Evaluation of Parallel Tempering to Accelerate Bayesian Parameter Estimation in Systems Biology

Sanjana Gupta, **Liam Hainsworth**, **Justin S. Hogg**, **Robin E. C. Lee**, and **James R. Faeder**

Department of Computational and Systems Biology, University of Pittsburgh School of Medicine, Pittsburgh, PA 15260, USA

Abstract

Models of biological systems often have many unknown parameters that must be determined in order for model behavior to match experimental observations. Commonly-used methods for parameter estimation that return point estimates of the best-fit parameters are insufficient when models are high dimensional and under-constrained. As a result, Bayesian methods, which treat model parameters as random variables and attempt to estimate their probability distributions given data, have become popular in systems biology. Bayesian parameter estimation often relies on Markov Chain Monte Carlo (MCMC) methods to sample model parameter distributions, but the slow convergence of MCMC sampling can be a major bottleneck. One approach to improving performance is parallel tempering (PT), a physics-based method that uses swapping between multiple Markov chains run in parallel at different temperatures to accelerate sampling. The temperature of a Markov chain determines the probability of accepting an unfavorable move, so swapping with higher temperatures chains enables the sampling chain to escape from local minima. In this work we compared the MCMC performance of PT and the commonly-used Metropolis-Hastings (MH) algorithm on six biological models of varying complexity. We found that for simpler models PT accelerated convergence and sampling, and that for more complex models, PT often converged in cases MH became trapped in non-optimal local minima. We also developed a freely-available MATLAB package for Bayesian parameter estimation called PTEMPEST (http://github.com/RuleWorld/ptempest), which is closely integrated with the popular BioNetGen software for rule-based modeling of biological systems.

Index Terms

Bayesian parameter estimation; Systems biology; Parallel tempering; Rule-based modeling

I. INTRODUCTION

Mathematical and computational models have been gaining widespread use as tools to summarize our understanding of biological systems and to make novel predictions that can be tested experimentally [12], [21]. Doing this requires a model to be correctly parameterized. Parameter estimation, the process of inferring model parameters from experimental data, typically involves defining a cost function that quantifies the discrepancy between the model output and the data, and then performing a search for parameterizations that minimize the cost [2], [22].

There are many commonly-used methods for finding parameter sets that minimize the model cost. These can broadly be divided into gradient-based and gradient-free methods. Gradient-based methods are local optimization methods that iteratively use the gradient of the cost function to compute a search direction and step length, followed by updating the parameters and checking for convergence [2]. Popular gradient-based methods in systems biology include gradient descent, Newton's method, the Gauss-Newton algorithm, and the Levenberg-Marquardt algorithm [1]. However, these methods can fail to find the global minimum when landscapes are discontinuous or multi-modal, as is frequently the case for large biological models, which can have many more parameters than independent data points to constrain the model [25].

Gradient-free methods have the advantage that the landscape need not be smooth, but local search methods, such as the Nelder-Mead simplex, become inefficient for high-dimensional problems [11]. Gradient-free global optimization methods such as genetic algorithms and particle swarm optimization can be effective at finding optimal solutions in high-dimensional spaces [19]. However, the combination of high-dimensional parameter spaces and the limited amount of data available from typical biological experiments often means that multiple parameter combinations equivalently describe the experimental data, which is referred to as the parameter identifiability problem [13]. When parameters are non-identifiable, a single parameter set is insufficient to describe the feasible space of parameters associated with a model.

Bayesian methods solve this problem naturally by attempting to estimate the probability distribution of the model parameters given the experimental data [22], which allows simultaneous determination of best-fit parameters and parameter sensitivities, while also providing a framework to introduce prior information that the modeler may have about the parameters. Bayesian methods include likelihood-based approaches, such as Markov Chain Monte Carlo (MCMC) methods [9], and likelihood-free approaches, such as Approximate Bayesian Computation (ABC) [22].

MCMC is commonly used in systems biology, but slow convergence is often a major bottleneck for standard sampling algorithms, such as Metropolis-Hastings (MH) [9]. The development of modular and rule-based software for model construction and simulation [16], [23], [36], allows for the construction of increasingly complex models (e.g., [7]), which combined with the increasing availability of single-cell data [35] motivates the need for accelerated methods for Bayesian parameter estimation. Parallel tempering (PT) is a physics-based MCMC method that efficiently samples a probability distribution and can accelerate convergence over conventional MCMC methods [8]. This method has been widely used for molecular dynamics simulations to sample the conformational space of biomolecules [15], [29], but is less common in systems biology [10], [24], [25]. Here, we describe key algorithmic elements of the method, provide a software implementation, and evaluate its performance on a series of biological models of increasing complexity.

The remainder of this paper is organized as follows: In Sec. II, we describe the MH and PT algorithms as well as the ABC and ABC-SMC methods used by the software ABC-SysBio

[22], which we will later use for comparison. We also include a brief description of the PTEMPEST software for Bayesian parameter estimation. In Sec. III we present a series of examples of increasing complexity to test the performance of PT relative to MH with regards to quality of fit, convergence speed, and sampling efficiency. We include a comparison with ABC-SysBio and further show an application of using Bayesian methods with Laplace priors to achieve model reduction. Finally, in Sec. IV we discuss our main findings, limitations, and areas for future work.

II. METHODS

Bayesian parameter estimation methods infer the posterior distribution that describes the uncertainty in the parameter values that remains even after the data is known [22]. The probability of observing the parameter set θ given the data Y is given by Bayes' rule

$$p(\theta|Y) \propto p(Y|\theta)p(\theta),$$

where $p(Y|\theta)$ is the conditional probability of Y given θ, which is described by a *likelihood model*, and $p(\theta)$ is the independent probability of θ, often referred to as the *prior distribution* on model parameters. This distribution represents our prior beliefs about the model parameters, and can be used to restrict parameters to a range of values or even to limit the number of nonzero parameters, as discussed further below.

A. MCMC Methods

MCMC methods for parameter estimation sample from the posterior distribution, $p(\theta|Y)$, by constructing a Markov chain with $p(\theta|Y)$ as its stationary distribution. The key required elements are:

- *A likelihood model* that gives $p(Y|\theta)$. Assuming the model is continuous (e.g., an ordinary differential equation (ODE) model) and Gaussian experimental measurement error, the likelihood function is given by

$$L = e^{-\sum s \sum r(Y_{sim} - Y_{expt})^2/2\sigma^2},$$

  where S is a list of the observed species and T is a list of the time points at which observations are made. PTEMPEST allows other likelihood models, such as the built-in t-distribution [34], or any user-supplied function.

- *Prior distributions* on the parameters to be estimated. Uniform priors are a common choice when little is known about the parameters except for upper and lower limits. Priors can also be introduced to simplify a model by reducing some of its parameters to zero, a process called regularization. For example Lasso regularization [27] penalizes the sum of absolute values of parameters (the L1 norm), and Ridge regression [37] penalizes the sum of the squared parameter values (the L2-norm).

- A *proposal function* to define the probability distribution for the next parameter set to sample given the current set. A common choice is a normal distribution centered at the current value with a user-specified variance, which determines the effective step size. PTEMPEST uses a single adaptive step-size to determine the change in all parameters, but there are other MCMC implementations which permit different step sizes to govern changes in different directions in parameter space [9].

Following Metropolis *et al.* [26], we define the energy of a parameter set θ as

$$E(\theta) = -\log L(\theta) - \log p(\theta),$$

where L and p are the likelihood and prior distribution functions defined above.

1) Metropolis-Hastings algorithm—The Metropolis-Hastings (MH) algorithm is one of the most popular MCMC methods [6]. If we assume a symmetric proposal function, i.e., the probability of moving from a parameter set θ_i to θ_j equals that of moving from θ_j to θ_i, then the algorithm to sample from $p(\theta/Y)$ is as follows:

1. Select an initial parameter vector θ_0 that has energy $E(\theta_0)$ and set i 0.

2. For each step i =until $i = N$

    a. Propose a new parameter vector θ_{new} and calculate the $E(\theta_{new})$.

    b. Set $\theta_i = \theta_{new}$ with probability $\min(1, e^{-E})$, where $E = E(\theta_{new}) - E(\theta_{i-1})$ (acceptance). Otherwise, set $\theta_i = \theta_{i-1}$ (rejection).

    c. Increment i by 1.

2) Parallel Tempering—One of the key differences between MH and PT is the existence of a temperature parameter, β, that scales the effective "shallowness" of the energy landscape. Several Markov chains are constructed in parallel, each with a different β. A Markov chain with a β value of 1 samples the true energy landscape, while higher temperature chains have lower values of β and sample shallower landscapes with the acceptance probability now given by $\min(1, e^{-\beta E})$. Higher temperature chains accept unfavorable moves with a higher probability and therefore sample parameter space more broadly. Tempering refers to periodic attempts to swap configurations between high and low temperature chains. These moves allow the low temperature chain to escape from local minima and improve both convergence and sampling efficiency [8]. The PT algorithm is as follows:

1. For each of N swap attempts (called "swaps" for short)

    a. For each of N_c chains (these can be run in parallel)

        i. Run N_{MCMC} MCMC steps

        ii. Record the values of the parameters and energy on the final step.

b. For each consecutive pair in the set of chains in decreasing order of temperature, accept swaps with probability $\min(1, e^{\beta E})$, where $E = E_j - E_{j-1}$, and $\beta = \beta_j - \beta_{j-1}$, and E_j and β_j are the energy and temperature parameter respectively of the j chain.

Adapting the step size and the temperature parameter can further increase the efficiency of sampling [8]. However, varying parameters during the construction of the chain violates the assumption of a symmetric proposal function (also referred to as "detailed balance"), and it is advisable to do this during a "burn-in" phase prior to sampling.

3) Implementation—In this work we present PTEMPEST, a MATLAB-based tool for parameter estimation using PT that is integrated with the rule-based modeling software BioNetGen [16]. Models specified in the BioNetGen language (BNGL) can be exported as ODE models that are called as MATLAB functions by PTEMPEST. The BioNetGen commands `writeMfile` or `writeMexfile` are used to export models in MATLAB's M-file format, which uses MATLAB's built-in integrators, or as a MATLAB MEX-file, which encodes the model in C and invokes the CVODE library [17], which is usually much more efficient in our experience. For additional compatibility, models can be imported into BioNetGen in the System Biology Markup Language (SBML) [18], or the user can write their own cost function in MATLAB. The Bayesian parameter estimation capabilities of PTEMPEST complement those of another tool for performing parameter estimation on rule-based models, BioNetFit [30].

PTEMPEST uses adaptive step sizes and temperatures. The user provides the following hyper-parameters to control sampling: initial step size, initial temperature, and adaptation intervals and target acceptance probabilities for steps and swaps. At given intervals, the step acceptance probabilities and swap acceptance probabilities are calculated, and the step sizes and chain temperatures are adjusted to bring the step and swap acceptance probabilities closer to their target values respectively. For example, if the step acceptance rates are too high, the step size will be increased and vice versa. Similarly, if the swap acceptance rates are too high, the chain temperatures will be increased and vice versa. Although there are a considerable number of hyper-parameters associated with this method, we have found that the default values provided in PTEMPEST generally work well in practice.

The MATLAB source code for PTEMPEST along with model and data files used in the experiments described below are available at http://github.com/RuleWorld/ptempest.

B. Approximate Bayesian Computation methods

1) ABC rejection—The simplest ABC algorithm is a rejection algorithm [32], which involves repeatedly sampling a parameter vector θ_i from the prior distribution, simulating the model with the sampled parameters, and calculating the discrepancy (often in the form of a distance function) between the simulated data Y_{sim} and the experimental data Y_{expt}. If the discrepancy is below a threshold, ε, θ_i is accepted as a member of the posterior distribution; otherwise, it is discarded and another θ_i is drawn. This process continues until the number of samples reaches a specified number, resulting in an approximation of the distribution $p(\theta|$

$Y_{\text{sim}} - Y_{\text{expt}} < \varepsilon$), which in the limit of $\varepsilon \to 0$, will approach the true posterior distribution $p(\theta/Y_{\text{expt}})$.

2) Approximate Bayesian Computation-Sequential Monte Carlo (ABC-SMC)—
The ABC rejection algorithm can suffer from low acceptance rates [32]. The ABC-SMC algorithm uses a tolerance schedule to decrease, and sequentially constructs approximate posterior distributions of increasing accuracy, which eventually converge to the true posterior distribution [22], [32]. We use ABC-SMC to generate the results shown in Sec. III-B.

C. Metrics for algorithm performance comparisons

In our analyses we fit ODE models to synthetic data generated using fixed parameter values. For the comparison to ABC presented in Sec. III-B we used synthetic data with additional noise, as was provided in the ABC-SysBio example files.

For models containing 3–6 parameters, both the MH and PT algorithms find the global minimum, and we compared the performance using convergence time and sampling efficiency. The *convergence time* is defined as the number of MCMC steps before the energy drops below a specified threshold, determined empirically [4]. For PT convergence time is based on the number of MCMC steps in the lowest temperature chain. With uniform priors and data simulated without noise, the negative log likelihood approaches zero when the chain converges to the global minimum.

The *sampling efficiency* is defined as the ratio of the range of the posterior distribution to the range of the prior distribution, either for a model parameter that is known to be uniformly distributed, or for an added control parameter that does not contribute to the model output and therefore should be uniformly distributed.

For more complex models (11-25 parameters), we do not always obtain parameter sets that fit the data. In this case we compare the algorithms in terms of the negative log likelihood of the best fit parameter sets. In the case of uniform priors, this directly corresponds to the minimum energy attained by the Markov chain.

To compare disparate algorithms in terms of the total amount of computational resource used, we allowed each to perform a specified number of model integrations. For MH the number of model integrations is the number of MCMC steps, while for PT it is the number of MCMC steps times the number of chains run in parallel. For ABC algorithms, which use rejection sampling, the number of model integrations equals the total number of parameter sets evaluated to generate the desired number of samples.

III. RESULTS

A. Michaelis-Menten Kinetics

We start with a simple model to demonstrate how Bayesian methods can identify constrained parameter relationships even when individual parameters are unidentifiable. The Michaelis-Menten model describes enzyme substrate kinetics using the following scheme:

$$E + S \underset{k_r}{\overset{k_f}{\rightleftharpoons}} ES \xrightarrow{k_{cat}} E + P$$
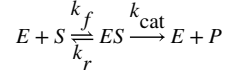
When the total enzyme concentration, $[E]_T$ is much smaller than that of the substrate, the rate of product formation is given by

$$\frac{d[P]}{dt} = \frac{k_{cat}[E]_T[S]}{(K_M + [S])},$$

where $K_M = (k_{cat} + k_r)/k_f$ is the Michaelis constant. The product trajectory only constrains k_{cat} and K_M, while the individual forward and backward rates k_f and k_r are unidentifiable. We generated a synthetic product trajectory using parameters $k_f = 10^{-2.77}, k_r = 10^{-1}, k_{cat} = 10^{-2}$, and constructed a likelihood function assuming 1% Gaussian error. The 3 model parameters are sampled in log-space, with uniform priors on the intervals $[-3,1]$, $[-1,3]$ and $[-3,3]$ for k_f, k_r and k_{cat} respectively. The fit is repeated 100 times using MH, and PT with 4 chains, starting from an initial parameter set of $[-1,1,0]$, corresponding to the midpoints of the priors. Both algorithms were run for 250,000 MCMC steps.

The quality of fit produced by MH and PT is comparable (Figure 1A-B). However, on average MH required 4203 MCMC steps to reach convergence, while PT required 369 (Figure 1C). Thus, even though each PT step needs 4 times as many model integrations, the total number of model integrations is smaller than for MH. This is consistent with the observation made in [8], that PT with M chains of length N can be more efficient than a single-chain Monte Carlo search of length MN. PT also has higher sampling efficiency for k_r and k_f compared to MH (Figure 1D).

As we would expect from the non-identifiability of k_f and k_r, the posterior distributions of $\log_{10}(k_r)$ and $\log_{10}(k_f)$ are uniform across the prior (Figure 1E), but their ratio is constrained (Figure 1F). $\log_{10}(k_{cat})$ is an identifiable parameter and has a constrained distribution centered at -2 (Figure 1E). The distributions shown in Figures 1E,F were obtained using PT with 4 chains run for 1,000,000 MCMC steps.

B. mRNA self-regulation

In this section we compare the efficiency of ABC-SMC, PT and MH for parameter estimation on a simple model of mRNA self-regulation (Figure 2A). The ABC-SysBio software is distributed with example files to estimate the parameters of this model assuming uniform priors using the ABC-SMC algorithm. The model has 5 parameters, one of which is

fixed [22]. The quality of fit is defined as the Euclidean distance between the fitted trajectory and the data. For the ABC-SMC algorithm we extended the default 18-step tolerance schedule provided in ABC-SysBio from 50-15 to a 23-step schedule from 50-5 and set the ensemble size as 100. We ran ABC-SMC 50 times, and found that each run used an average of 6.7×10^4 model integrations. We then ran 50 repeats of 4-chain PT for 16750 MCMC steps, and of MH for 6.7×10^4 MCMC steps, using a likelihood function with 1% Gaussian error. The sampling efficiency of PT and MH was compared using a control parameter as described above. The quality of fit produced by the MCMC-based algorithms is substantially higher than what we get from ABC-SMC (Figure 2B-D). PT takes fewer steps to reach convergence (Figure 2E), and has higher sampling efficiency than MH given the same number of model integrations (Figure 2F).

C. Model reduction with Lasso

In this section we demonstrate the use of MCMC approaches to perform model reduction, by coupling parameter estimation with regularization. Lasso regularization penalizes the L1-norm of the parameter vector while minimizing the cost function during parameter estimation. This performs variable selection by finding the minimum number of non-zero parameters required to fit the data [31]. The Lasso penalty is equivalent to assuming a Laplace prior on the parameters [27], and the width of the prior is inversely related to the regularization parameter that governs the strength of the penalty. Here, we present an example of using the Bayesian Lasso for model reduction, and compare the use of PT and MH for this problem.

A core model of negative feedback regulation with three processes (blue arrows in Figure 3A) was simulated to get a synthetic trajectory for species A using a value of 10 for all three rate constants (Figure 3C). Three extraneous process were added to the model (red arrows in Figure 3A), so that only a subset of the reactions in the reaction scheme are required to fit the data. We constructed a likelihood function assuming 2% Gaussian error, and assumed Laplace priors of width b on each of the 6 model parameters, where b is the regularization parameter that needs to be tuned. High values of b, i.e., wide priors, will not impact the log likelihood but will not achieve much variable selection. Conversely for low values of b most of the parameters will go to 0 at the cost of degrading the log likelihood.

Here, we tested a range of b values. For each we ran PT with 500,000 MCMC steps 50 times to obtain a distribution of negative log-likelihood values (Figure 3B). Figure 3C shows examples of fitted ensembles obtained with different regularization strengths. We chose the smallest value of b, 0.5, that does not significantly increase the negative log likelihood (Figure 3B), and used this for further analysis.

The posterior distributions for the model parameters obtained with regularization show the extraneous parameters peaking at 0, while the essential parameters have well defined distributions that peak close to their true values (Figure 3D, top row). Without regularization the extraneous parameters p_1, p_2 and p_5 (red boxes in Figure 3D) take on non-zero values and make the other parameters unidentifiable (Figure 3D, bottom row). PT converges faster

than MH (Figure 3E), but the sampling efficiencies calculated over 200,000 MCMC steps are comparable (Figure 3F).

D. Calcium signaling

The models considered so far have a relatively small number of parameters and both MH and PT achieve convergence readily. For models with more parameters and more complex dynamics, convergence becomes difficult to achieve. As an example, we consider a four-species model of calcium oscillations that has 12 free parameters [20]. The model describes the dynamics of G_a subunits of the G-protein, active PLC, free cytosolic calcium, and calcium in the endoplasmic reticulum. We generated synthetic data for free cytosolic calcium (Figure 4A), and constructed a likelihood function assuming 20% Gaussian error. The 12 free parameters were sampled in log-space with uniform priors, 6 units wide and centered at the true values. We generated 100 random initial parameter sets, and from each starting point sampled using MH, PT with 4 chains (PT-4) and PT with 6 chains (PT-6). Only a fraction of the chains converged to the global minimum in 500,000 MCMC steps. Figure 4A shows an example of an MH chain that has converged to a local minimum with high energy, and another of a PT chain that has converged to the global minimum. The distributions of minimum energy for chains obtained from each algorithm (Figure 4B) show that PT-6 found better fits than PT-4, which in turn did better than MH, which returned highly variable results and frequently did not reach the global minimum.

E. Negative feedback oscillator

We also considered the three species negative feedback oscillator from Tyson et al. [33], to evaluate the more difficult case of fitting a model to complex dynamics of multiple species. We generated synthetic data for all three model species under conditions where all three species undergo sustained oscillations. 11 model parameters are sampled in log-space, with uniform priors that are 10 units wide and centered at the true values. The likelihood function is a t-distribution with 10% error. We generated 15 random initial parameter sets, and from each starting set ran MH, PT with 4 chains (PT-4) and PT with 6 chains (PT-6) for 500,000 MCMC steps.

Figure 5A shows examples of chains converging to different minima. The top row shows an example of convergence to a high energy. As in the case of calcium signaling, PT with 6 chains outperforms PT with 4 chains, which in turn outperforms the MH algorithm in finding the global minimum (Figure 5B). Interestingly, the data that we generated did not sufficiently constrain the frequency of oscillations exhibited by the model, and we find parameter sets corresponding to different frequencies that all fit the data. Figure 5C shows the posterior distributions of the 11 model parameters corresponding to the fit shown in the middle panel of Figure 5A, obtained using PT with 6 chains. The first parameter shows 3 clear peaks, one of which is centered at the true value. Separating the parameter sets corresponding to these peaks shows that they correspond to specific differences in oscillation frequencies that are all part of the fitted ensemble (Figure 5D), reinforcing the need to use Bayesian methods with such problems.

F. Growth factor signaling model

Finally we apply MH and PT to a substantially larger model that has 24 parameters — a rule-based model of Shp2 regulation in growth factor signaling [3] that generates 149 species and 1032 reactions. We generated synthetic data for the micromolar concentration of phosphorylated receptors (pYR), an observable that combines the time courses of 136 model species, and constructed a likelihood function assuming 2% Gaussian error. The parameters were sampled in log-space with a uniform prior on the interval [−6,6]. We generated 25 random initial parameter sets, and from each starting point we obtain Markov chains with 200,000 MCMC steps using MH and PT with four chains. Figure 6A shows chain convergence to different minima, and Figure 6B shows that PT more consistently finds good fits than MH.

IV. DISCUSSION

In this study we have shown that even with relatively simple biochemical models, there are significant benefits to using PT over MH in terms of convergence speed and sampling efficiency. With more complex models we found that given a fixed budget of MCMC steps MH often fails to find the global minimum, whereas PT consistently succeeds. We also showed an example in which Bayesian parameter estimation can effectively perform model reduction through the introduction of a regularizing prior. While ABC methods constitute a popular class of alternative methods for Bayesian parameter estimation in cases where likelihood models are expensive or not available (such as for stochastic models), we found that PT outperformed ABC for parameter sampling on a relatively simple ODE model. Our direct performance comparison supports the previous observation [22] that likelihood-based methods are preferable to likelihood-free methods when likelihood models are feasible to compute, such as with ODE models.

One limitation of our evaluation procedure is that we have not attempted to compare wall clock times for the different algorithms. Instead, as a performance metric we have used the number of MCMC steps or the number of model integrations required, which are independent of the implementation. In practice, the fits reported in Sec. III can all be performed on a typical workstation computer in times ranging from a few minutes for the smallest model (Michaelis-Menten) to a few hours for the largest model (growth factor signaling). However, we found that despite requiring the same number of model integrations per processor per step, the single-chain MH sampling ran significantly faster per step (20–30%) in terms of wall clock time than PT with four or six chains. Preliminary tests showed that these differences likely arise from the requirement in our current implementation for each chain to complete a fixed number of steps before a swap is attempted. Parallel efficiency decreases when trajectories on different processors take different amounts of time to complete. We found that the difference in wall clock time decreased when the PT chains are all run at the same temperature, but so do the algorithmic benefits. When chains are run at different temperatures, the high temperature chains tend to sample parameter space more broadly, which results in greater variability in the model integration time [5] and causes slow

down due to the synchronization requirement. We plan to investigate asynchronous swapping between chains in order to alleviate this problem.

Another limitation of the current work is that the comparisons were made using specific choices for the hyper-parameters that control the PT algorithm, such as those that control step sizes for the moves and temperatures. Adjustment of these may result in further improvements to sampling efficiency and convergence rates. We would also like to investigate the effect of using different proposal functions, such as Hessian-guided MCMC [9], as well as different likelihood models.

While we have restricted our MCMC comparisons to MH, there has been considerable work toward improving the efficiency of MCMC methods, such as Differential Evolution Adaptive Metropolis (DREAM) [28] and Delayed Rejection Adaptive Metropolis (DRAM) [14]. It would be interesting to investigate whether parallel tempering could be fruitfully combined with these approaches.

Finally, PT, as it has been presented and used to this point both here and in the molecular simulation literature, is only a *moderately* parallel algorithm because it uses just a handful of chains. It remains to be seen whether using a much larger number of chains would retain the advantages of sampling simultaneously at multiple temperatures and result in further acceleration.

Acknowledgments

References

1. Ahearn TS, Staff RT, Redpath TW, Semple SIK. The use of the Levenberg–Marquardt curve-fitting algorithm in pharmacokinetic modelling of DCE-MRI data. Physics in Medicine and Biology. 2005; 50:N85–N92. [PubMed: 15843726]

2. Ashyraliyev M, Fomekong-Nanfack Y, Kaandorp JA, Blom JG. Systems biology: Parameter estimation for biochemical models. FEBS Journal. 2009; 276:886–902. [PubMed: 19215296]

3. Barua D, Faeder JR, Haugh JM. Structure-based kinetic models of modular signaling protein function: focus on Shp2. Biophysical journal. 2007; 92:2290–300. [PubMed: 17208977]

4. Bhatnagar N, Bogdanov A, Mossel E. The Computational Complexity of Estimating MCMC COnvergence Time. Approximation, Randomization, and Combinatorial Optimization Algorithms and Techniques Lecture Notes in Computer Science, vol 6845 Springer, Berlin, Heidelberg. 2011; 6845

5. Brown KS, Sethna JP. Statistical mechanical approaches to models with many poorly known parameters. Phys Rev E. Aug.2003 68:021904.

6. Chib S, Greenberg E. Understanding the Metropolis-Hastings algorithm. The American Statistician. 1995; 49(4):327–335.

7. Chylek LA, Akimov V, Dengjel J, Rigbolt KTG, Hu B, Hlavacek WS, Blagoev B. Phosphorylation site dynamics of early T-cell receptor signaling. PLoS ONE. 2014; 9(8):e104240. [PubMed: 25147952]

8. Earl DJ, Deem MW. Parallel tempering: Theory, applications, and new perspectives. Physical Chemistry Chemical Physics. 2005; 7(23):3910. [PubMed: 19810318]

9. Eydgahi H, Chen WW, Muhlich JL, Vitkup D, Tsitsiklis JN, Sorger PK. Properties of cell death models calibrated and compared using Bayesian approaches. Molecular Systems Biology. 2014; 9(1):644–644.

10. Fernández Slezak D, Suárez C, Cecchi GA, Marshall G, Stolovitzky G. When the optimal is not the best: Parameter estimation in complex biological models. PLoS ONE. 2010; 5(10)

11. Gao F, Han L. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. Computational Optimization and Applications. 2012; 51(1)

12. Goldstein B, Faeder JR, Hlavacek WS. Mathematical and computational models of immune-receptor signalling. Nature Reviews Immunology. 2004; 4:445–456.

13. Gutenkunst RN, Waterfall JJ, Casey FP, Brown KS, Myers CR, Sethna JP. Universally Sloppy Parameter Sensitivities in Systems Biology Models. PLOS Computational Biology. 2007; 3(10): 1871–1878. [PubMed: 17922568]

14. Haario H, Laine M, Mira A, Saksman E. DRAM: Efficient adaptive MCMC. Statistics and Computing. 2006; 16:339–354.

15. Hansmann UH. Parallel tempering algorithm for conformational studies of biological molecules. Chemical Physics Letters. 1997; 281:140–150.

16. Harris LA, Hogg JS, Tapia JJ, Sekar JA, Gupta S, Korsunsky I, Arora A, Barua D, Sheehan RP, Faeder JR. BioNetGen 2.2: Advances in rule-based modeling. Bioinformatics. 2016; 32(21):3366–3368. [PubMed: 27402907]

17. Hindmarsh AC, Brown PN, Grant KE, Lee SL, Serban R, Shumaker DE, Woodward CS. Sundials: Suite of nonlinear and differential/algebraic equation solvers. ACM Transactions on Mathematical Software. 2005; 31(3):363–396.

18. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, et al. The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. Bioinformatics. 2003; 19(4):524–531. [PubMed: 12611808]

19. Iadevaia S, Lu Y, Morales FC, Mills GB, Ram PT. Identification of optimal drug combinations targeting cellular networks: Integrating phospho-proteomics and computational network analysis. Cancer Research. 2010; 70(17):6704–6714. [PubMed: 20643779]

20. Kummer U, Olsen LF, Dixon CJ, Green AK, Bornberg-Bauer E, Baier G. Switching from simple to complex oscillations in calcium signaling. Biophysical Journal. 2000; 79(3):1188–1195. [PubMed: 10968983]

21. Lee REC, Walker SR, Savery K, Frank DA, Gaudet S. Fold Change of Nuclear NF- k B Determines TNF-Induced Transcription in Single Cells. Molecular Cell. 2014; 53(6):867–879. [PubMed: 24530305]

22. Liepe J, Kirk P, Filippi S, Toni T, Barnes CP, Stumpf MPH. A framework for parameter estimation and model selection from experimental data in systems biology using approximate Bayesian computation. Nature Protocols. 2014; 9(2):439–456. [PubMed: 24457334]

23. Lopez CF, Muhlich JL, Bachman JA, Sorger PK. Programming biological models in Python using PySB. Molecular Systems Biology. 2013; 9(1):1–19.

24. Lukens S, DePasse J, Rosenfeld R, Ghedin E, Mochan E, Brown ST, Grefenstette J, Burke DS, Swigon D, Clermont G. A large-scale immuno-epidemiological simulation of influenza A epidemics. BMC Public Health. 2014; 14(1):1019. [PubMed: 25266818]

25. Malkin AD, Sheehan RP, Mathew S, Federspiel WJ, Redl H, Clermont G. A Neutrophil Phenotype Model for Extracorporeal Treatment of Sepsis. PLoS Computational Biology. 2015; 11(10):1–30.

26. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of State Calculations by Fast Computing Machines. The Journal of Chemical Physics. 1953; 21(6):1087–1092.

27. Park T, Casella G. The Bayesian Lasso. Journal of the American Statistical Association. 2008; 103(482):681–686.

28. Shockley EM, Vrugt JA, Lopez CF. PyDREAM: High-dimensional parameter inference for biological models in Python. Bioinformatics. 2017

29. Sugita Y, Okamoto Y. Replica-exchange molecular dynamics method for protein folding. Chemical Physics Letters. 1999; 314:141–151.

30. Thomas BR, Chylek LA, Colvin J, Sirimulla S, Clayton AH, Hlavacek WS, Posner RG. BioNetFit: a fitting tool compatible with BioNetGen, NFsim and distributed computing environments. Bioinformatics. 2015; 32(5):798–800. [PubMed: 26556387]

31. Tibshirani R. Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. 1996; 58(1):267–288.

32. Toni T, Welch D, Strelkowa N, Ipsen A, Stumpf MPH. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. Journal of the Royal Society Interface. 2009; 6:187–202.

33. Tyson JJ, Chen KC, Novak B. Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. Current Opinion in Cell Biology. 2003; 15(2):221–231. [PubMed: 12648679]

34. Wasserman L. All of Statistics A Concise Course in Statistical Inference. 2005

35. Yao J, Pilko A, Wollman R. Distinct cellular states determine calcium signaling response. Molecular systems biology. 2016; 12(894):1–12.

36. Zhang F, Angermann BR, Meier-Schellersheim M. The Simmune Modeler visual interface for creating signaling networks based on bi-molecular interactions. Bioinformatics. 2013; 29(9):1229–30. [PubMed: 23508970]

37. Zou H, Hastie T. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society. 2005; 67:301–320.
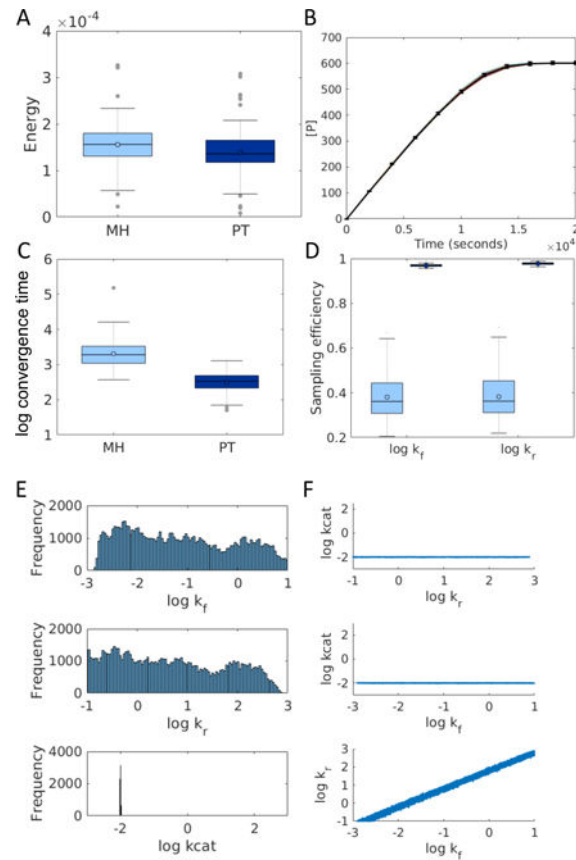
Fig. 1.
Parameter estimation for the Michaelis-Menten model. (A) Distribution of minimum energy values obtained via MH and PT. (B) Example of a fitted ensemble (colored lines) obtained for the synthetic data (black lines with error bars) using PT. (C) Distribution of convergence times for PT vs. MH with an energy threshold of 1. (D) Sampling efficiency for parameters k_r and k_f over 100 repeats using MH (light blue) and PT (dark blue). (E) Estimated posterior distributions for each of the model parameters. The x-axis limits are the uniform prior boundaries. (F) Scatter plots of sampled parameter sets for each pair of model parameters. Axis limits reflect prior boundaries.
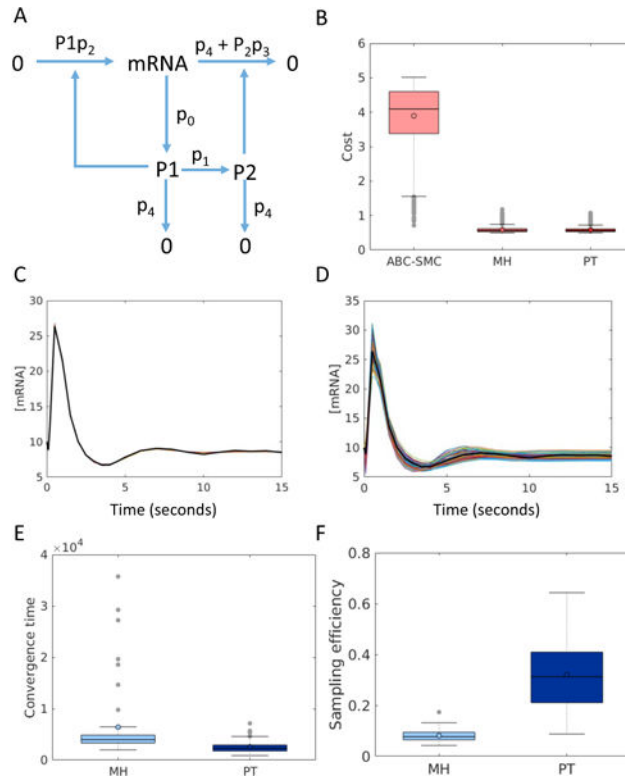
Fig. 2.
Parameter estimation for the model of mRNA self regulation. (A) Reaction network diagram
of the mRNA self regulation model from [22] (B) Quality of fit of the final ensemble
obtained from ABC-SysBio, PT and MH. The box plots show the distribution of the
Euclidean distances of the 100 members of each of 50 fitted ensembles from the synthetic
data. Example of a typical fitted ensemble obtained from (C) PT and (D) ABC-SysBio.
Black lines show the synthetic data, and the colored lines show the fitted trajectories.
[mRNA] refers to the number of mRNA molecules. (E) Distribution of convergence times
for PT vs. MH with an energy threshold of 20. PT takes on average ~2-fold fewer steps to
reach convergence. (F) Comparison of sampling efficiency of MH vs. PT.
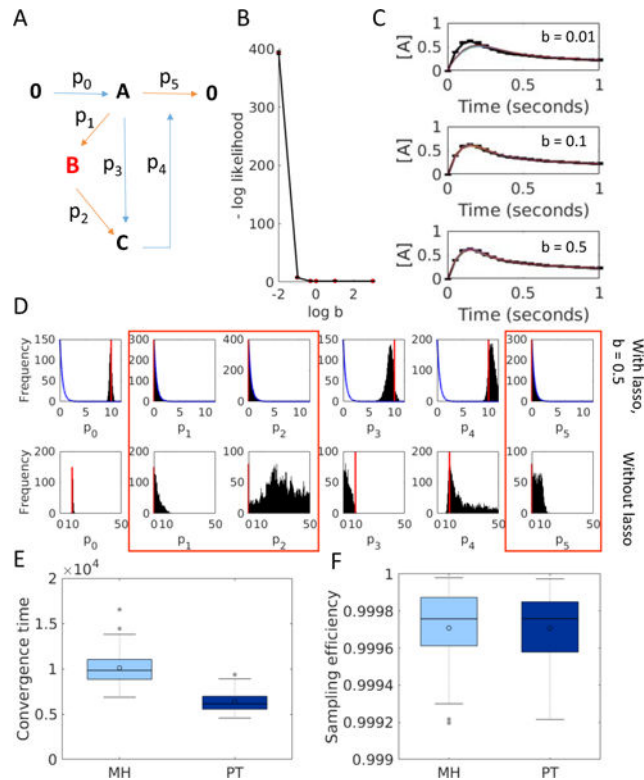
Fig. 3.
Model reduction with Lasso. (A) Reaction network diagram of a toy negative feedback model. The core model used to obtain the synthetic data for the fit is in blue, and extraneous elements are in red (B) Tuning the regularization parameter, i.e. the width of the Laplace prior, w.r.t the negative log likelihood of the fitted ensembles (C) Examples of fitted ensembles corresponding to different regularization strengths. Error bars show synthetic data. Solid lines show the simulated fits. (D) Posterior distributions with lasso (top row) for parameters show extraneous parameters peaking at 0. Red lines indicate true parameter values, and the blue lines show the Laplace prior ($b = 0.5$). The bottom row shows the posterior distributions obtained without Lasso. Red boxes indicate extraneous parameters. (E) Distribution of convergence times for PT vs. MH with an energy threshold of 65. (F) Distributions of of sampling efficiency for PT and MH across 50 repeats.
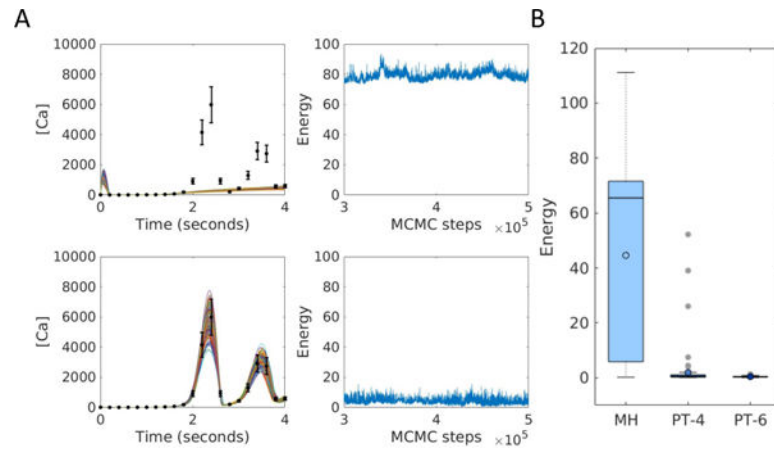
Fig. 4.
Parameter estimation for the model of calcium signaling. (A) Examples of convergence to a local minimum (top) and to the global minimum (bottom). Error bars show synthetic data. Solid lines show the simulated fits. The right column shows the energy chains corresponding to the fits on the left. (B) Distributions of the minimum energy from MH, PT-4 and PT-6 over 100 repeats.
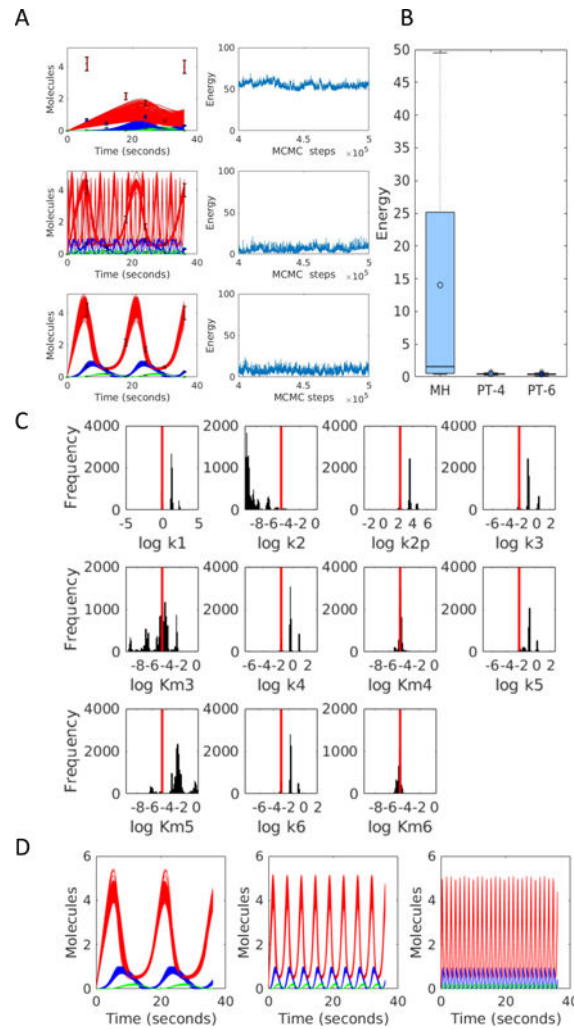
Fig. 5.
Parameter estimation for the negative feedback oscillator. (A) Examples of convergence to different minima. Error bars show synthetic data. Solid lines show the simulated fits. The three colors correspond to the three different model species. The right column shows energy chains corresponding to the fits shown on the left. (B) Distributions of the minimum energy by MH, PT-4 and PT-6 over 15 repeats. (C) Posterior distributions corresponding to the middle panel in (A). (D) Simulated fits corresponding to each of the three peaks in the posterior distribution of the first parameter shown in (C).
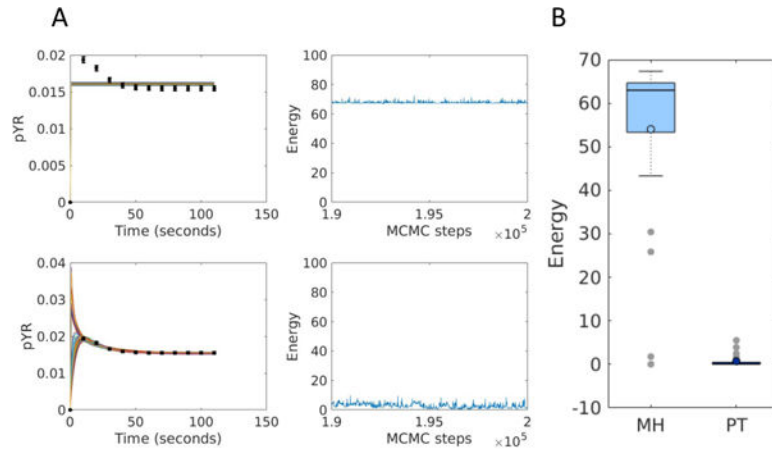
Fig. 6.
Parameter estimation for the growth factor signaling model. (A) Examples of convergence to different minima. Error bars show synthetic data. Solid lines show the simulated fits. The right column shows energy chains corresponding to the fits shown on the left. (B) Distributions of the minimum energy obtained via by MH and PT with 4 chains over 25 repeats.