BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

Hamiltonian Monte Carlo methods for efficient parameter estimation in steady state dynamical systems

Andrei Kramer[1*†], Ben Calderhead[2†] and Nicole Radde[1]

Abstract

Background: Parameter estimation for differential equation models of intracellular processes is a highly relevant bu challenging task. The available experimental data do not usually contain enough information to identify all parameters uniquely, resulting in ill-posed estimation problems with often highly correlated parameters. Sampling-based Bayesian statistical approaches are appropriate for tackling this problem. The samples are typically generated via Markov chain Monte Carlo, however such methods are computationally expensive and their convergence may be slow, especially if there are strong correlations between parameters. Monte Carlo methods based on Euclidean or Riemannian Hamiltonian dynamics have been shown to outperform other samplers by making proposal moves that take the local sensitivities of the system's states into account and accepting these moves with high probability. However, the high computational cost involved with calculating the Hamiltonian trajectories prevents their widespread use for all but the smallest differential equation models. The further development of efficient sampling algorithms is therefore an important step towards improving the statistical analysis of predictive models of intracellular processes.

Results: We show how state of the art Hamiltonian Monte Carlo methods may be significantly improved for steady state dynamical models. We present a novel approach for efficiently calculating the required geometric quantities by tracking steady states across the Hamiltonian trajectories using a Newton-Raphson method and employing local sensitivity information. Using our approach, we compare both Euclidean and Riemannian versions of Hamiltonian Monte Carlo on three models for intracellular processes with real data and demonstrate at least an order of magnitude improvement in the effective sampling speed. We further demonstrate the wider applicability of our approach to other gradient based MCMC methods, such as those based on Langevin diffusions.

Conclusion: Our approach is strictly beneficial in all test cases. The Matlab sources implementing our MCMC methodology is available from https://github.com/a-kramer/ode_rmhmc.

Keywords: MCMC methods, Parameter estimation, Hybrid monte carlo, Steady state data, Systems biology

Background

Parameter estimation is a major task that paves the way for building predictive models of intracellular regulation processes. Experimental data used for fitting these models however, often do not contain enough information to identify parameter values uniquely. Especially for parameter estimation of quantitative dynamic models such as ordinary differential equations (ODE) biological experiments usually do not have the required time resolution, rendering the estimation particularly challenging. Sparse data often lead to ill-posed optimization problems with multiple solutions that are indistinguishable in terms of fit quality, but might differ substantially when used for the prediction of new scenarios. Faced with these problems, optimization based point estimates are generally not appropriate, since an analysis of the parameter space around such points of high quality fits is often needed. Sampling-based Bayesian methods are advantageous in

*Correspondence: andrei.kramer@ist.uni-stuttgart.de
†Equal contributor
[1]Institute for Systems Theory and Automatic Control, Pfaffenwaldring 9, 70550 Stuttgart, Germany
Full list of author information is available at the end of the article

these settings as they offer an insight into the surroundings of local objective function minima and facilitate the uncertainty analysis of predictions. Consequently, statistical methods are being used more and more frequently for parameter estimation in systems biology (see e.g. [1,2]). In the Bayesian approach, parameters are considered random variables and inferences are formulated in terms of probability densities.

A major hindrance is the time required to generate samples from the resulting posterior distributions. Evaluation of the posterior can be computationally expensive; this is particularly the case for ODE models, where each sampling step requires numerical calculation of model trajectories. Metropolis-Hastings [3,4] algorithms are among the most popular choices for this task (see e.g. [2,5]). Variants of these core algorithms are also widely available in parameter estimation software packages, e.g. GNU MCSIM [6], or the MCMCSTAT MATLAB toolbox [7].

The speed of convergence of the Markov chain is crucial for the efficiency of the MCMC approach. Finding a good tuning of the Markov chain's transition distance can be time consuming and difficult [1,8,9], and it has been recognized that high correlations between the parameters can substantially slow down the convergence speed of standard MCMC algorithms [10]. In these cases the step size must often be chosen to be very small in order to obtain a reasonable acceptance rate, resulting in highly auto-correlated samples. A larger number of samples are then required to obtain low variance estimates of the inferred quantities.

Various strategies are employed to increase the distance of transitions in the sampling space, yet at the same time maintain a high acceptance rate. Several adaptive versions of the Metropolis-Hastings algorithm have been suggested in this context (see e.g. Ch.4 in [8]). These adaptation processes however only make proposals based on a global Gaussian approximation of the posterior, which can be a disadvantage when the posterior density has a complex lower scale structure.

Hybrid or Hamiltonian Monte Carlo (HMC) algorithms ([11]), can dramatically increase the acceptance rate while still providing samples with low auto-correlation. This is accomplished by an elaborate transition step that requires the integration of a secondary ODE system describing dynamics in parameter space that depend on the gradient of the target distribution. In this way, the HMC algorithm uses the structure of the ODE model more exhaustively. At first glance this approach increases the computational costs, but this is often compensated by the improved sample quality compared to the simpler algorithms. The advantage of low auto-correlations in typical HMC samples is that the sample size need not be as large to compute summaries to a satisfactory precision. A wasteful algorithm might provide a sample with high auto-correlations very quickly but at the expense of requiring a much larger number of samples to obtain low variance Monte Carlo estimates. Of course, these larger samples also need to be stored and empirical estimates of posterior predictive distributions using a larger number of sampled points will be computationally slower.

The statistical superiority of HMC has already been demonstrated on a variety of examples, e.g. using a 100-dimensional multivariate Gaussian distribution as target density (see Ch.5 in [8]); it seems to be a very promising approach, however HMC has so far rarely been used to infer parameters for nonlinear ODE models. A recent simple example is given in [1], where it is shown that the sample quality and convergence of HMC to the posterior can be improved even further by using an appropriate metric to define a Riemannian geometry over the parameter space. This approach employs the local sensitivity of the model at each step to inform the proposal distribution, leading to more efficient sampling in cases where the posterior has a complicated covariance structure.

The major difficulty that arises generally for HMC type algorithms when dealing with ODE models is that the model outputs and their sensitivities have to be simulated at every point along trajectories in parameter space. We address this computational issue by proposing an extension to HMC algorithms especially designed to sample efficiently from models with *steady state data* under multiple perturbations. In particular, we use sensitivity analysis within a Newton-Raphson approach to efficiently track the steady states across each of the Hamiltonian trajectories, instead of calculating them explicitly, thus drastically reducing this additional cost.

Steady state observations are typically not as informative as time series data, but can be obtained at comparatively low cost. They can be used in the first cycle of modeling where a qualitative model is validated via parameter fitting. If the model is not invalidated, dynamic data should be gathered for further analysis to narrow down model parameters further. The posterior parameter density from previous analysis cycles may be employed to inform the prior density of future experiments, and herein lies one of the benefits of the Bayesian approach. The properties of steady states offer the possibility to use analytical calculations to obtain output sensitivities. This is a lot faster than the numerical integration of dynamic sensitivities and improves sampling performance. We describe this further in Section 'Efficient calculation of geometry'. Typically, the steady state data will not be sufficient to uniquely identify all parameters; however issues of unidentifiability may also occur for dynamic time series data. Bayesian model analysis is designed to deal with precisely this case and allows full characterisation of the uncertainties that arise when making predictions.

We evaluate our approach on three different steady state models with real data: a model for Erk phosphorylation in the MAPK signaling pathway, and two alternative models for the phosphorylation of insulin receptor substrate IRS after insulin stimulation. We also provide standard HMC sampling speeds as reference.

Approach

Our modeling framework is motivated by real biological experiments: measurements on systems subject to long term or permanent perturbations, such as gene expression regulation via different promoters, silencing of genes or the stable transfection with variants of the system's enzymes with altered reactive properties. These types of experiment, among others, provide steady state responses to perturbations. In the following, we assume data is accumulated by observing the steady state under different conditions u.

Model structure

Intracellular processes are often described as biochemical reaction networks, i.e. ODE models with vector fields that are based on chemical reaction kinetics. Here we consider systems of the following form:

$$\dot{x} = f(x, \rho, u_j) \qquad j = 1, \ldots, n_{\mathrm{E}}\,, \tag{1}$$

where the state $x \in \mathbb{R}_+^n$ describes molecular concentrations, $\rho \in \mathbb{R}_+^m$ denotes the model parameters, $u_j \in \mathbb{R}_+^\nu$ is an input vector which describes experimental conditions such as the suppression of certain reactions or the introduction of mutants, and n_{E} denotes the number of performed experiments, i.e. the number of different input vectors. We consider constant inputs, the initial conditions $x(0) = x_0$ are assumed to be known and unparameterised and the function f shall be smooth. If the model exhibits complex bifurcations, special care has to be taken when dealing with initial conditions and basins of attraction. We do not address these issues here.

In this study we assume that the system converges to a steady state characterised by $f(\bar{x}, \rho, w) = 0$ for any available input w, which is the generic behaviour of such models [12]. Measurements are taken only after the system has reached steady state:

$$\begin{aligned} \bar{x}(\rho, u_j) &= \lim_{t\to\infty} \varphi(t, x_0, \rho, u_j)\,, \\ z_{ij} &= h_i(\bar{x}, \rho, u_j) = C\bar{x}(\rho, u_j), \qquad C \in \mathbb{R}^{\nu\times n}\,, \end{aligned} \tag{2}$$

where $z \in \mathbb{R}^{\nu\times n_{\mathrm{E}}}$ is the observable model output when input u_j is applied, and $\bar{x}(\rho, u_j)$ is the steady state of the respective trajectory $\varphi(t, x_0, \rho, u_j)$. We assume linear output functions characterized by the real matrix C.

Standard HMC methods and the proposed Newton-Raphson variants differ in the way the steady states are calculated, but are both initialized with a numerical solution to the initial value problem. The parameters ρ are always positive in chemical reaction networks. In order to avoid dealing with borders of the positive orthant it is advantageous to sample in logarithmic space $\theta = \log(\rho)$. Since we operate on θ we will drop ρ from all subsequent expressions and consider it implicitly understood that the model functions f and h will perform the transformation (the same applies to symbolic expressions like $\bar{x}(\theta, u)$).

Measurements are obscured by noise δ_{ij}, that is, the observed data

$$\mathcal{D} = \left\{ \left(y_{ij}, \sigma_{ij}\right) : i = 1, \ldots, \nu; j = 1, \ldots, n_{\mathrm{u}} \right\} \tag{3}$$

relates to the measurement model in this way,

$$y_{ij} \overset{!}{=} z_{ij} + \delta_{ij}, \qquad \delta_{ij} \sim \mathcal{N}\left(0, \sigma_{ij}^2\right)\,. \tag{4}$$

Gaussian noise is important for a vast amount of biological applications. However, the noise model is not of crucial importance to the sampling algorithms and can be changed as needed [13]. We assume that variances σ_{ij}^2 are available, e.g. from empirical variance estimates of different experimental replicates.

Bayesian parameter estimation

Equipped with our model and a data set $\mathcal{D}$, we define the likelihood function $p(\mathcal{D}|\theta)$, as a measure of plausibility for each θ. In the case of a Gaussian error model, the likelihood given all experimental measurements follows as

$$p(\mathcal{D}|\theta) \propto \exp\left(-\frac{1}{2} \sum_{j=1}^{n_{\mathrm{E}}} \sum_{i=1}^{\nu} \left(\frac{y_{ij} - h_i(\bar{x}, \theta, u_j)}{\sigma_{ij}} \right)^2 \right)\,.$$

We note that we often employ the log-likelihood in many calculations, which we denote $L_{\mathcal{D}}(\theta) = \log p(\mathcal{D}|\theta)$. The likelihood of a set of parameters therefore requires the steady state solution $\bar{x}(\theta, u_j)$ of the ODE model, which is usually calculated by numerical integration of (1) until convergence is achieved. In a Bayesian approach, prior knowledge regarding the plausible values of $p(\theta)$ is incorporated via Bayes' theorem:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})}\,, \tag{5}$$

where $p(\theta)$ and $p(\theta|\mathcal{D})$ are the prior and posterior distributions, respectively. In this work we assume Gaussian priors $\mathcal{N}(\mu, \Xi)$. The evidence $p(\mathcal{D})$ is a normalization constant that is independent of θ and is not needed during Markov chain Monte Carlo sampling, since this term cancels in the calculation of the Metropolis-Hastings acceptance ratio. Expected values of a function $F(\theta)$ with respect to the posterior,

$$E_{p(\theta|\mathcal{D})}[F(\theta)] = \int F(\theta)p(\theta|\mathcal{D})\,d\theta\,, \tag{6}$$

may be estimated using Monte Carlo integration given posterior samples,

$$\hat{F} = \frac{1}{N}\sum_{i=1}^{N} F(\theta_i), \qquad \theta_i \sim p(\theta|\mathcal{D})\,. \tag{7}$$

Riemannian structure of parameter space

Exploration of the posterior distribution for models defined by systems of ODEs is often severely hampered by the strong correlation structure present in the parameter space, which makes it difficult to propose MCMC moves that are far from the current point and accepted with high probability. Recent advances in MCMC attempt to circumvent these issues by utilising the underlying geometric structure induced by the sensitivity of a statistical model [1]. In addition to exploiting gradient information, we can construct MCMC algorithms based on higher order geometric structure by considering the expected Fisher Information, which [14] noted satisfies all the mathematical properties of a metric tensor and hence induces a Riemannian geometry on the parameter space. The use of this geometry allows us to define a measure of distance between sets of parameters in terms of the change in posterior probability, rather than changes in the values of the parameters themselves. In other words, MCMC algorithms based on Riemannian geometry make local moves according to a local coordinate system that automatically adapts based on the local sensitivity of the model, taking small steps in directions of high sensitivity and bigger steps in directions of lower sensitivity, while also taking into account local correlation between pairs of parameters. Such approaches have been shown to work incredibly well on a variety of complex statistical models [1], although computational expense often remains an issue for some classes of models.

The main quantity of interest here is the metric tensor. From the metric tensor, gradient and log-likelihood, we can define a Hamiltonian Monte Carlo algorithm using Riemannian coordinates, rather than the standard Euclidean coordinate system which is typically used. Both of these algorithms are given in the next section, and we refer the reader to [15] for an introductory exposition of Riemannian geometry for MCMC. We may define the (r,s)th element of a metric $G(\theta)$ based on the posterior distribution in the following manner,

$$\begin{aligned} G(\theta)^{rs} &= E_{p(y|\theta)}\left[\frac{\partial \log p(\theta|\mathcal{D})}{\partial\theta_r}\frac{\partial \log p(\theta|\mathcal{D})}{\partial\theta_s}\right] \\ &= \underbrace{E_{p(y|\theta)}\left[\frac{\partial L_{\mathcal{D}}(\theta)}{\partial\theta_r}\frac{\partial L_{\mathcal{D}}(\theta)}{\partial\theta_s}\right]}_{G_y(\theta)^{rs}} - \underbrace{E_{p(y|\theta)}\left[\frac{\partial^2 \log p(\theta)}{\partial\theta_r\partial\theta_s}\right]}_{-(\Xi^{-1})^{rs}}, \end{aligned}$$

where $G_y(\theta)$ is the expected Fisher Information and Ξ is the covariance of the prior. We note that the normalising factor of the likelihood, and prior for that matter, is a constant with respect to θ and vanishes in the derivative of its logarithm. One of the advantages of employing Riemannian geometry is that the calculation of $G_y(\theta)$ requires only *first order* sensitivities $S_l^{\,r} = \partial\bar{x}_l/\partial\theta_r$. For steady state ODE models, we can calculate a general expression for the expected Fisher Information based on a likelihood derived from Gaussian measurement errors with a linear observation model, as defined in (1) and (2):

$$\begin{aligned} G_y(\theta)^{rs} &= E_{p(y|\theta)}\left[\left(\frac{\partial}{\partial\theta_r}L_{\mathcal{D}}(\theta)\right)\left(\frac{\partial}{\partial\theta_s}L_{\mathcal{D}}(\theta)\right)\right] \\ &= E_{p(y|\theta)}\left[\left(\sum_{i_1 j_1}\left(\frac{y_{i_1 j_1} - \sum_l C_{i_1 l}\bar{x}_l(\theta, u_{j_1})}{\sigma^2_{i_1 j_1}}\right)\sum_l C_{i_1 l} S_l^{\,r}\right)\right. \\ &\quad \times \left.\left(\sum_{i_2 j_2}\left(\frac{y_{i_2 j_2} - \sum_k C_{i_2 k}\bar{x}_k(\theta, u_{j_2})}{\sigma^2_{i_2 j_2}}\right)\sum_k C_{i_2 k} S_k^{\,s}\right)\right] \\ &= \int \sum_{ij}\frac{(y_{ij} - \sum_l C_{il}\bar{x}_l(\theta, u_j))^2}{\sigma^2_{ij}\sigma^2_{ij}}\sum_l C_{il}S_l^{\,r} \\ &\quad \times \sum_k C_{ik}S_k^{\,s}\prod_{\iota\kappa} p(y_{\iota\kappa}|\theta)dy_{\iota\kappa} + \underbrace{0}_{\text{asymmetric terms: } i_1 \neq i_2, j_1 \neq j_2} \\ &= \sum_{ij}\sum_l C_{il}S_l^{\,r}\sum_k C_{ik}S_k^{\,s} \\ &\quad \times \int \frac{\left(y_{ij} - \sum_l C_{il}\bar{x}_l\left(\theta, u_j\right)\right)^2}{\sigma^2_{ij}\sigma^2_{ij}} p(y_{ij}|\theta)dy_{ij} \\ &= \sum_{ij}\sum_l C_{il}S_l^{\,r}(\theta, u_j)\sum_k C_{ik}S_k^{\,s}(\theta, u_j)\frac{\sigma^2_{ij}}{\sigma^2_{ij}\sigma^2_{ij}}\,. \end{aligned} \tag{8}$$

This calculation yields, taking the prior's contribution into account, the overall metric tensor G comprised of an inner product of the sensitivity matrices and the a-priori covariance matrix:

$$G(\theta) = \sum_j S(\theta, u_j)^{\mathrm{T}} C^{\mathrm{T}} \Sigma_j^{-1} C S(\theta, u_j) + \Xi^{-1}\,, \tag{9}$$

where $\Sigma_j = \operatorname{diag}\left(\sigma^2_{\cdot j}\right)$.

Methods

We have named the algorithms presented in this work using the prefix NR for *Newton-Raphson* and the prefix RM for *Riemannian Manifold*, which will be further explained in this section. When we define the variants of Hamiltonian Monte Carlo, we restrict our description to those aspects of the algorithm that are impacted by the modifications we make. We note that the modifications do not affect the correctness of the algorithms, and we therefore refer the reader to the original HMC and RMHMC publications for proofs of correctness and convergence [1,11].

The original HMC algorithm is defined on Euclidean space and performs well in practice for probability distributions that do not exhibit strong correlations between parameters. The state of the art RMHMC performs far better for correlated distributions, however the algorithm is computationally more complex to implement. We consider both HMC and RMHMC on our variety of examples, and then present an approach for significantly decreasing the computational cost of implementing both algorithms for steady state systems.

Hamiltonian Monte Carlo

The Hamiltonian Monte Carlo algorithm can be constructed by introducing an auxiliary variable to extend the state space. We may interpret the auxiliary variable as a momentum variable and the negative log joint distribution may be interpreted as a Hamiltonian system [1,16]. The main idea is that the induced dynamics of this system may then be used for proposing moves within an MCMC scheme. This is desirable since the dynamics may propose points that are far from the current point and accepted with high probability.

We begin by rewriting the posterior probability as

$$P(\theta|\mathcal{D}) \propto \exp(-V_{\mathcal{D}}(\bar{x},\theta,u)) \tag{10}$$

where

$$V_{\mathcal{D}}(\bar{x},\theta,u)) = \frac{1}{2}\sum_{j=1}^{n_{\mathrm{u}}}\sum_{i=1}^{\nu}\left(\frac{y_{ij}-h_i(\bar{x},\theta,u_j)}{\sigma_{ij}}\right)^2 - \log(p(\theta))$$

The sampling space is then extended by introducing the momentum variable η, and we may write the joint distribution as

$$\begin{aligned} p(\eta,\theta|\mathcal{D}) &= e^{-H(\eta,\theta)} \qquad (11)\\ &= \exp\left(-\frac{1}{2}\eta^{\mathrm{T}}\eta\right)\exp\left(-V_{\mathcal{D}}\left(\left\{\bar{x}_{\theta,u,x_0}\right\},\theta,u\right)\right) . \end{aligned}$$

We note that the Hamiltonian function $H(\eta,\theta)$ is simply the negative log joint distribution of our extended state space and can be used to calculate Hamiltonian trajectories according to the differential equations defined in the algorithm below. Given current values for the parameter and momentum variables, we can simulate the Hamiltonian dynamics to propose a new pair of parameter and momentum variables, which are then accepted according to a Metropolis-Hastings ratio to ensure convergence to the correct stationary distribution. The advantage of this approach is that this ratio may be close to 100%, far higher than the typical optimal acceptance ratios for other MCMC algorithms, which are typically between 20% and 60%. The standard HMC algorithm is given by,

1. **Transition step**
   Starting at $\theta =: \theta(0) = \theta_0$, solve the differential equations,

   $$\begin{aligned} \partial_\tau \eta(\tau) &= -\nabla_\theta V_{\mathcal{D}}\left(\left\{\bar{x}_{\theta,u,x_0}\right\},\theta,u\right) ,\\ \partial_\tau \theta(\tau) &= \eta , \end{aligned} \tag{12}$$

   for $\tau \in [\,0,\mathcal{T}]$ with initial conditions:

   $$\begin{aligned} \eta(0) &\sim \frac{1}{\sqrt{2\pi}^{m}}\exp\left(-\frac{1}{2}\eta^{T}\eta\right) , & \eta' &:= \eta(\mathcal{T}) ,\\ \theta(0) &= \theta_0 , & \theta' &:= \theta(\mathcal{T}) , \end{aligned} \tag{13}$$

   where the proposed parameter and momentum variables at time $\mathcal{T}$ are given on the right. The above equations are Hamilton's equations of motion for a particle with momentum η in a potential field $V_{\mathcal{D}}(\{\bar{x}_{\theta,u,x_0}\},\theta,u)$.
2. **Acceptance step**
   Accept θ' and η' with probability

   $$\begin{aligned} P_{\mathrm{acc.}} &= \min\left(1,\frac{p(\eta',\theta'|\mathcal{D})}{p(\eta,\theta|\mathcal{D})}\right)\\ &= \min\left(1,\exp\left(H(\eta,\theta)-H(\eta',\theta')\right)\right) . \end{aligned} \tag{14}$$

The numerical solution of (12) must be calculated using a symplectic integrator [1], which is exactly time reversible and volume preserving; these properties are required for ensuring convergence to the correct stationary distribution. Since the Hamiltonian laws of motion conserve energy, an analytic solution would provide sample points with perfect acceptance. However the required numerical integrators introduce a small amount of error and typically sample at very high but lower than 100% acceptance.

This standard version of HMC also profits from efficiently calculating the sensitivities and steady states along Hamiltonian trajectories using our approach based on a multivariate Newton-Raphson method, which we explain in Section 'Efficient calculation of geometry'. We call this version NR-HMC and include it in our performance evaluation.

Riemannian manifold Hamiltonian Monte Carlo

Hamiltonian Monte Carlo can also be defined using the induced Riemannian geometry, rather than the usual Euclidean geometry. The RMHMC algorithm may be derived in a similar manner to HMC, except now the momentum variable η is defined to be a Gaussian distribution with *position specific* covariance matrix, such that the joint distribution follows as,

$$\begin{aligned} p(\eta,\theta|\mathcal{D}) &= e^{-H(\eta,\theta)} \qquad (15)\\ &= \exp\left(-\frac{1}{2}\eta^{\mathrm{T}}\Sigma(\theta)^{-1}\eta\right)\exp\left(-V_{\mathcal{D}}\left(\{\bar{x}_{\theta,u,x_0}\},\theta,u\right)\right), \end{aligned}$$

where $\Sigma(\theta) = G(\theta)$. In other words, the dynamics now take into account the second order sensitivities of the statistical model of interest, since the covariance of the momentum variable is based on the expected Fisher Information at each point.

This Hamiltonian system is however harder to integrate numerically, since its equations of motion are now defined implicitly due to the dependence of the momentum on the position parameter θ. The generalised Leapfrog integrator is the simplest such algorithm for this task, however its use may significantly add to the computational expense of simulating proposals, since a much larger number of likelihood evaluations are generally necessary to solve the equations of motion compared to standard HMC [1,15]. In addition to the state sensitivities, we require the gradients of $G(\theta)$ for solving the Hamiltonian system, which necessitates the calculation of second order sensitivities, denoted from now on by $\widetilde{S}$. This provides the motivation for the next subsection, in which we propose a computationally efficient approach to evaluate the required geometric quantities for enabling faster inference in steady state ordinary differential equation models.

Efficient calculation of geometry

Practical implementation of Hamiltonian Monte Carlo samplers depends critically on the ability to efficiently calculate the required geometric quantities. Here we show how this may be done for steady state systems.

Output sensitivities $dh_i(\bar{x}, \theta, w)/d\theta_k$ (for any given input $w = u_{[1,n_{\mathrm{E}}]}$) occur in several places in the RMHMC algorithm. For ODE models, they are needed to construct the metric tensor G and the gradient of $V_{\mathcal{D}}(\{\bar{x}\}, \theta, u)$. In the case of steady state data, the sensitivities can be obtained from the steady state condition:

$$\begin{aligned}
0 &\overset{!}{=} f_i(\bar{x}(\theta, w), \theta, w)\\
0 &\overset{!}{=} \frac{df_i(\bar{x}(\theta, w), \theta, w)}{d\theta_j}\\
&= \sum_{l=1}^{n} \frac{\partial f_i(\bar{x}, \theta, w)}{\partial \bar{x}_l} \frac{\partial \bar{x}_l(\theta, w)}{\partial \theta_j} + \frac{\partial f_i(\bar{x}, \theta, w)}{\partial \theta_j}\,.
\end{aligned} \tag{16}$$

We will drop the arguments of $\bar{x}$ to shorten notation. In (16) we see that the steady state sensitivity is obtained by solving the following linear algebraic equation,

$$0 = J_f(\bar{x}, \theta, w) S(\theta, w) + K(\bar{x}, \theta, w)\,, \tag{17}$$

$$\Rightarrow S(\bar{x}, \theta, w) = -J_f(\bar{x}, \theta, w)^{-1} K(\bar{x}, \theta, w)\,,$$

where $K(\bar{x}, \theta, w)_i^{\,j} = \partial_{\theta_j} f_i(\bar{x}, \theta, w)$. We denote the solution to (17) as $S(\bar{x}, \theta, w)$, which is easy to obtain when the Jacobian is invertible[a]. Similarly, we can write the following equation for the second order sensitivity,

$$\begin{aligned}
0 &= \frac{d}{d\theta_k}\left(\sum_l \frac{\partial f_i(\bar{x}, \theta, w)}{\partial \bar{x}_l} \frac{\partial \bar{x}_l(\theta, w)}{\partial \theta_j} + \frac{\partial f_i(\bar{x}, \theta, w)}{\partial \theta_j}\right)\\
&= \sum_{r=1}^{n} \sum_{l=1}^{n} \left(\frac{\partial^2 f_i(\bar{x}, \theta, w)}{\partial \bar{x}_r \partial \bar{x}_l} \frac{\partial \bar{x}_r}{\partial \theta_k} + \frac{\partial^2 f_i(\bar{x}, \theta, w)}{\partial \theta_k \partial \bar{x}_l}\right) \frac{\partial \bar{x}_l(\theta, w)}{\partial \theta_j}\\
&\quad + \sum_{l=1}^{n} \frac{\partial f_i(\bar{x}, \theta, w)}{\partial \bar{x}_l} \frac{\partial^2 \bar{x}_l(\theta, w)}{\partial \theta_k \partial \theta_j}\\
&\quad + \sum_{r=1}^{n} \frac{\partial^2 f_i(\bar{x}, \theta, w)}{\partial \bar{x}_r \partial \theta_j} \frac{\partial \bar{x}_r}{\partial \theta_k} + \frac{\partial^2 f_i(\bar{x}, \theta, w)}{\partial \theta_k \partial \theta_j}\,,
\end{aligned} \tag{18}$$

leading to a linear equation for the second order sensitivity $\widetilde{S}_i^{\,jk} = \partial_{\theta_k} S_i^{\,j}$,

$$\begin{aligned}
0 = &\sum_{r=1}^{n} \sum_{l=1}^{n} \left(\frac{\partial^2 f_i(\bar{x}, \theta, w)}{\partial \bar{x}_r \partial \bar{x}_l} S_r^{\,k} + \frac{\partial J_i^{\,l}(\bar{x}, \theta, w)}{\partial \theta_k}\right) S_l^{\,j} + \sum_{l=1}^{n} J_i^{\,l} \widetilde{S}_l^{\,jk}\\
&+ \sum_{r=1}^{n} \frac{\partial^2 f_i(\bar{x}, \theta, w)}{\partial \bar{x}_r \partial \theta_j} S_r^{\,k} + \frac{\partial^2 f_i(\bar{x}, \theta, w)}{\partial \theta_k \partial \theta_j}\,.
\end{aligned} \tag{19}$$

Again, the existence of a solution depends on the invertibility of the Jacobian $J_f(\bar{x}, \theta, w)$. We note that the same LU-decomposition of J_f can be used for the first and second order sensitivities. Usually all derivatives of f appearing in (16) and (19) can be calculated analytically; for large systems, a symbolic calculation package will do the job, e.g. GiNaC, GNU Octave[forge]'s symbolic package or MATLAB's symbolic toolbox. A particularly convenient way of storing the models and doing the symbolic calculations is VFGEN [17] which provides MATLAB output.

Equipped with these instructions for sensitivity calculations we can easily calculate the metric $G(\theta)$ and the gradient of the log-likelihood for a given value of θ,

$$\begin{aligned}
-\frac{dV_{\mathcal{D}}(\bar{x}, \theta, u)}{d\theta_k} &= -\frac{1}{2} \frac{d}{d\theta_k} \sum_{j=1}^{n_{\mathrm{E}}} \sum_{i=1}^{\nu}\\
&\quad \times \left(\frac{y_{ij} - h_i(\bar{x}(\theta, u_j), \theta, u_j)}{\sigma_{ij}}\right)^2 - \frac{d\log(p(\theta))}{d\theta_k},\\
&= \sum_{j=1}^{n_{\mathrm{E}}} \sum_{i=1}^{\nu} \left(\frac{y_{ij} - \sum_l C_{il} \bar{x}_l(\theta, u_j)}{\sigma_{ij}^2}\right)\\
&\quad \times \sum_l C_{il} S_l^{\,k}(\bar{x}, \theta, u_j) - \frac{d\log(p(\theta))}{d\theta_k}\,.
\end{aligned} \tag{20}$$

Although HMC algorithms make large transitions across the parameter space, the transitions are constructed using multiple small integration steps of the Hamiltonian dynamics (12). Since these parameter changes in the integration are gradual, the sensitivity can be used directly to estimate the steady state $\bar{x}(\theta + \Delta, w)$ after each small parameter step Δ,

$$\bar{x}_i(\theta + \Delta, w) = \bar{x}_i(\theta, w) + \sum_{j=1}^{m} S_i^j \Delta_j \,, \tag{21}$$

For this reason, it is very convenient and efficient to recalculate the steady states using a Newton Raphson method, where we begin with $\bar{x}_0$ at the estimate (21),

$$\bar{x}_{r+1} = \bar{x}_r - J_f(\bar{x}_r, \theta, w)^{-1} f(\bar{x}, \theta, w) \,, \tag{22}$$

and for any input w, (22) is iterated until satisfactory precision is reached, otherwise we proceed exactly as in the original HMC algorithm. We call these variants NR-HMC and NR-RMHMC.

Sampling efficiency

For the estimation of the auto-correlation we employed the MATLAB script `UWerr` documented in [18]. By measuring the execution time t_{E} of the sampling we can calculate an effective sampling speed, corrected for auto-correlations:

$$v = \frac{N}{2\tau_{\mathrm{int.},L}\, t_{\mathrm{E}}} \,, \tag{23}$$

where N is the sample size and $\tau_{\mathrm{int.},L}$ the integrated auto-correlation length with respect to the estimation of the mean log-likelihood.

Large auto-correlations reduce the effective speed v: the sampled points strongly depend on their predecessors and many Markov chain moves are required to obtain independent information about the posterior distribution. A sampling method with increased cost per Markov chain transition might outperform a simple-but-cheap method if the returned points are sufficiently less correlated. The effective speed v takes the purpose of sampling into account. When comparing algorithms we also list a relative speed v_r, where we normalize each v_i using the lowest observed speed.

Results and discussion

In this section we apply the Bayesian parameter estimation framework to three examples from literature, which feature the type of data and modeling approach described in Section 'Methods'. A comparison of the performance of all algorithms is provided in Additional file 1: Figure S1.

All simulations were done in MATLAB. We used the SBPOP Toolbox bundle [19] to store the models and solve the ODE ($\bar{x}$) for RMHMC. The implementation makes use of the symmetry of the second order sensitivity: $\widetilde{S}_i^{jk} = \widetilde{S}_i^{kj}$, and reuses appropriate fluxes, nevertheless the integrator used by the toolbox has to integrate an initial value problem with $n + nm + nmm$ state variables (number of states, number of first order sensitivities, number of second order sensitivities).

The models are stored as symbolic variables for the Newton-Raphson type algorithms, symbolic calculations of the required derivatives of f then yield a standard matlab function for the Jacobian and the sensitivities. All linear equations are solved using MATLAB's backslash operator. We provide MATLAB code for easy reproduction of all of the results in this section.

Since the methods we used are not restricted to the RMHMC algorithm, we also tested the simplified Metropolis-adjusted Langevin algorithm (SMMALA) [1]. We applied the same modifications to SMMALA and measured the relative sampling speed, which can be inspected in Table 1.

Example: Erk phosphorylation in MAPK signaling

We consider a modification of the model for Erk phosphorylation in the MAPK signaling cascade introduced in [20], who investigated robustness of steady state phosphorylation of Erk with respect to changes in the total Erk amount $\mathrm{Erk_T} = u$:

$$\begin{aligned}
\dot{x}_1 &= \rho_1 \frac{u}{1+u} - (2\rho_1 g(u) + \rho_2)x_1 + (\rho_2 - \rho_1 g(u))x_2 \,,\\
\dot{x}_2 &= \rho_1 g(u) x_1 - \rho_2 x_2 \,,\\
g(u) &= \frac{1}{1+u}
\end{aligned} \tag{24}$$

where $x = ([\mathrm{pErk}], [\mathrm{ppErk}])^{\mathrm{T}}$ are the once and twice phosphorylated modifications of Erk.

According to the conclusions in [20] the robustness of this system with respect to varying $\mathrm{Erk_T}$ is due to negative feedback from Erk to Raf and MEK; this was not investigated by the authors using an ODE model but directly in experiment and by modifying the steady state solution of the open loop system. To account for this negative feedback we modified the phosphorylation rate of the

Table 1 Effective sampling speed measurements for SMMALA and the modified NR-SMMALA

Problem size		NR-SMMALA	SMMALA
2 × 2	v in s^{-1}	90 ± 2	71 ± 2
	$\tau_{\mathrm{int.},L}$	0.88 ± 0.02	0.88 ± 2
3 × 6	v in s^{-1}	62 ± 1	32 ± 1
	$\tau_{\mathrm{int.},L}$	1.20 ± 0.03	1.52 ± 0.04
6 × 14	v in s^{-1}	0.30 ± 0.07	0.031 ± 0.006
	$\tau_{\mathrm{int.},L}$	290 ± 67	166 ± 31

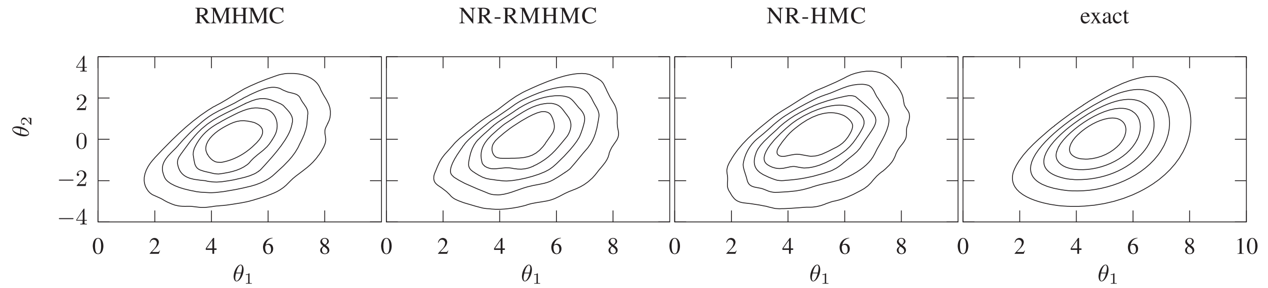

Figure 1 MAPK posterior comparisons fltr. (1) posterior inferred from RMHMC sample. (2) Newton Raphson driven RMHMC posterior inferred from sample. (3) posterior inferred from NR-HMC sample (4) exact posterior on a grid. We used a kernel density estimator (*kde*) to infer densities from samples.

original model by multiplying it with $u/(1+u)$. This modification of the model does provide negative feedback, though we do not suggest any specific mechanism or interpretation for it, but rather we aim at illustrating how a modeling hypothesis can be tested. The two different Erk variants can be knocked out individually, but have a very similar function. This enables the experimentalists to reduce the amount of Erk to several intermediate levels. For more details on the biological system we refer to [20].

The normalized data provided in the supplementary material of [20] contains 10 steady state measurements of x_2 obtained with western blots under different perturbation conditions in which Erk_1 and/or Erk_2 were knocked down, resulting in different Erk_T concentrations. The data point belonging to the experiment in which no knockdown was performed serves as control experiment with $u = 1$ in normalized units.

Unfortunately these measurements are not acompanied by information about standard deviations in the publication. For the purposes of an illustrative example we suggest that the value $\sigma_{ij} = 0.2$ seems reasonable. The corresponding measurement model reads

$$\begin{aligned} h(\bar{x}, \rho, u_j) &= C\bar{x}(\rho, u_j)\,, \\ C = (0,1)\,, &\quad \sigma_{ij} = 0.2 \quad \forall\, i,j\,. \end{aligned} \tag{25}$$

We used a normal prior in the logarithmic parameter space which covers several orders of magnitude for both parameters. This example has two parameters, making it convenient to compare the kernel density estimate of the posteriors to an evaluation of the posterior on a regular grid, which removes any sampling related errors. This comparison can be seen in Figure 1. All posteriors look very similar, which indicates proper convergence for all three algorithms.

When looking at Additional file 1: Figure S1, we see that the effective sampling speed of NR-RMHMC is 2.6 times higher than that of the original RMHMC, while NR-HMC is better still.

Example: insulin dose response

A larger example is provided in [21], in which the authors analyze the insulin pathway in eukaryotic cells (primary human adipocytes). Different models were tested in their ability to describe phosphorylation of the insulin receptor substrate (IRS) after insulin stimulation. The data sets provided in their supplementary material consist of dose response curves, which we interpret as steady state data, as well as dynamic responses, which we disregard here.

From the set of models provided, we chose one of the least complex models (shorthand label: Mma) with $m = 6$ parameters, which was nevertheless sufficient to fit the steady state data, as well as the best fitting model (Mifa). The interaction graph of Mma is shown in Figure 2. The model comprises 5 molecular species: the insulin receptor (IR and IR*), phosphorylated IR (IRP), IR substrate (IRS), and phosphorylated IRS (IRSP). These reactants form two groups with built in conservation relations. Since the sums

$$\begin{aligned} \frac{d}{dt}\left([\mathrm{IR}] + [\mathrm{IRP}] + \left[\mathrm{IR}^*\right]\right) &= 0\,, \\ \frac{d}{dt}\left([\mathrm{IRS}] + [\mathrm{IRSP}]\right) &= 0\,, \end{aligned} \tag{26}$$

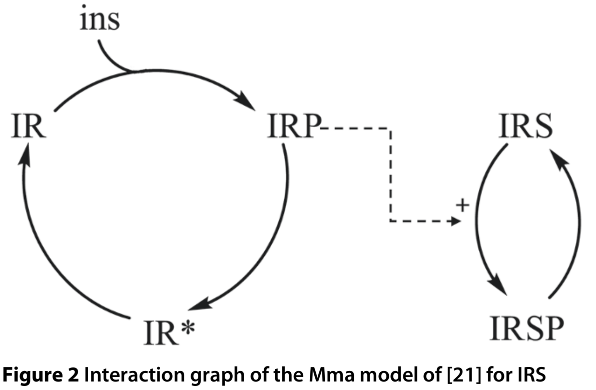

Figure 2 Interaction graph of the Mma model of [21] for IRS phosphorylation after insulin stimulation.

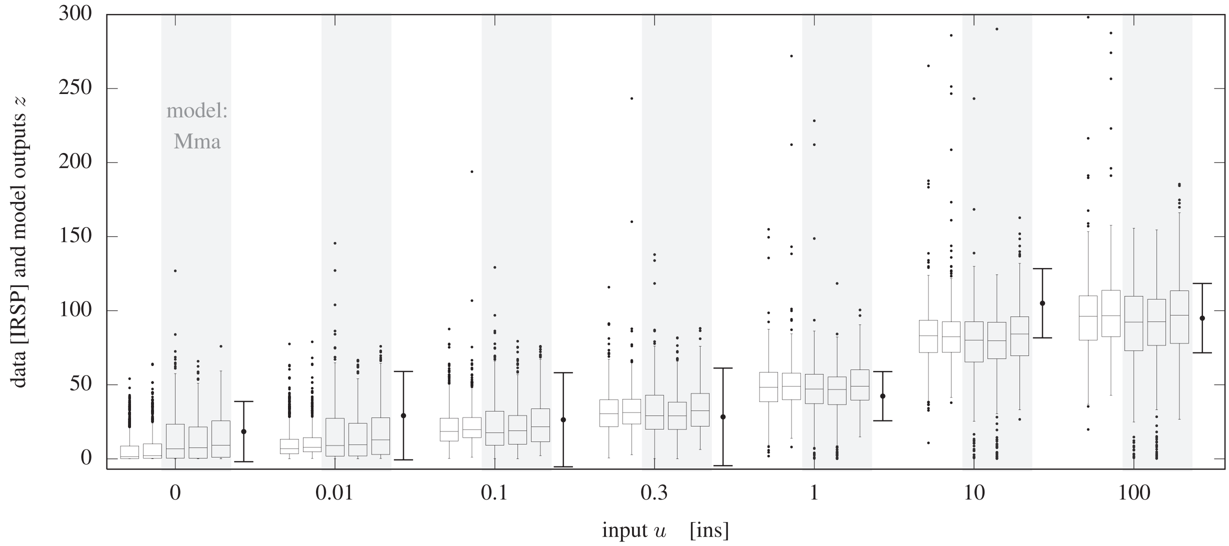

Figure 3 Comparison of five different fits for the insulin dose resonse data. The data is shown as error bars (last in each group), the fits are shown in boxplot style in the order: (i) Mifa model and NR-RMHMC sampler, (ii) Mifa model NR-HMC sampler, (iii) Mma model NR-RMHMC sampler, (iv) Mma model NR-HMC sampler, (v) Mma model and unchanged RMHMC sampler with numerical ODE solutions using SBToolBox2[mex]. All algorithms succeeded in fitting the data.

do not change over time, we only require $n = 3$ independent state variables to write down the ODE model:

$$x_1 = [\mathrm{IR}]\ , \quad x_2 = [\mathrm{IRP}]\ , \quad x_3 = [\mathrm{IRSP}]\ ,$$

$$u = [\mathrm{ins}] \quad \texttt{\% the insulin concentration,}$$

$$x_0 = \begin{pmatrix} 10 & 0 & 0 \end{pmatrix}^{\mathrm{T}}\ ,$$

which defines the initial value problem (ivp) for

$$\dot{x} = \begin{pmatrix} -1 & -1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} \mathrm{diag}(\rho)\phi(x)\,,$$

$$\phi(x) = \begin{pmatrix} x_1 u \\ x_1 \\ 10 - x_1 - x_2 \\ x_2 \\ x_2(10 - x_3) \\ x_3 \end{pmatrix},$$

with measurement model

$$h(\{\bar{x}_{\rho,u}\}, \rho, u_j) = C\bar{x}(\rho, u_j)\,, \qquad C = (0, 0, 98)\,. \qquad (27)$$

We have used the value for the output parameter $C_{1,3}$ reported in the publication and will not estimate it during sampling.

The larger Mifa model, which comprises 6 independent state variables and 14 parameters and is capable of fitting dynamic responses as well, is treated similarly and included as SB model in the supplement.

Figure 3 shows a comparison of fits using different models and samplers, illustrated as box plots. The boxes fit the observed experimental data well (error bars). It was not computationally feasible to sample the 14 parameter Mifa model on our desktop machine using the unmodified RMHMC algorithm.

As shown in Additional file 1: Figure S1, the effective speed of NR-RMHMC was about 23 times higher than that of standard RMHMC for the 3 × 6 Mma

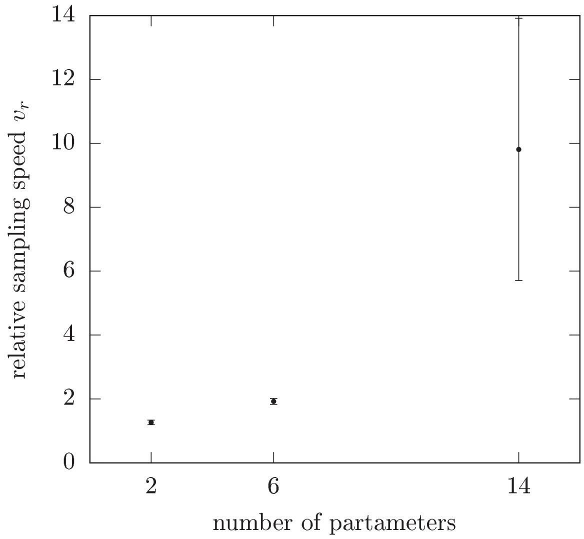

Figure 4 Relative speed for the Simplified Manifold Metropolis-Adjusted Langevin Algorithm (SMMALA) using the same modifications as in the RMHMC code. We calculate the sensitivities using systems of linear equations, we use the sensitivities to obtain the metric and the starting point for the Newton Raphson method which is used to calculate steady states. We tested the relative speed of both the modified and unmodified SMMALA on all three example models with uninformative priors. In all cases, the modified version of SMMALA was faster than the reference implementation.

model, indicating a significant speed up. The largest observed correlation coefficient of the posterior was $\varrho_{56}(\mathrm{Mma}) \approx 0.8$. Since the posterior does not exhibit strong enough correlations between the parameters, the flat (Euclidean) metric used in NR-HMC was of no disadvantage and hence NR-HMC performed best. With the larger Mifa model RMHMC came to its limits, while the modified NR-RMHMC was still able to generate samples in an acceptable time. The use of a Newton-Raphson approach for this model resulted in two orders of magnitude improvement in sampling performance for the NR-HMC algorithm. Although NR-HMC was superior in cases with uninformative priors, the advantage of NR-RMHMC becomes evident in the case of an informative prior. The informative prior was built by assigning smaller variances to some of the parameters θ_i while keeping the rest vague (large variances). See the setup files in the supplement for details.

Conclusion

We have demonstrated on three real world examples of different sizes how Hamiltonian Monte Carlo methods may be implemented in an efficient manner for the class of steady state dynamical systems. Using sensitivity analysis to track the steady states during each (RM)HMC trajectory calculation in the proposal step leads to a significant improvement in terms of effective sampling speed in all examples. Furthermore, the speed up was even more pronounced for larger problems comprising more parameters.

The proposed approach is also applicable to other Riemannian manifold MCMC algorithms like SMMALA. Figure 4 shows that there are significant albeit less dramatic benefits for the introduced techniques for this algorithm as well. Once again, we can use the comparatively inexpensive steady state sensitivity analysis to obtain the metric tensor G as well as the Newton-Raphson method for the calculation of steady states to great effect.

There remains the question of whether to employ NR-HMC or the more complicated NR-RMHMC algorithm for performing inference over steady state systems. In most cases, steady state data is not sufficient to uniquely fit the parameters to specific values, and we are unable to know a priori whether the parameters in the system will exhibit strong correlation structure. In practice, one might therefore start with the simpler NR-HMC scheme, and resort to the Riemannian version of it if the need arises. In cases where the prior may be well specified, the differences in scale between different parameters may require NR-RMHMC for improved efficiency, as we observed in the example section.

We conclude that the use of Newton-Raphson methods for calculating the steady states within HMC algorithms is a valuable contribution to the field of numerical parameter estimation for this special class of ODE models and improves scalability of statistical sampling-based approaches for these models in general. The success of this approach motivates further development of HMC algorithms for the more general case of dynamic time series data, which would broaden its utility.

Endnote

[a] We note that this is not always the case: whenever conservation relations are present, for example, the Jacaobian is not invertible anywhere. However, in such cases it is sufficient to use these conservation relations to reduce the number of state variables, as we do in the examples.

Additional file

Additional file 1: Figure S1. Performance analysis for the original RMHMC for ODE models and two steady state data adapted HMC algorithms.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

BC developed the RMHMC methodology and co-wrote the manuscript. AK implemented the changes for effective sensitivity calculations using the Newton-Raphson method and co-wrote the manuscript. NR co-wrote the manuscript. All authors read and approved the final manuscript.

Acknowledgements

We thank Jan Hasenauer for discussions and numerous suggestions for good example systems.

Author details

[1]Institute for Systems Theory and Automatic Control, Pfaffenwaldring 9, 70550 Stuttgart, Germany. [2]Department of Mathematics, Imperial College London, SW7 2AZ London, UK.

Received: 15 January 2014 Accepted: 7 July 2014
Published: 28 July 2014

References

1. Girolami M, Calderhead B: **Riemann manifold langevin and hamiltonian monte carlo methods.** *J R Stat Soc: Series B (Stat Methodol)* 2011, **73**(2):123–214. doi:10.1111/j.1467-9868.2010.00765.x.
2. Wilkinson DJ: *Stochastic Modelling for Systems Biology Mathematical and Computational Biology, vol. 11*. London, UK: Chapman & Hall/CRC; 2006.
3. Gelman A, Carlin JB, Stern HS, Rubin DB: *Bayesian Data Analysis, Texts in Statistical Science*. 2edn. London, UK: Chapman & Hall, CRC; 2004.
4. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E: **Equation of state calculations by fast computing machines.** *J Chem Phys* 1953, **21**(6):1087–1092. doi:10.1063/1.1699114.
5. Kaderali L, Dazert E, Zeuge U, Frese M, Bartenschlager R: **Reconstructing signaling pathways from RNAi data using probabilistic Boolean threshold networks.** *Bioinformatics* 2009, **25**(17):2229–2235.
6. Bois FY: **GNU MCSim: Bayesian statistical inference for SBML-coded systems biology models.** *Bioinformatics* 2009, **25**(11):1453–1454. doi:10.1093/bioinformatics/btp162.
7. Haario H, Laine M, Mira A, Saksman E: **DRAM: Efficient adaptive MCMC.** In *Statistics and Computing. Volume 16*. Switzerland: Springer; 2006:339–354.
8. Brooks S, Gelman A, Jones G. L, Meng X-L (Eds): *Handbook of Markov Chain Monte Carlo. Handbooks of Modern Statistical Methods*. London, UK: Chapman & Hall/CRC; 2011.

9. Lawrence N. D, Girolami M, Rattray M, Sanguinetti G (Eds): *Learning and Inference in Computational Systems Biology. Computational Molecular Biology*. Cambridge, Massachusetts, London, England: The MIT Press; 2010.
10. Rannala B: **Identifiability of parameters in MCMC Bayesian inference of phylogeny.** *Syst Biol* 2002, **51**(5):754–760.
11. Duane S, Kennedy AD, Pendleton BJ, Roweth D: **Hybrid Monte Carlo.** *Phys Lett B* 1987, **195**(2):216–222. doi:10.1016/0370-2693(87)91197-X.
12. Radde N: **The role of feedback mechanisms in biological network models - a tutorial.** *Asian J Control* 2011, **13**(5):597–610.
13. Calderhead B, Sustik M: **Sparse approximate manifolds for differential geometric mcmc.** *Adv Neural Inf Process Syst* 2012, **26:**2888–2896.
14. Rao CR: **Information and accuracy attainable in the estimation of statistical parameters.** *Bull Calc Math Soc* 1945, **1**(37).
15. Calderhead B: **Differential geometric mcmc methods and applications.** *PhD thesis.* University of Glasgow 2011.
16. Brooks S, Gelman A, Jones GL, Meng X-L (Eds): *Handbook of Markov Chain Monte Carlo Handbooks of Modern Statistical Methods, Chap. 5*. London, UK: Chapman & Hall/CRC; 2011.
17. Weckesser W: **Vfgen: A code generation tool.** *JNAIAM* 2008, **3**(1-2):151–165.
18. Wolff U: **Monte Carlo errors with less errors.** *Comput Phys Commun* 2004, **156**(2):143–153. doi:10.1016/S0010-4655(03)00467-3.
19. Schmidt H, Jirstrand M: **Systems biology toolbox for matlab: a computational platform for research in systems biology.** *Bioinformatics* 2006, **22**(4):514–515. doi:10.1093/bioinformatics/bti799. [http://bioinformatics.oxfordjournals.org/content/22/4/514.full.pdf+html]
20. Fritsche-Guenther R, Witzel F, Sieber A, Herr R, Schmidt N, Braun S, Brummer T, Sers C, Blüthgen N: **Strong negative feedback from Erk to Raf confers robustness to mapk signalling.** *Mol Syst Biol* 2011, **7**(489):1–13.
21. Brännmark C, Palmer R, Glad ST, Cedersund G, Stralfors P: **Mass and information feedbacks through receptor endocytosis govern insulin signaling as revealed using a parameter-free modeling framework.** *J Biol C* 2010, **285**(26):20171–9. doi:10.1074/jbc.M110.106849. [http://www.jbc.org/content/early/2010/04/26/jbc.M110.106849.full.pdf+html]

doi:10.1186/1471-2105-15-253
Cite this article as: Kramer *et al.*: Hamiltonian Monte Carlo methods for efficient parameter estimation in steady state dynamical systems. *BMC Bioinformatics* 2014 **15**:253.