.36	8.21	10.05
		AE	0.7668	0.7294	0.7073	0.7832	0.6173
	0.05	MN	175.72	30.66	24.47	28.22	26.5
		AE	0.6120	0.5036	0.5521	0.7175	0.6132
	vary	MN	46.46	25.07	22.76	30.09	88.56
		AE	0.7669	0.5306	0.6780	0.5858	0.5945
5	0.1	MN	26.96	10.47	9.07	11.18	13.19
		AE	0.7107	0.5607	0.5366	0.4693	0.4853
	0.05	MN	130.64	27.38	25.42	35.36	35.79
		AE	0.5826	0.6495	0.4260	0.7548	0.4139
	vary	MN	141.97	30.28	53.47	127.16	2911.58
		AE	0.5587	0.4793	0.5416	0.5960	0.5375
Algorithm 2							
3	0.05	MN	467.61	52.34	41.08	69.17	195.69
		AE	0.5834	0.6091	0.4867	0.4995	0.4402
	vary	MN	100.26	32.04	24.78	80.15	1793.64
		AE	0.7132	0.6657	0.6305	0.6705	0.4833
5	0.05	MN	333.17	24.26	32.85	21.11	21.84
		AE	0.5962	0.5340	0.5761	0.4983	0.5518
	vary	MN	243.78	22.6	31.29	34.6	70.25
		AE	0.6565	0.6035	0.5759	0.5488	0.4263

Tests are experimented under different strategies of discrepancy tolerance such as α = 0.1, 0.05 or varies over iterations (AE:Averaged Error; MN: Mean count Number).

The observation numbers using a step size of $\Delta t = 3$ is 10 and the maximum error that can incur calculated from step 2.d is 0.1 with one hundred particles. Similarly, this critical $\in_k$ value is 0.06 for a step size of $\Delta t = 5$.

Meanwhile all averaged errors have a decreasing trend over iterations. Looking at different cases with various values of discrepancy tolerance α , it is also observed that using $\alpha = 0.1$ results in more discrepancies of the estimated parameters on average than the other two cases, in particular, than the case $\alpha = 0.05$. Thus in our following tests, we just concentrate on the cases of $\alpha = 0.05$ and varying α. In addition, we observe that by taking $\alpha = 0.05$ for the case with step size of $\Delta t = 3$, it leads to more accurate approximation since $\alpha = 0.05$ is less than most values of α in the case of varying values of α. It is consistent with the cases of a step size of $\Delta t = 5$ in which little differences can be found comparing strategies using $\alpha = 0.05$ and $\alpha = \in_k$ since the values of $\in_k$ are quite close to 0.05. In the case of varying values of α, a small value of ε_5 leads to a small value of α_5, which results in a substantial increase in mean count number. However, this large mean count number does not necessary bring more accurate estimated parameters. With these findings, we simulated results using $\alpha = 0.05$ and $\alpha = \in$ only for algorithm 2. Consistent results are obtained using algorithm 2. Moreover, results obtained using algorithm 2 is more accurate than those from algorithm 1.

The second test system with eight reactions

Although numerical results of the first test system are promising regarding the accuracy, that system has only four reactions. Thus the second test system, namely a prokaryotic auto-regulatory gene network, includes more reactions. This network involves both transcriptional and translational processes of a particular gene. In addition, dimers of the protein suppress its own gene transcription by binding to a regulatory region upstream of the gene [32-34]. This gene regulatory network consists of eight chemical reactions which are given below:

$$
\begin{aligned}
&R_1 : DNA + P_2 \xrightarrow{c_1} DNA \cdot P_2,\\
&R_2 : DNA \cdot P_2 \xrightarrow{c_2} DNA + P_2,\\
&R_3 : DNA \xrightarrow{c_3} DNA + mRNA,\\
&R_4 : mRNA \xrightarrow{c_4} \emptyset,\\
&R_5 : 2P \xrightarrow{c_5} P_2,\\
&R_6 : P_2 \xrightarrow{c_6} 2P,\\
&R_7 : mRNA \xrightarrow{c_7} mRNA + P_2,\\
&R_8 : P \xrightarrow{c_8} \emptyset.
\end{aligned}
$$

This gene network includes five species, namely DNA, messenger RNA, protein product, dimeric protein, and the compound formed by dimeric protein binding to the DNA promoter site, which are denoted by DNA, mRNA, P, P_2 and DNA · P_2, respectively. In this network, the first two reactions R_1 and R_2 are reversible reactions for dimeric protein binding to the DNA promoter site. Reactions R_3 and R_7 are transcriptional and translation processes for producing mRNA and protein, respectively. Reactions R_5 and R_6 represent the interchange between protein P and dimeric protein P_2. The system ends up with a degradation process of protein P [32].

To apply our algorithms, we start up with an initial condition of molecular copy number

$$(\text{DNA, mRNA, P, } P_2, \text{ DNA} \cdot P_2) = (10, 100, 100, 800, 100).$$

In addition, the following reaction rate constants

$$(c_1, ..., c_8) = (0.1, 0.7, 0.35, 0.01, 0.1, 0.9, 0.2, 0.01)$$

are used as the exact rate constants to generate a simulation for each molecular species during a period of $T = 50$ in a step size of $\Delta t = 1$ and results are presented by Figure 3. This simulated dataset is used as observation data for inferring the rate constants.

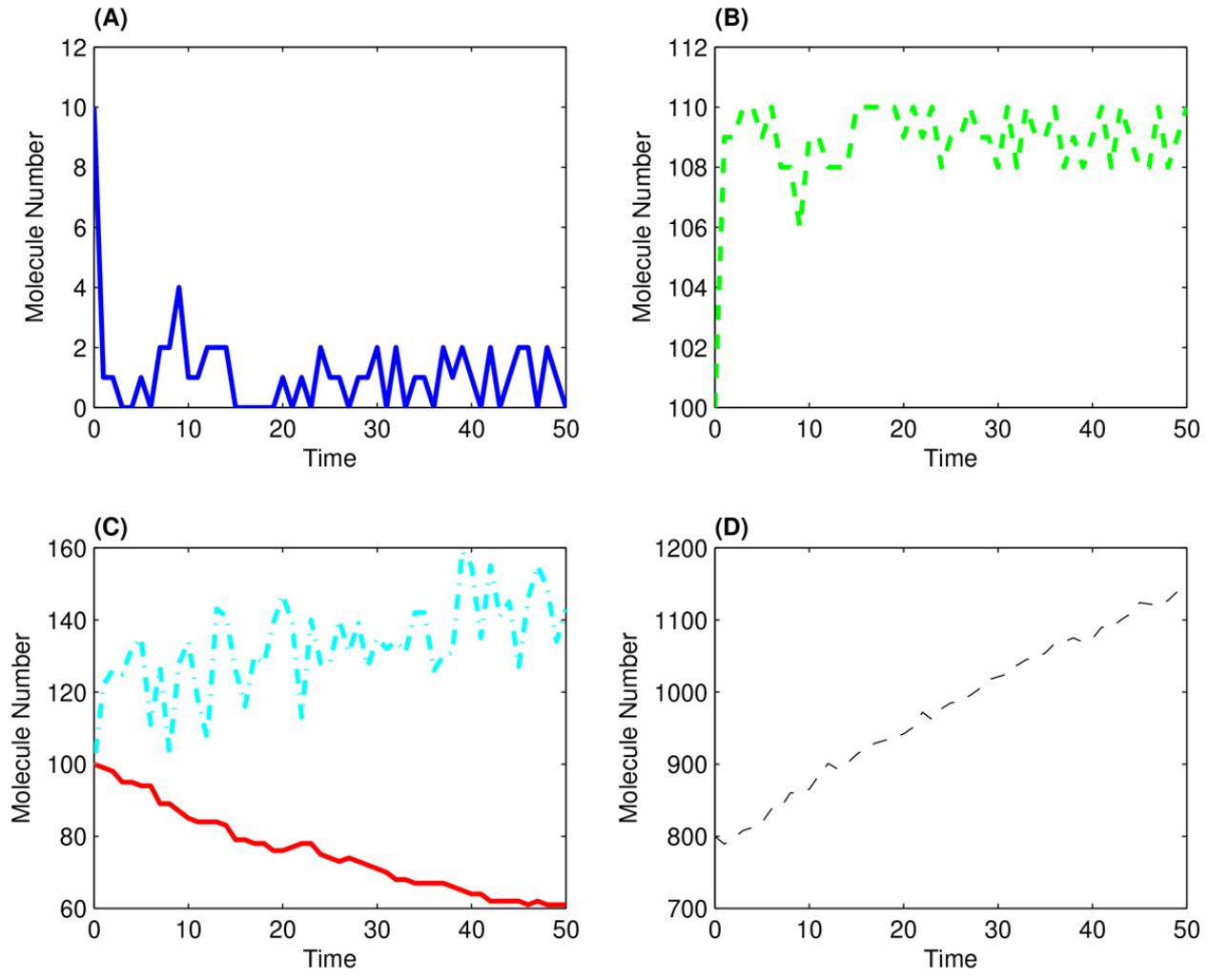

Figure 3 Simulated molecular numbers for system 2 in a time length of 50 with step size Δ*t* of 1. (a): DNA numbers; (B): numbers of DNA · P_2; (C): Red line for the numbers of mRNA black and cyan dash-dotted line for the numbers of P; (D): numbers of P_2.

The prior distribution of each parameter follows a uniform distribution $\pi(\theta) \sim U(0, B)$. For rate constants $c_1 \sim c_8$, the values of B are (0.5,2,1,0.1,0.5, 5,1,0.1). The proposed two algorithms were implemented over five iterations and each iteration contains 100 particles. We choose step sizes Δt = 2 or 5 and the number of stochastic simulation B_k = 10.

Figure 4 gives the probabilistic distribution of the estimated rate constant c_7 over 2nd ~ 5th iterations. The distribution of the first iteration is close to the uniform distribution, and this is not presented. Since the second iteration, the estimated rate constant begins to accumulate around the exact value c_7 = 0.2. At the last iteration, the probability in Figure 4D shows a normalized-like distribution. Compared with the results of system 1 in Figure 2, the convergence rate of the parameter distribution of system 2 is slower. Our numerical results suggested that this convergence rate depends on the strategy of choosing the values of discrepancy tolerance α.

To analyze the factors that influence the convergence property of estimates, the mean count number as well as the averaged error for each iteration k are obtained. Results are presented in Table 2. Using algorithm 1 and 2, we tested for step sizes of Δt = 2 and Δt = 5. Since the errors of estimates obtained using a fixed value of α = 0.1 are always larger than those obtained by a = 0.05, we only tested with the cases of a fixed value α = 0.05 and varying values of α. For algorithm 1, we tested two cases for the varying values of discrepancy tolerance α. In the first test, the values are $\in_k$ = (0.21,0.2,0.19,0.18,0.175) and $\alpha = \in_k$ for varying values of α, which is the case "Same $\in_k$" in Table 2. The values of $\in_k$ are also applied to the case of a fixed value α = 0.05. In this case, the averaged count number of varying α is much smaller than that of

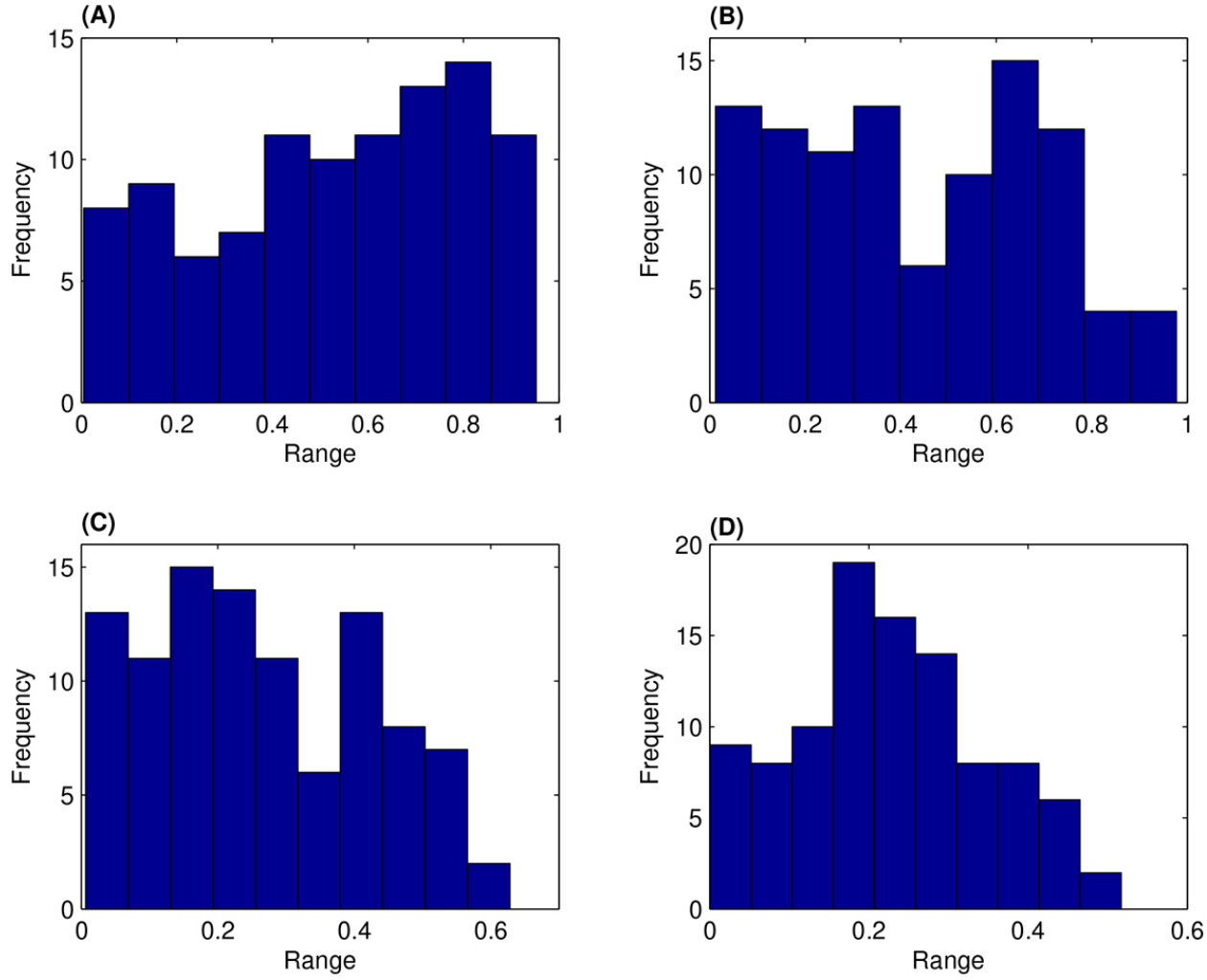

Figure 4 Probabilistic distributions of the estimated rate constant c_7 over four iterations using algorithm **1**. (A):Iteration 2; (B): 3: (C):4; (D):5.

a fixed value of α. Thus we further decreased the value of α to (0.15,0.125,0.1,0.075,0.07), which is the case "Diff. $\in_k$" in Table 2. In this case, the mean count numbers are similar to those using a fixed α. Numerical results suggested that the strategy of using a fixed value of α generates estimates with better accuracy than the strategies of using varying α values, even when the computing time of the varying α strategy is larger than that of the fixed α strategy.

For algorithm 2, we carried out similar tests. In the first case, we set $\in_k$ =(0.24,0.23,0.22,0.21,0.2), which is applied to the strategy of fixing α = 0.05 and varying α with α= $\in_k$ that is the case "Same $\in_k$" in Table 2. Again, the averaged count numbers of varying α strategy are much smaller than those using a fixed α. Thus we decreased the value to (0.095,0.09,0.085,0.08, 0.075), which is the case "Diff. $\in_k$" in Table 2; However, the averaged count numbers in the "Diff. $\in_k$" case are similar to those of the previous two strategies, namely a fixed a and "Same $\in_k$". For algorithm 2, Table 2 suggests that the varying α strategy generates estimates that are more accurate than those obtained from the fixed α strategy. However, the best estimates in Table 2 are obtained using algorithm 1 and fixed α strategy.

Conclusions

To uncover the information of biological systems, we proposed two algorithms for the inference of unknown parameters in complex stochastic models for chemical reaction systems. Algorithm 1 is in the framework of ABC SMC and uses transitional density based on the simulations over two consecutive observation time points. Algorithm 2 generates simulations of the whole time interval but differs from the published method in the error finding steps by comparing errors of simulated data to experimental data at each time point. The proposed new algorithms impose

Table 2 Comparison of averaged error and mean count number for estimated rate constants of system 2 using algorithms 1 and 2

Δt	$\alpha \backslash k$		1	2	3	4	5
Algorithm 1							
2	0.05	MN	18.29	7.53	9.8	12.7	14.23
		AE	4.6211	4.4179	4.7138	4.2188	3.8119
	Same ϵ_k	MN	2.69	2.07	2.16	1.93	1.93
		AE	4.7006	4.9603	4.8841	4.6833	4.7298
	Diff. ϵ_k	MN	15.26	7.85	8.78	13.06	12.28
		AE	4.8295	4.5322	5.0418	4.7346	4.6069
5	0.05	MN	9.69	3.48	3.12	58.2	74.07
		AE	4.1076	4.3243	4.1868	3.5311	3.5194
	Same ϵ_k	MN	2.34	2.31	2.42	16.9	11.38
		AE	4.9862	4.7669	4.6716	3.8873	4.0017
	Diff. ϵ_k	MN	25.72	8.14	10.45	25.8	174.88
		AE	4.0461	3.9583	3.7474	3.5655	3.6951
Algorithm 2							
2	0.05	MN	89.7	19.75	17.8	40.42	69.52
		AE	4.0540	4.1339	4.1376	3.9696	3.9009
	Same ϵ_k	MN	2.52	3.85	3.55	3.82	3.84
		AE	5.0456	4.6069	4.3666	4.5876	3.8958
	Diff. ϵ_k	MN	197.49	15.05	22.09	36.85	94.24
		AE	3.8712	3.7934	4.3158	3.6485	3.5989
5	0.05	MN	138.14	30.52	46.66	98.87	377.66
		AE	4.0258	3.7218	3.8258	3.8445	3.9205
	Same ϵ_k	MN	21.67	11.34	11.17	26.65	59.64
		AE	4.0545	3.5715	4.1910	3.7252	3.8667
	Diff. ϵ_k	MN	185.54	28.39	33.81	89.81	846.61
		AE	3.7810	3.6694	3.6939	3.9806	3.8515

Three strategies are used to choose the discrepancy tolerance α: a fixed value of α= 0.05; varying αvalues; and α= $\epsilon_k$denoted as same ϵ_k); varying αvalues that are smaller than ϵ_k (denoted as diff. ϵ_k).(AE:Averaged Error; MN: Mean count Number).

stricter criteria to measure the simulation error. Using two chemical reaction systems as the test problems, we examined the accuracy and efficiency of proposed new algorithms. Based on the results of two algorithms for system 1, we discovered that taking smaller values of discrepancy tolerance α will result in more accurate estimates of unknown model parameters. This conclusion is confirmed by the second system that we tested under different conditions. Numerical results suggested that the proposed new algorithms are promising methods to infer parameters in high-dimensional and complex biological system models and have better accuracy compared with the results of the published method [28]. The encouraging result is that new algorithms do not need more computing time to achieve such accuracy. Our computational tests showed that the selection for the value of fitness tolerance is a key step in the success of ABC algorithms. The advantage of the population Monte-Carlo methods is the ability to reduce the fitness tolerance gradually over populations. Generally, a smaller value of fitness tolerance will lead to a larger number of iteration count and consequently larger computing time. For deterministic inference problems, a smaller value of fitness tolerance normally will generate estimates with better accuracy. However, for stochastic models, this conclusion is not always true. In addition to the fitness tolerance, our numerical results suggested that other factors, such as the simulation algorithm for chemical reaction systems and the strategy of discrepancy tolerance, also have influences on the accuracy of estimates. Thus more skilled approaches, such as the adaptive selection process for the fitness tolerance, should be considered to improve the performance of ABC algorithms.

In this work, we used the SSA to simulate chemical reaction systems [30]. This approach may be appropriate when the biological system is not large. In fact, for the two biological systems discussed in this work, the computing time of inference is still very large. To reduce the computing time, more effective methods should be used to simulate the biological systems, such as the τ-leap methods [35] and multi-scale simulation methods [36,37]. Another alternative approach is to use parallel computing to reduce the heavy computing loads. All these issues are potential topics for future research work.

Methods

ABC SMC algorithm

ABC algorithms bypass the requirement for evaluating likelihood functions directly in order to obtain the posterior distributions of unknown parameters. Instead, ABC methods simulate the model with given parameters, compare the observed and simulated data, and then accept or reject the particular parameters based on the error of simulation data. Thus there are three key steps in the implementations of ABC algorithms. The first step is the generation of a sample of parameters $\boldsymbol{\theta}^*$ from the prior distribution of parameters or from other distributions that are determined in ABC algorithms. The second step is to define distance function d($\mathbf{X}$, $\mathbf{Y}$) between the simulated data $\mathbf{X}$ and experimental observation data $\mathbf{Y}$. Finally, a tolerance value is needed as a selection criterion to accept or reject the sampled parameter $\boldsymbol{\theta}^*$. Based on the generic form of ABC algorithm [17], a number of methods have been developed including ABC rejection sampler and ABC MCMC [38,39]. The ABC rejection algorithm is one of the basic ABC algorithm that may result in long computing time when a badly prior distribution that is far away from posterior distribution is chosen. ABC MCMC introduces a concept of acceptance probability during the decision making step which saves computing time. However, this may result in getting stuck in the regions of low probability for the chain and we may never be able to get a good approximation. To tackle these challenges, the idea of particle filtering has

been introduced. Instead of having one parameter vector at a time, we sample from a pool of parameter sets simultaneously and treat each parameter vector as a particle. The algorithm starts from sampling a pool of N particles for parameter vector $\boldsymbol{\theta}$ through prior distribution $\pi(\theta)$. The sampled particle candidates $(\boldsymbol{\theta}_1^*, \cdots, \boldsymbol{\theta}_N^*)$ will be chosen randomly from the pool and we will assign each particle a corresponding weight ω to be considered as the sampling probability. A perturbation and filtering process following through a transition kernel $q(\cdot|\theta^*)$ finds the particles $\boldsymbol{\theta}^{**}$. Similarly with $\boldsymbol{\theta}^{**}$, data $\mathbf{Y}$ can be simulated and compared with experimental data $\mathbf{X}$ to further fulfil the requirements for estimating posterior distribution.

The basic form of algorithm described above is as follows [19]:

Algorithm: ABC SMC

1. Define the threshold values $\in_1, \cdots, \in_K$, start with iteration $k = 1$.

2. Set the particle indicator $i = 1$.

3. If $k = 1$, sample θ^* from the proposed prior distribution $\pi(\theta)$. Generate a candidate data set $D(_b)(\theta^*)$ B_k times and calculate the value of $b_k(\theta^*)$, where $D(_b) \sim p(D|\theta)$ for any fixed parameter θ,

$$b_k\left(\theta^*\right) = \sum_{b=1}^{B_k} 1\left(d\left(D_0, D_{(b)}\left(\theta^*\right)\right) \leq \in_k\right) \quad (1)$$

and D_0 is the experimental data set.

If $k > 1$, sample θ from the previous population $\left\{\theta_{k-1}^i\right\}$ with weights w_{k-1} and perturb the particle to obtain θ^* using a kernel function $\mathbb{K}_k$.

If $\pi(\theta^*) = 0$ or $b_k(\theta^*) = 0$, return to the beginning of step 3.

4. Set $\theta_k^i = \theta^*$ and determine the weight for each estimated particles θ_k^i,

$$w_k^{(i)} = \begin{cases} b_k\left(\theta_k^i\right) & \text{if } k = 1; \\ \dfrac{\pi\left(\theta_k^i\right) b_k\left(\theta_k^i\right)}{\sum_{j=1}^N \mathbb{K}_k\left(\theta_{k-1}^j, \theta_k^i\right)} & \text{if } k > 1. \end{cases}$$

If $i < N$, update $i = i + 1$ and return to step 3.

5. Normalize the weights $w_k^{(i)}$ If $k < K$, update $k = k + 1$ and go back to step 2.

A number of algorithms have been developed using the particle filtering technique, such as the partial rejection control, population Monte-Carlo and SMC. Each of them differs in the formation of weight w and the transition kernels.

ABC using simulated likelihood density

ABC SMC method uses the simulation over the entire time period to measure the fitness to experimental data, which is consistent to the approaches used for deterministic models [17]. For stochastic models, the widely used approach is treating transitional density as the likelihood function [40,41]. Based on a sequence of n + 1 observations $\mathbf{X} = [X_0, X_1, \cdots, X_n]$ at time points $[t_0, t_1, \cdots, t_n]$, for a given parameter set θ the joint transitional density is defined as

$$f_0\left[(t_0, X_0)\,|\theta\right] \prod_{i=1}^{n} f\left[(t_i, X_i)\,(t_{i-1}, X_{i-1}), \cdots, (t_0, X_0); \theta\right], \quad (2)$$

where $f_0[\cdot]$ is the density of initial state, and

$$f\left[(t_i, X_i) \mid (t_{i-1}, X_{i-1}), \cdots, (t_0, X_0); \theta\right] \quad (3)$$

is the transitional density starting from (t_{i-1}, X_{i-1}) and evolving to (t_i, X_i). When the process X is Markov, the density (3) is simplified as

$$f\left[(t_i, X_i) \mid (t_{i-1}, X_{i-1}); \theta\right]. \quad (4)$$

In the simulated likelihood density (SLD) methods, this transitional density is approximated by that obtained from a large number of simulations.

Based on the discrete nature of biochemical reactions with low molecular numbers, it was proposed to use the frequency distribution of simulated molecular numbers to calculate the transitional density [31]. The frequency distribution is evaluated by

$$F[X = X_l] = \frac{1}{B_k} \sum_{m=1}^{B_k} [1 - \delta(X_l, X_{ml})]$$

using B_k simulations with the simulated state X_{ml}. Here the function $\delta(x)$ is defined by

$$\delta(X_l, X_{ml}) = \begin{cases} 0 & \text{if } d(X_l, X_{ml}) < \alpha X_l; \\ 1 & \text{else,} \end{cases}$$

where $d(x, y)$ is a distance measure between x and y.

Here we propose a new algorithm that uses the simulated transitional density function as the objective function. Unlike ABC SMC algorithm [17], the new method considers the transitional density function from t_{i-1} to t_i only at each step. Based on the framework of ABC SMC, the new algorithm using transitional density is proposed as follows.

ABC SLD algorithm 1

1. Given data $\mathbf{X}$ and any assumed prior distribution $\pi(\theta)$, define a set of threshold values $\in_1, \cdots, \in_K$.
2. For iteration $k = 1$,
   (a) Set the particle indicator $i = 1$, sample $\theta^* \sim \pi(\theta)$.
   (b) For time step $l = 1,2, \cdots, n$, use initial condition $\mathbf{X}_{l-1}$ and parameter θ^* to generate data $\mathbf{Y}$ at t_l for B_k times.

(c) For $m = 1, \cdots, B_k$, calculate the value of discrepancy and test for

$$d(\mathbf{X}_l, \mathbf{Y}_{m1}) \leq \alpha \mathrm{X}_l, \quad (5)$$

where α is a defined constant.
If it is true, let $\beta_{ml}(\theta^*) = 0$, otherwise it is one. Then determine

$$b_l(\theta^*) = \sum_{m=1}^{B_k} \beta_{ml}(\theta^*). \quad (6)$$

(d) Calculate

$$\in = \sum_{l=1}^{m} \frac{1}{B_k} (B_k - b_l(\theta^*)). \quad (7)$$

If $\in < \in_k$, update $\theta_i^k = \theta^*$ and move to the next particle $i = i + 1$.
(e) Assign weight $w_i^k = \frac{1}{N}$ for each particle.

3. Determine the variance for the particles in the first iteration

$$\sigma_1 = \sqrt{var\left(\theta_{\{1:N\}}^1\right)}$$

4. For iteration $k = 2, \cdots, K$
(a) Start with $i = 1$, Sample $\theta^* \sim \theta_{i:N}^{k-1}$ using the calculated weights $w_{i:N}^{k-1}$.
(b) Perturb θ^* through sampling $\theta^{**} \sim q(\theta|\theta^*>)$, where $q = N\left(\theta^*, \sigma_{k-1}^2\right)$ or $q = U(a, b)$. Here values of a,b depend on θ^* and σ_{k-1}^2.
(c) Generate simulations and calculate the error $\in$ using the same steps as in $2(b) \sim (d)$.
(d) For each particle, assign weights

$$w_i^k = \frac{\pi\left(\theta_i^k\right) b_k\left(\theta_k^i\right)}{\sum_{j=1}^{N} w_j^{k-1} q\left(\theta_j^{k-1}|\theta_j^k, \sigma_{k-1}\right)}.$$

(e) Determine the variance for the particles in the k-th iteration

$$\sigma_k = \sqrt{var\left(\theta_{\{1:N\}}^k\right)}.$$

An alternative approach is to generate simulations over the observation time period but compare the error to experimental data at each time point. The approach locates somewhere between ABC SMC algorithm [17] and the proposed Algorithm 1. which is presented below. For simplicity we do not give a detailed algorithm, but just provide the key steps 2.b) ~ 2.d) that are different from those in Algorithm 1.

ABC SLD algorithm 2

2.b) Generate data $\mathbf{Y}$ B_k times using θ^*.

2.c) For $m = 1, \cdots, B_k$ and $l = 1,2, \cdots, n$, calculate the value of discrepancy $d(\mathbf{X}_l, \mathbf{Y}_{ml})$ and test for

$$|\mathbf{X}_1 - \mathbf{Y}_{ml}| \leq \alpha \mathbf{X}_l.$$

If it is true, let $b_{ml}(\theta^*) = 0$, otherwise it is one.

2.d) Calculate

$$\in = \sum_{l=1}^{n} \frac{1}{B_k} \sum_{m=1}^{B_k} b_{ml}(\theta^*).$$

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

TT conceived and designed the study. QW and TT developed algorithms and carried out research. QW, KS and TT analyzed the data, interpreted the results and wrote the paper. All authors edited and approved the final version of the manuscript.

Acknowledgements

The authors would like to thank the Australian Research Council for the Discovery Project (T.T. DP120104460). T.T. is also an ARC Future Fellow (FT100100748).

Declarations

The publication costs for this article were funded by the corresponding author.
This article has been published as part of *BMC Bioinformatics* Volume 15 Supplement 12, 2014: Selected articles from the IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2013): Bioinformatics. The full contents of the supplement are available online at http://www.biomedcentral.com/bmcbioinformatics/supplements/15/S12.

Published: 6 November 2014

References

1. Zhan C, Yeung LF: **Parameter estimation in systems biology models using spline approximation.** *BMC systems biology* 2011, **5**:14.
2. Kikuchi S, Tominaga D, Arita M, Takahashi K, Tomita M: **Dynamic modeling of genetic networks using genetic algorithm and S-system.** *Bioinformatics* 2003, **19(5)**:643-650.
3. Gadkar KG, Gunawan R, Doyle FJ: **Iterative approach to model identification of biological networks.** *BMC bioinformatics* 2005, **6**:155.
4. Gonzalez OR, Küper C, Jung K, Naval PC, Mendoza E: **Parameter estimation using Simulated Annealing for S-system models of biochemical networks.** *Bioinformatics* 2007, **23(4)**:480-486.
5. Deng Z, Tian T: **A continuous approach for inferring parameters in mathematical models of regulatory networks.** *BMC bioinformatics* 2014, **15**:256.
6. Tian T, Smith-Miles K: **Mathematical modelling of GATA-switching for regulating the differentiation of hematopoietic stem cell.** *BMC bioinformatics* 2014, **8(S8)**:S8.
7. Kirkpatrick S, Gelatt CD, Vecchi MP: **Optimization by Simulated Annealing.** *Science* 1983, **220(4598)**:671-680.
8. Mendes P, Kell D: **Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation.** *Bioinformatics* 1998, **14(10)**:869-883.
9. Srinivas M, Patnaik LM: **Genetic algorithms: A survey.** *Computer* 1994, **27(6)**:17-26.
10. Ashyraliyev M, Jaeger J, Blom J: **Parameter estimation and determinability analysis applied to Drosophila gap gene circuits.** *BMC Systems Biology* 2008, **2**:83.
11. Moles CG, Mendes P, Banga JR: **Parameter estimation in biochemical pathways: a comparison of global optimization methods.** *Genome research* 2003, **13(11)**:2467-2474.

12. Lall R, Voit EO: **Parameter estimation in modulated, unbranched reaction chains within biochemical systems.** *Computational biology and chemistry* 2005, **29(5)**:309-318.
13. Lillacci G, Khammash M: **Parameter estimation and model selection in computational biology.** *PLoS computational biology* 2010, **6(3)**:el000696.
14. Goel G, Chou IC, Voit EO: **System estimation from metabolic time-series data.** *Bioinformatics* 2008, **24(21)**:2505-2511.
15. Raj A, van Oudenaarden A: **Nature, nurture, or chance: stochastic gene expression and its consequences.** *Cell* 2008, **135(2)**:216-226.
16. Wilkinson DJ: **Bayesian methods in bioinformatics and computational systems biology.** *Briefings in, bioinformatics* 2007, **8(2)**:109-116.
17. Toni T, Welch D, Strelkowa N, Ipsen A, Stumpf MP: **Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems.** *Journal of the Royal Society Interface* 2009, **6(31)**:187-202.
18. Battogtokh D, Asch DK, Case ME, Arnold J, Schüttler HB: **An ensemble method for identifying regulatory circuits with special reference to the qa gene cluster of Neurospora crassa.** *Proceedings of the National Academy of Sciences* 2002, **99(26)**:16904-16909 [http://www.pnas.org/content/99/26/16904.abstract].
19. Sisson SA, Fan Y, Tanaka MM: **Sequential monte carlo without likelihoods.** *Proceedings of the National Academy of Sciences* 2007, **104(6)**:1760-1765.
20. Beaumont MA, Zhang W, Balding DJ: **Approximate Bayesian Computation in Population Genetics.** *Genetics* 2002, **162(4)**:2025-2035.
21. Marjoram P, Molitor J, Plagnol V, Tavaré S: **Markov chain Monte Carlo without likelihoods.** *Proceedings of the National Academy of Sciences* 2003, **100(26)**:15324-15328.
22. Pritchard JK, Seielstad MT, Perez-Lezaun A, Feldman MW: **Population growth of human Y chromosomes: a study of Y chromosome microsatellites.** *Molecular Biology and Evolution* 1999, **16(12)**:1791-1798.
23. Fearnhead P, Prangle D: **Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation.** *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 2012, **74(3)**:419-474.
24. Marjoram P, Tavaré S: **Modern computational approaches for analysing molecular genetic variation data.** *Nature Reviews Genetics* 2006, **7(10)**:759-770.
25. Tanaka MM, Francis AR, Luciani F, Sisson S: **Using approximate Bayesian computation to estimate tuberculosis transmission parameters from genotype data.** *Genetics* 2006, **173(3)**:1511-1520.
26. Thornton K, Andolfatto P: **Approximate Bayesian inference reveals evidence for a recent, severe bottleneck in a Netherlands population of Drosophila melanogaster.** *Genetics* 2006, **172(3)**:1607-1619.
27. Picchini UL: **Inference for SDE models via Approximate Bayesian Computation.** *Journal of Computational and Graphical Statistics in press* 2014.
28. Wu Q, Smith-Miles K, Tian T: **Approximate Bayesian computation for estimating rate constants in biochemical reaction systems.** *Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on 2013* 416-421.
29. Daigle BJ, Roh MK, Petzold LR, Niemi J: **Accelerated maximum likelihood parameter estimation for stochastic biochemical systems.** *BMC bioinformatics* 2012, **13**:68.
30. Gillespie DT: **Exact stochastic simulation of coupled chemical reactions.** *The journal of physical chemistry* 1977, **81(25)**:2340-2361.
31. Tian T, Xu S, Gao J, Burrage K: **Simulated maximum likelihood method for estimating kinetic rates in gene expression.** *Bioinformatics* 2007, **23**:84-91.
32. Wang Y, Christley S, Mjolsness E, Xie X: **Parameter inference for discretely observed stochastic kinetic models using stochastic gradient descent.** *BMC systems biology* 2010, **4**:99.
33. Golightly A, Wilkinson DJ: **Bayesian inference for stochastic kinetic models using a diffusion approximation.** *Biometrics* 2005, **61(3)**:781-788.
34. Reinker S, Altman R, Timmer J: **Parameter estimation in stochastic biochemical reactions.** *IEE Proceedings-Systems Biology* 2006, **153(4)**:168-178.
35. Tian T, Burrage K: **Binomial leap methods for simulating stochastic chemical kinetics.** *The Journal of chemical physics* 2004, **121(21)**:10356-10364.
36. Pahle J: **Biochemical simulations: stochastic, approximate stochastic and hybrid approaches.** *Briefings in bioinformatics* 2009, **10**:53-64.
37. Burrage K, Tian T, Burrage P: **A multi-scaled approach for simulating chemical reaction systems.** *Progress in biophysics and molecular biology* 2004, **85(2)**:217-234.
38. Boys RJ, Wilkinson DJ, Kirkwood TB: **Bayesian inference for a discretely observed stochastic kinetic model.** *Statistics and Computing* 2008, **18(2)**:125-135.
39. Golightly A, Wilkinson DJ: **Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo.** *Interface Focus* 2011, **1(6)**:807-820.
40. Hurn AS, Jeisman J, Lindsay KA: **Seeing the wood for the trees: A critical evaluation of methods to estimate the parameters of stochastic differential equations.** *Journal of Financial Econometrics* 2007, **5(3)**:390-455.
41. Hurn A, Lindsay K: **Estimating the parameters of stochastic differential equations.** *Mathematics and computers in simulation* 1999, **48(4)**:373-384.

doi:10.1186/1471-2105-15-S12-S3
Cite this article as: Wu *et al.*: Approximate Bayesian computation schemes for parameter inference of discrete stochastic models using simulated likelihood density. *BMC Bioinformatics* 2014 **15**(Suppl 12):S3.