frontiers in Bioengineering and Biotechnology

REVIEW
published: 20 November 2015
doi: 10.3389/fbioe.2015.00180

Gradient Matching Methods for Computational Inference in Mechanistic Models for Systems Biology: A Review and Comparative Analysis

Benn Macdonald * *and Dirk Husmeier* *

School of Mathematics and Statistics, University of Glasgow, Glasgow, UK

Parameter inference in mathematical models of biological pathways, expressed as coupled ordinary differential equations (ODEs), is a challenging problem in contemporary systems biology. Conventional methods involve repeatedly solving the ODEs by numerical integration, which is computationally onerous and does not scale up to complex systems. Aimed at reducing the computational costs, new concepts based on gradient matching have recently been proposed in the computational statistics and machine learning literature. In a preliminary smoothing step, the time series data are interpolated; then, in a second step, the parameters of the ODEs are optimized, so as to minimize some metric measuring the difference between the slopes of the tangents to the interpolants, and the time derivatives from the ODEs. In this way, the ODEs never have to be solved explicitly. This review provides a concise methodological overview of the current state-of-the-art methods for gradient matching in ODEs, followed by an empirical comparative evaluation based on a set of widely used and representative benchmark data.

Keywords: ordinary differential equations, gradient matching, Gaussian processes, reproducing kernel Hilbert space, parallel tempering, B-splines

OPEN ACCESS

Edited by:
Marcio Luis Acencio, Universidade Estadual Paulista, Brazil

Reviewed by:
Adriano Velasque Werhli, Universidade Federal do Rio Grande, Brazil
Paulo F. A. Mancera, Universidade Estadual Paulista, Brazil

***Correspondence:**
Benn Macdonald
b.macdonald.1@research.gla.ac.uk, b.macdonald.research@gmail.com;
Dirk Husmeier
dirk.husmeier@glasgow.ac.uk

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *15 June 2015*
Accepted: *23 October 2015*
Published: *20 November 2015*

Citation:
Macdonald B and Husmeier D (2015) Gradient Matching Methods for Computational Inference in Mechanistic Models for Systems Biology: A Review and Comparative Analysis. Front. Bioeng. Biotechnol. 3:180. doi: 10.3389/fbioe.2015.00180

1. INTRODUCTION

The elucidation of the structure and dynamics of biopathways is a central objective of systems biology. A standard approach is to view a biopathway as a network of biochemical reactions, which is modeled as a system of ordinary differential equations (ODEs). Following Barenco et al. (2006), this system can typically be expressed as[1]:

$$\frac{dx_i(t)}{dt} = g_i(\mathbf{x}(t), \boldsymbol{\rho}_i, t) - \delta_i x_i(t), \tag{1}$$

where $i \in \{1, \ldots, n\}$ denotes one of n components (henceforth referred to as "species") in the biopathway, $x_i(t)$ denotes the concentration of species i at time t, δ_i is a decay rate and $\mathbf{x}(t)$ is a vector of concentrations of all system components that influence or regulate the concentration of species i at time t. If, for instance, species i is an mRNA, then $\mathbf{x}(t)$ may contain the concentrations of transcription factors (proteins) that bind to the promoter of the gene from which i is transcribed.

[1]We do not make the baseline transcription rate explicit in our notation, but include it in the function $g_i(.)$.

The regulation is modeled by the regulation function g. Depending on the species involved, g may define different types of regulatory interactions, e.g., mass action kinetics, Michaelis–Menten kinetics, allosteric Hill kinetics, etc. All of these interactions depend on a vector of kinetic parameters, $\boldsymbol{\rho}_i$. For complex biopathways, only a small fraction of $\boldsymbol{\rho}_i$ can typically be measured. Hence, the explication of the biopathway dynamics requires the majority of kinetic parameters to be inferred from observed (typically noisy and sparse) time course concentration profiles. In principle, this can be accomplished with standard techniques from machine learning and statistical inference. These techniques are based on first quantifying the difference between predicted and measured time course profiles by some appropriate metric to obtain the likelihood of the data. The parameters are then either optimized to maximize the likelihood (or a regularized version thereof), or sampled from a distribution based on the likelihood (the posterior distribution).

However, the nature of the ODE-based model in equation (1) renders the inference problem computationally challenging in two respects. First, the ODE system often does not permit closed-form solutions. One therefore has to resort to numerical simulations every time the kinetic parameters $\boldsymbol{\rho}_i$ are adapted, which is computationally onerous. Second, the likelihood function in the space of parameters $\boldsymbol{\rho}_i$ is typically not unimodal, but suffers from multiple local optima. Hence, even if a closed-form solution of the ODEs existed, inference by maximum likelihood would face an NP-hard optimization problem, and Bayesian inference would suffer from poor mixing and convergence of the Markov chain Monte Carlo (MCMC) simulations.

Conventional inference methods involve numerically integrating the system of ODEs to produce a signal, which is compared to the data by some appropriate metric defined by the chosen noise model, allowing for the calculation of a likelihood. This process is repeated as part of an iterative optimization or sampling procedure to produce estimates of the parameters. **Figure 1A** is a graphical representation of the model for these conventional inference methods. For a given set of initial concentrations of the entire system $\mathbf{X}(0)$ and set of ODE parameters $\boldsymbol{\theta}$ [where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$ and $\boldsymbol{\theta}_i = (\boldsymbol{\rho}_i, \delta_i)$], a signal can be produced by integration of the ODEs. As mentioned previously, for many ODE systems a closed-form solution does not exist, so in practice, numerical integration is implemented instead. Assuming an appropriate noise model (for example, a Gaussian additive noise model) with standard deviation (SD) of the observational error $\boldsymbol{\sigma}$, the differences between the resultant signal and the data $\mathbf{Y}$ can be used to calculate the likelihood of the parameters $\boldsymbol{\theta}$. The process is repeated for different parameters $\boldsymbol{\theta}$ until the maximum likelihood of the parameters is found (in the classical approach) or until convergence to the posterior distribution is reached (in the Bayesian approach). However, the computational costs involved with repeatedly numerically solving the ODEs are large.

To reduce the computational complexity, several authors have adopted an approach based on gradient matching [e.g., Calderhead et al. (2008) and Liang and Wu (2008)]. The idea is based on the following two-step procedure. In a preliminary smoothing step, the time series data are interpolated; then, in a second step, the parameters $\boldsymbol{\theta}$ of the ODEs are optimized so as to minimize some metric measuring the difference between the slopes of the

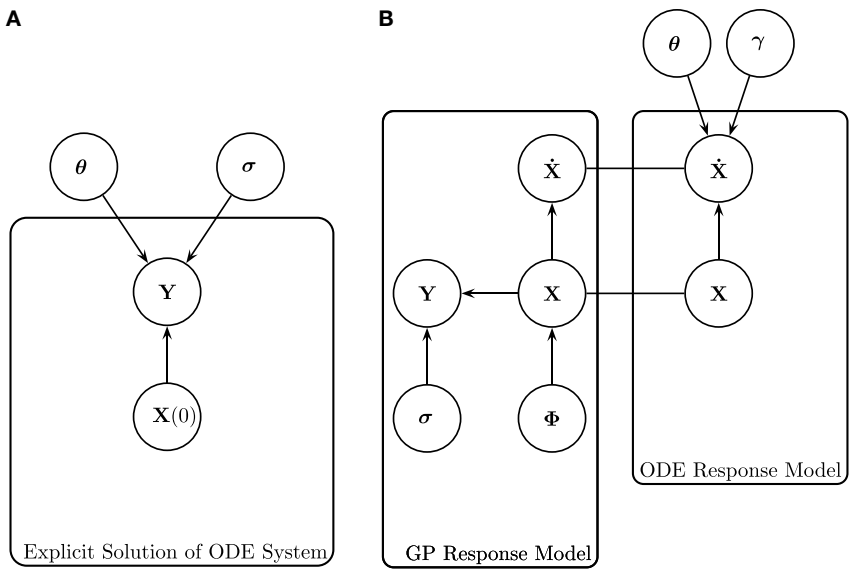

FIGURE 1 | Graphical representations of (left) the explicit solution of the ODE system, as shown in Calderhead et al. (2008), and (right) gradient matching with Gaussian processes, as proposed in Calderhead et al. (2008) and Dondelinger et al. (2013). **(A)** Explicit solution of the ODE system, as shown in Calderhead et al. (2008). The noisy data signals $\mathbf{Y}$ are described by some initial concentration $\mathbf{X}(0)$, ODE parameters $\boldsymbol{\theta}$ and observational errors with SD $\boldsymbol{\sigma}$. For a given set of initial concentrations $\mathbf{X}(0)$ and set of ODE parameters $\boldsymbol{\theta}$, the ODEs can be integrated to produce a signal, which is then compared to the data signal by some metric defined by the chosen noise model. **(B)** Gradient matching with Gaussian processes, as proposed in Calderhead et al. (2008) and Dondelinger et al. (2013). The gradients $\dot{\mathbf{X}}$ are compared from two modeling approaches; the Gaussian process model and the ODEs themselves. The distribution of $\mathbf{Y}$ is given in equation (4), the Gaussian process on $\mathbf{X}$ defined in equation (5), the derivatives of the Gaussian process $\dot{\mathbf{X}}$ in equation (10), the ODE model in equation (2), and the gradient matching in equation (17). All symbols are detailed in Section 2.1.

tangents to the interpolants, and the $\boldsymbol{\theta}$-dependent time derivatives from the ODEs. In this way, the ODEs never have to be solved explicitly, and the typically unknown initial conditions are effectively profiled over. A disadvantage of this two-step scheme is that the results of parameter inference critically hinge on the quality of the initial interpolant. A better approach, first suggested in Ramsay et al. (2007), is to regularize the interpolants by the ODEs themselves. Dondelinger et al. (2013) applied this idea to the non-parametric Bayesian approach of Calderhead et al. (2008), using Gaussian processes (GPs), and demonstrated that it substantially improves the accuracy of parameter inference and robustness with respect to noise. As opposed to Ramsay et al. (2007), all smoothness hyperparameters are consistently inferred in the framework of non-parametric Bayesian statistics, dispensing with the need to adopt heuristics and approximations.

This review compares the current state-of-the-art in gradient matching, specifically in the context of parameter inference in ODEs. This comparison aids in understanding the difference between key components of methods without confounding influence from other modeling choices. For instance, we compare the inference paradigm of the parameter that governs the degree of mismatch between the gradients of the interpolants and ODEs [using the method in Dondelinger et al. (2013)] with a tempering approach [from the method in Macdonald and Husmeier (2015)], using the same interpolation scheme (namely, Gaussian processes). This way, we are able to gain an understanding as to what approach may be more suitable, without concern that differences may be due to interpolation choice. If the ODEs provide the correct mathematical description of the system, ideally there should be no difference between the interpolant gradients and those predicted from the ODEs. In practice, however, forcing the gradients to be equal is likely to cause parameter inference techniques to converge to a local optimum of the likelihood. A parallel tempering scheme is the natural way to deal with such local optima, as opposed to inferring the degree of mismatch, since different tempering levels correspond to different strengths of penalizing the mismatch between the gradients. A parallel tempering scheme (which uses smoothed versions of the posterior distribution as well as the usual posterior distribution, see Section 2.2 for more details) was explored by Campbell and Steele (2012).

When comparing one method to another, in order to assess the strengths and weaknesses of an approach, often results are not directly comparable, since different approaches use different methodological paradigms. For example, if the method by Campbell and Steele (2012) (which uses B-splines interpolation) was compared to Dondelinger et al. (2013) (which uses a GP approach) in order to examine the difference between parallel tempering and inference of the parameter controlling the degree of mismatch between the gradients, then the results would be confounded by the choice of interpolation scheme. In this review, we present a comparative evaluation of parallel tempering versus inference in the context of gradient matching for the same modeling framework, i.e., without any confounding influence from the model choice. We also compare the method of Bayesian inference with Gaussian processes with other methodological paradigms, within the specific context of adaptive gradient matching, which is highly relevant to current computational systems biology. We look at the methods of: Campbell and Steele (2012), who carry out parameter inference using adaptive gradient matching and B-splines interpolation; González et al. (2013), who implement a reproducing kernel Hilbert space (RKHS) and penalized maximum likelihood approach in a non-Bayesian fashion; Ramsay et al. (2007), who optimize the gradient mismatch, interpolant, and ODE parameters using a hierarchical regularization method and penalize the difference between the gradients using B-splines in a non-Bayesian approach; Dondelinger et al. (2013), who use adaptive gradient matching with Gaussian processes, inferring the degree of mismatch between the gradients; and Macdonald and Husmeier (2015), who use adaptive gradient matching with Gaussian processes and temper the parameter that controls the degree of mismatch between the gradients.

2. METHODOLOGY

2.1. Adaptive Gradient Matching with Gaussian Processes

The following covers the background of methodology for Dondelinger et al. (2013), and Macdonald and Husmeier (2015), which combines the former method with a parallel tempering scheme for the gradient mismatch parameter (the details on parallel tempering will be given in Section 2.2).

Consider a set of T arbitrary timepoints $t_1 < \ldots < t_T$, and a set of noisy observations $\mathbf{Y} = (\mathbf{y}(t_1), \ldots, \mathbf{y}(t_T))$, where $\mathbf{y}(t) = \mathbf{x}(t) + \boldsymbol{\epsilon}(t)$, $\mathrm{n} = \dim(\mathbf{x}(t))$, $\mathbf{X} = (\mathbf{x}(t_1), \ldots, \mathbf{x}(t_T))$, $\mathbf{y}(t)$ is the data vector of the observations of all species concentrations at time t, $\mathbf{x}(t)$ is the vector of the concentrations of all species at time t, $\mathbf{y}_i$ is the data vector of the observations of species concentrations i at all timepoints, $\mathbf{x}_i$ is the vector of concentrations of species i at all timepoints, $y_i(t)$ is the observed datapoint of the concentration of species i at time t, $x_i(t)$ is the concentration of species i at time t and $\boldsymbol{\epsilon}$ is multivariate Gaussian noise, $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma_i^2 \mathbf{I})$. The signals of the system are described by ordinary differential equations

$$\dot{\mathbf{x}}_i = \frac{d\mathbf{x}_i}{dt} = f_i(\mathbf{X}, \boldsymbol{\theta}_i, \mathbf{t}), \tag{2}$$

or alternatively, represented in scalar form

$$\dot{x}_i(t) = \frac{dx_i(t)}{dt} = f_i(\mathbf{x}(t), \boldsymbol{\theta}_i, t), \tag{3}$$

where $\dot{\mathbf{x}}_i$ is the vector containing the ODE gradients for species i at all timepoints, $f_i\ (\mathbf{t}) = (f_i(t_1), \ldots, f_i(t_T))^{\mathrm{T}}$, $\boldsymbol{\theta}_i = (\boldsymbol{\rho}_i,\ \delta_i)$, $\boldsymbol{\rho}_i$ is a vector of kinetic parameters, δ_i is a decay rate parameter and $f_i\ (\mathbf{x}(t), \boldsymbol{\theta}_i, t) = g_i\ (\mathbf{x}(t), \boldsymbol{\rho}_i, t) - \delta_i\ x_i$. Then,

$$p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma}^2) = \prod_i \prod_t N(y_i(t)|x_i(t), \sigma_i^2), \tag{4}$$

and the matrices $\mathbf{X}$ and $\mathbf{Y}$ are of dimension n by T. Following Calderhead et al. (2008), we place a Gaussian process (GP) prior on $\mathbf{x}_i$,

$$p(\mathbf{x}_i|\boldsymbol{\mu}_i, \boldsymbol{\phi}_i) = N(\mathbf{x}_i|\boldsymbol{\mu}_i, \mathbf{C}_{\phi_i}), \tag{5}$$

where $\boldsymbol{\mu}_i$ is a mean vector, for simplicity set as the sample mean, and $\mathbf{C}_{\phi_i}$ is a positive definite matrix of covariance functions with hyperparameters ϕ_i. Since differentiation is a linear operation, a Gaussian process is closed under differentiation, and the joint distribution of the state variables $\mathbf{x}_i$ and their time derivatives $\dot{\mathbf{x}}_i$ is multivariate Gaussian with mean vector $(\boldsymbol{\mu}_i, \mathbf{0})^{\mathsf{T}}$ and covariance functions

$$cov[x_i(t), x_i(t')] = C_{\phi_i}(t, t'), \tag{6}$$

$$cov[\dot{x}_i(t), x_i(t')] = \frac{\partial C_{\phi_i}(t, t')}{\partial t} := C'_{\phi_i}(t, t'), \tag{7}$$

$$cov[x_i(t), \dot{x}_i(t')] = \frac{\partial C_{\phi_i}(t, t')}{\partial t'} := {}'C_{\phi_i}(t, t'), \tag{8}$$

$$cov[\dot{x}_i(t), \dot{x}_i(t')] = \frac{\partial^2 C_{\phi_i}(t, t')}{\partial t \partial t'} := C''_{\phi_i}(t, t'), \tag{9}$$

where $C_{\phi_i}(t, t')$ are the elements of the covariance matrix $\mathbf{C}_{\phi_i}$. Using elementary transformations of Gaussian distributions [for example, see page 87 of Bishop (2006)], the conditional distribution for the state derivatives is then

$$p(\dot{\mathbf{x}}_i | \mathbf{x}_i, \boldsymbol{\mu}_i, \boldsymbol{\phi}_i) = N(\mathbf{m}_i, \mathbf{K}_i), \tag{10}$$

where

$$\mathbf{m}_i = {}'\mathbf{C}_{\phi_i}{\mathbf{C}_{\phi_i}}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_i) \text{ and } \mathbf{K}_i = \mathbf{C}''_{\phi_i} - {}'\mathbf{C}_{\phi_i}{\mathbf{C}_{\phi_i}}^{-1}\mathbf{C}'_{\phi_i}. \tag{11}$$

Assuming additive Gaussian noise with a state-specific error variance γ_i, from equation (2) we get

$$p(\dot{\mathbf{x}}_i | \mathbf{X}, \boldsymbol{\theta}_i, \gamma_i) = N(f_i(\mathbf{X}, \boldsymbol{\theta}_i, \mathbf{t}), \gamma_i \mathbf{I}). \tag{12}$$

Calderhead et al. (2008), and Dondelinger et al. (2013) link the interpolant in equation (10) with the ODE model in equation (12) using a product of experts approach, as illustrated in **Figure 1B**, obtaining the following distribution

$$\begin{aligned} p(\dot{\mathbf{x}}_i | \mathbf{X}, \boldsymbol{\theta}_i, \boldsymbol{\mu}_i, \boldsymbol{\phi}_i, \gamma_i) &\propto p(\dot{\mathbf{x}}_i | \mathbf{x}_i, \boldsymbol{\mu}_i, \boldsymbol{\phi}_i) p(\dot{\mathbf{x}}_i | \mathbf{X}, \boldsymbol{\theta}_i, \gamma_i) \\ &= N(\mathbf{m}_i, \mathbf{K}_i) N(f_i(\mathbf{X}, \boldsymbol{\theta}_i, \mathbf{t}), \gamma_i \mathbf{I}). \end{aligned} \tag{13}$$

The joint distribution is therefore

$$\begin{aligned} &p(\dot{\mathbf{X}}, \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\phi}, \boldsymbol{\gamma}) \\ &\quad = p(\boldsymbol{\theta}) p(\boldsymbol{\phi}) p(\boldsymbol{\gamma}) \prod_i p(\dot{\mathbf{x}}_i | \mathbf{X}, \boldsymbol{\theta}_i, \boldsymbol{\mu}_i, \boldsymbol{\phi}_i, \gamma_i) p(\mathbf{x}_i | \boldsymbol{\phi}_i), \end{aligned} \tag{14}$$

where $\boldsymbol{\gamma}$ is the vector containing all the gradient mismatch parameters and $\mathrm{p}(\boldsymbol{\theta}), p(\boldsymbol{\phi}), p(\boldsymbol{\gamma})$ are the priors over the respective parameters. Dondelinger et al. (2013) show that you can marginalize over the derivatives to get a closed-form solution to

$$p(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\phi}, \boldsymbol{\gamma}) = \int p(\dot{\mathbf{X}}, \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\phi}, \boldsymbol{\gamma}) d\dot{\mathbf{X}}. \tag{15}$$

Using equations (4) and (15), our full joint distribution becomes

$$\begin{aligned} &p(\mathbf{Y}, \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\phi}, \boldsymbol{\gamma}, \boldsymbol{\sigma}^2) \\ &\quad = p(\mathbf{Y} | \mathbf{X}, \boldsymbol{\sigma}^2) p(\mathbf{X} | \boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\phi}, \boldsymbol{\gamma}) p(\boldsymbol{\theta}) p(\boldsymbol{\phi}) p(\boldsymbol{\gamma}) p(\boldsymbol{\sigma}^2), \end{aligned} \tag{16}$$

where the likelihood $p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma})$ is defined in equation (4) and $p(\boldsymbol{\sigma}^2)$ is the prior over the variances of the observational error. Dondelinger et al. (2013) show

$$\begin{aligned} &p(\mathbf{X} | \boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\phi}, \boldsymbol{\gamma}) \\ &\propto \frac{1}{Z} \exp\left[-\frac{1}{2} \sum_i \left(\mathbf{x}_i^{\mathsf{T}} \mathbf{C}_{\phi_i} \mathbf{x}_i + (\mathbf{f}_i - \mathbf{m}_i)^{\mathsf{T}} (\mathbf{K}_i + \gamma_i \mathbf{I})^{-1} (\mathbf{f}_i - \mathbf{m}_i) \right) \right], \end{aligned} \tag{17}$$

where $Z = \prod_i |2\pi(\mathbf{K}_i + \gamma_i \mathbf{I})|^{\frac{1}{2}}$ and $\mathbf{f}_i$ is the vector containing the gradients from the ODEs for species *i*. The sampling is conducted using MCMC, where the whitening approach of Murray and Adams (2010) is used to efficiently sample in the joint space of GP hyperparameters $\boldsymbol{\phi}$ and latent variables $\mathbf{X}$. The concept of gradient matching with Gaussian processes can be seen graphically in **Figure 1B**. The data $\mathbf{Y}$ are explained by the latent variables $\mathbf{X}$, which are modeled by a Gaussian process with hyperparameters $\boldsymbol{\phi}$, and SD of the observational errors $\boldsymbol{\sigma}$. The gradients from the ODE model are compared to those from the Gaussian process, subject to some degree of mismatch controlled by parameter $\boldsymbol{\gamma}$, dispensing with the need to explicitly solve the ODEs.

2.2. Parallel Tempering

A challenging problem, which sampling methods face, is that of local optima. The aim of sampling is to represent fully the configuration space weighted by the volume of the corresponding posterior density peaks. In order to do this, the sampling algorithm implemented must be able to adequately explore the posterior distribution. If this landscape is rugged, with many local optima and low-probability barriers separating areas of high posterior probability, mixing and convergence of the Markov chain Monte Carlo simulations can be poor. For example, consider the Metropolis–Hastings algorithm, which proposes a move and computes the acceptance probability p_{move} by taking the ratio of the posterior densities of the proposed state to the current state. If $p_{\text{move}} > 1$, the algorithm accepts the proposed move. If $p_{\text{move}} < 1$, the proposed state is accepted with probability p_{move}. If then, the parameter location of the algorithm is currently situated at a local optimum, then the proposed move could result in a small p_{move}. Theoretically, the algorithm will eventually be able to move the parameter location out of this region; however, in practice, this could take a considerable amount of time. If the total number of MCMC iterations has been specified in advance, the simulation could finish before the parameter position of the algorithm has escaped the local optimum and explored the remainder of the region. Entrapment in local optima can mislead established convergence tests and erroneously indicate a sufficient degree of convergence.

Parallel tempering is a method that tackles the problem of local optima. It involves running multiple MCMC simulations at different levels or "temperatures"[2] of the likelihood in parallel. Low "temperatures" flatten the posterior landscape, making it easier to explore the region, since the peaks have been smoothed. This can be seen graphically in **Figure 2**. As the "temperature"

[2]By "temperature", we mean a tempering parameter that defines the degree of flattening of the likelihood. Formally, our "temperature" is equivalent to an inverse temperature in Statistical Physics.

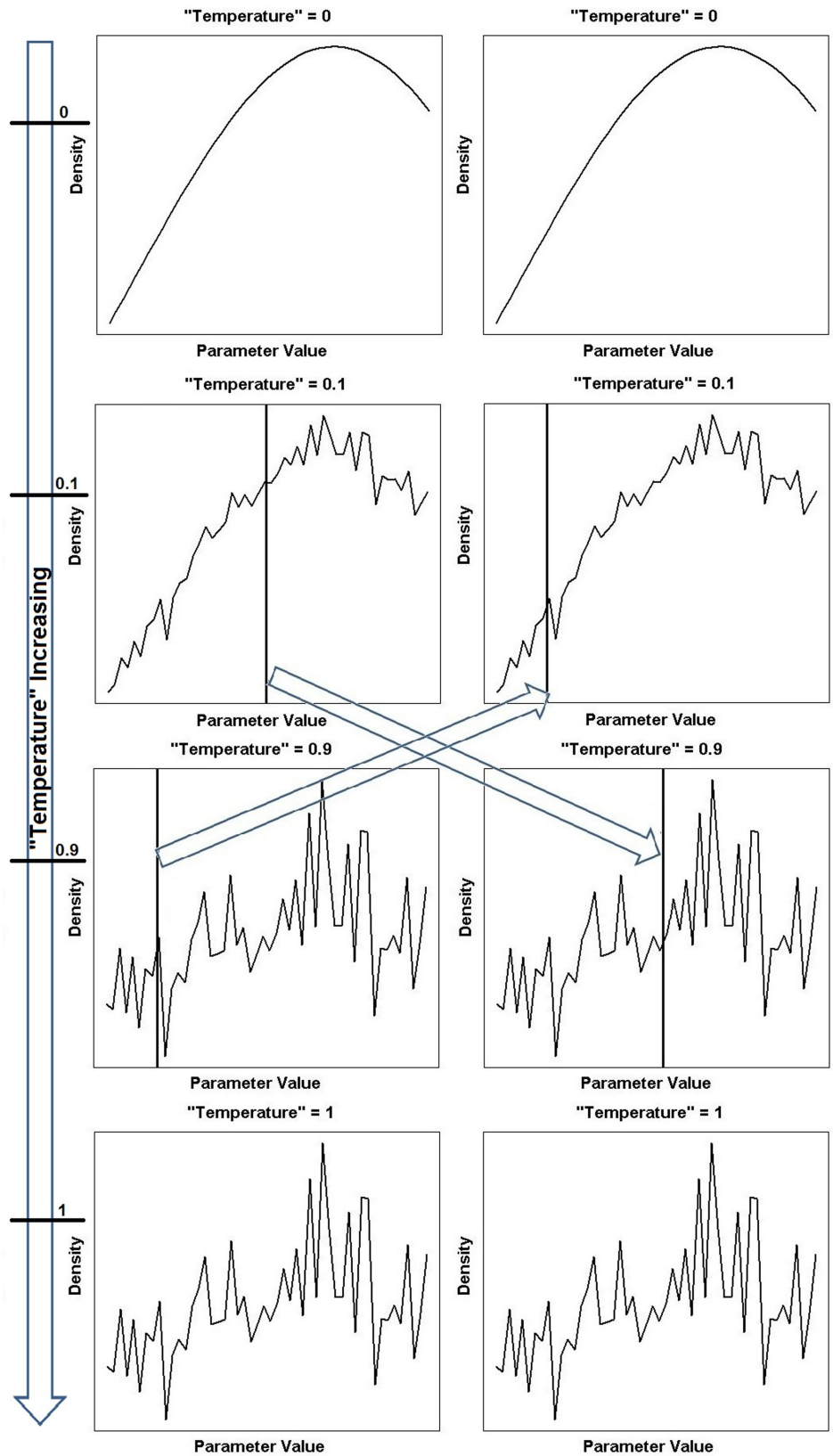

FIGURE 2 | Continued

FIGURE 2 | Continued
A one-dimensional illustration of equation (18) showing different power posterior distributions for different levels or "temperatures" of the likelihood. The posterior landscape is smoother at lower "temperatures" (corresponding to chains closer to the prior) and becomes increasingly rugged until the true posterior landscape is recovered for "temperature" = 1. The arrow on the far left depicts the increase in "temperature" and the horizontal ticks mark the specific "temperature" of that chain. Two chains ("temperature" = 0.1 and "temperature" = 0.9) have been chosen to swap parameter locations (locations indicated by vertical line). The left column shows the parameter locations of the tempering algorithm before the swap and the right column shows the parameter locations of the tempering algorithm after the swap. The swapping of locations is indicated by the arrows in the center of the figure.

is increased to the highest value, the landscape becomes more rugged and eventually the original posterior landscape is recovered (see bottom of **Figure 2**).

At every MCMC iteration, two "temperature" chains are chosen and the parameter locations where the sampling algorithm is currently situated are swapped, see middle of **Figure 2**. This way, the algorithm can move the parameter position from a local optimum to somewhere else on the posterior landscape, dispensing with the need to gradually navigate away from the region and the problems associated with doing so.

Consider a series of "temperatures", $0 = \beta^{(1)} < \ldots < \beta^{(M)} = 1$ and a power posterior distribution of our ODE parameters [Friel and Pettitt (2008)]

$$p_{\beta^{(j)}}(\boldsymbol{\theta}^{(j)}|\mathbf{y}) \propto p(\boldsymbol{\theta}^{(j)}) p(\mathbf{y}|\boldsymbol{\theta}^{(j)})^{\beta^{(j)}}. \tag{18}$$

Equation (18) reduces to the prior for $\beta^{(j)} = 0$ (see top of **Figure 2**), and becomes the posterior when $\beta^{(j)} = 1$ (see bottom of **Figure 2**), with $0 < \beta^{(j)} < 1$ creating a distribution between our prior and posterior (see **Figure 2**). The M $\beta^{(j)}$ annealed likelihoods in equation (18) are used as the target densities of M parallel MCMC chains [Campbell and Steele (2012)]. At each MCMC step, each "temperature" chain independently performs a Metropolis–Hastings step to update $\boldsymbol{\theta}^{(j)}$, the parameter vector associated with temperature $\beta^{(j)}$

$$p_{\text{move}} = \min \left(1, \frac{\begin{array}{c} p\left(\mathbf{y}|\boldsymbol{\theta}^{\text{proposed}(j)}\right)^{\beta^{(j)}} p\left(\boldsymbol{\theta}^{\text{proposed}(j)}\right) \\ \times q\left(\boldsymbol{\theta}^{\text{current(j)}}|\boldsymbol{\theta}^{\text{proposed(j)}}\right) \end{array}}{\begin{array}{c} p\left(\mathbf{y}|\boldsymbol{\theta}^{\text{current}(j)}\right)^{\beta^{(j)}} p\left(\boldsymbol{\theta}^{\text{current}(j)}\right) \\ \times q\left(\boldsymbol{\theta}^{\text{proposed(j)}}|\boldsymbol{\theta}^{\text{current(j)}}\right) \end{array}} \right), \tag{19}$$

where $q(\,)$ is the proposal distribution and the superscripts "proposed" and "current" indicate whether the algorithm is being evaluated at the proposed or current state. Also, at each MCMC step, two chains are randomly selected, and a proposal to exchange parameters is made, with acceptance probability

$$p_{\text{swap}} = \min \left(1, \frac{p_{\beta^{(k)}}(\boldsymbol{\theta}^{(j)}|\mathbf{y}) p_{\beta^{(j)}}(\boldsymbol{\theta}^{(k)}|\mathbf{y})}{p_{\beta^{(j)}}(\boldsymbol{\theta}^{(j)}|\mathbf{y}) p_{\beta^{(k)}}(\boldsymbol{\theta}^{(k)}|\mathbf{y})} \right). \tag{20}$$

A graphical representation of the swap moves between chains can be seen in **Figure 2**.

The method by Macdonald and Husmeier (2015) focuses on the intrinsic slack parameter γ_i [see equation (12)], which theoretically should be $\gamma_i = 0$, since this corresponds to no mismatch between the gradients. In practice, it is allowed to take on larger values, $\gamma_i > 0$, to prevent the inference scheme from getting stuck in sub-optimal states. However, rather than inferring γ_i like a model parameter, as carried out in Dondelinger et al. (2013), other authors [e.g., Campbell and Steele (2012)] propose that γ_i should be gradually set to zero, since values closer to zero force the gradients to be more similar and tie the interpolants closer to the ODEs. It is possible to abruptly set the values to zero, rather than gradually; however, this is likely to cause the parameter inference techniques to converge to a local optimum of the likelihood. To this end, Macdonald and Husmeier (2015) combine the gradient matching with Gaussian processes approach in Dondelinger et al. (2013) with the tempering approach in Campbell and Steele (2012) and temper this parameter to zero.

We choose values of γ_i and assign them to the variance parameter in equation (12) for each "temperature" $\beta^{(j)}$, such that chains closer to the prior ($\beta^{(j)}$ closer to 0) allow the gradients from the interpolant to have more freedom to deviate from those predicted by the ODEs (which corresponds to a larger γ_i), chains closer to the posterior ($\beta^{(j)}$ closer to 1) more closely match the gradients (corresponding to a smaller γ_i), and for the chain corresponding to $\beta^{(M)} = 1$, we wish that the mismatch is approximately zero ($\gamma_i \approx 0$). Since γ_i corresponds to the variance of our state-specific error [see equation (12)], as $\gamma_i \to 0$, we have less mismatch between the gradients, and as γ_i gets larger, the gradients have more freedom to deviate from one another. Hence, we temper γ_i toward zero. Now, each $\beta^{(j)}$ chain in equation (18) has a $\gamma_i^{(j)}$ [where the superscript (j) indicates the gradient mismatch parameter associated with "temperature" $\beta^{(j)}$] fixed in place for the strength of the gradient mismatch.

Continuing the notation, anything with a superscript (j) is the associated variable or fixed parameter for "temperature" chain $\beta^{(j)}$. The ODE model in equation (12) now becomes

$$p(\dot{\mathbf{x}}_i^{(j)}|\mathbf{X}^{(j)}, \boldsymbol{\theta}_i^{(j)}, \gamma_i^{(j)}) = N(f_i^{(j)}(\mathbf{X}^{(j)}, \boldsymbol{\theta}_i^{(j)}, \mathbf{t}), \gamma_i^{(j)}\mathbf{I}), \tag{21}$$

where this distribution is evaluated at each of the j chains. Following equations (13)–(16), we obtain for the joint distribution

$$\begin{aligned} p(\mathbf{Y}, \mathbf{X}^{(j)}, \boldsymbol{\theta}^{(j)}, \boldsymbol{\mu}, \boldsymbol{\phi}^{(j)}, \boldsymbol{\gamma}^{(j)}, \boldsymbol{\sigma}^{2(j)}) &= p(\mathbf{Y}|\mathbf{X}^{(j)}, \boldsymbol{\sigma}^{2(j)})^{\beta^{(j)}} \\ &\times p(\mathbf{X}^{(j)}|\boldsymbol{\theta}^{(j)}, \boldsymbol{\mu}, \boldsymbol{\phi}^{(j)}, \boldsymbol{\gamma}^{(j)}) p(\boldsymbol{\theta}^{(j)}) p(\boldsymbol{\phi}^{(j)}) p(\boldsymbol{\sigma}^{2(j)}). \end{aligned} \tag{22}$$

Equation (22) is calculated for each of the j chains. The particular schedules used for γ_i in this review are given in **Table 1**. For more details on tempering, see Calderhead and Girolami (2009) and Mohamed et al. (2012). The computational times for the methods from Dondelinger et al. (2013) and Macdonald and Husmeier (2015), in comparison to numerically integrating the ODEs, can be found in **Table 2**.

TABLE 1 | Ranges of the penalty parameter γ_i for LB2 and LB10.

Method	Chains	Range of penalty γ	Method	Chains	Range of penalty γ
LB2	4	[1, 0.125]	LB10	4	[1, 0.001]
LB2	10	[1, 0.00195]	LB10	10	$[1, 1 \times 10^{-9}]$

In this review, $\boldsymbol{\gamma}_i = \boldsymbol{\gamma} \forall i$.

TABLE 2 | Computational times for INF and a method that numerically integrates the ODEs for the protein signaling transduction pathway in equations (63)–(67).

INF		Numerical integration	
Exectution time of 1×10^5 MCMC steps (seconds)		Execution time of 1×10^5 MCMC steps (seconds)	
Median	**Interquartile range**	**Median**	**Interquartile range**
2500	[2400, 2600]	12,500	[12,000, 13,000]
Number of steps until convergence		**Number of steps until convergence**	
Median	**Interquartile range**	**Median**	**Interquartile range**
3.5×10^4	$[3.25 \times 10^4, 4.5 \times 10^4]$	7.9×10^4	$[7.5 \times 10^4, 8.25 \times 10^4]$

Table constructed from the boxplots in Dondelinger et al. (2013). The LB2 and LB10 methods were equivalent to INF in terms of computational time.

2.3. B-Splines

Splines are used for function interpolation, where the function of interest is approximated by a weighted linear combination of basis functions. These basis functions, called "splines", are "local" polynomials, where the exact functional form depends on the particular type of spline that is used (for example, a truncated power basis). See Hastie et al. (2009) for an overview of different types of splines.

The advantage of spline interpolation over global polynomial interpolation is that the interpolation error can be made small even when using low degree polynomials for the splines. This, in particular, avoids the problem of Runge's phenomenon, in which oscillations can occur between data points when interpolating using high degree polynomials.

B-splines interpolation takes the form

$$x(t) = \sum_{i=0}^{m} \alpha_i \phi_{i,d}(t), \tag{23}$$

where $m+1$ is the number of basis functions, d is the degree of polynomial, α_i is a coefficient and $\phi_{i,d}$ (t) is the i^{th} basis function of polynomial degree d evaluated at time t. For some vector of fixed points called knots [denoted $\boldsymbol{\tau}$, where $x(t)$ is continuous at each knot], the basis functions are calculated with the following recursive formulae

$$\phi_{i,0}(t) = \begin{cases} 1 & \text{if} \quad \tau_i \leq t < \tau_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

$$\phi_{i,d}(t) = \frac{t - \tau_i}{\tau_{i+d} - \tau_i} \phi_{i,d-1}(t) + \frac{\tau_{i+d+1} - t}{\tau_{i+d+1} - \tau_{i+1}} \phi_{i+1,d-1}(t). \tag{25}$$

The coefficients α_i are then estimated by

$$\hat{\boldsymbol{\alpha}} = \left(\boldsymbol{\Phi}^{\mathsf{T}} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^{\mathsf{T}} \mathbf{y}, \tag{26}$$

where $\hat{\boldsymbol{\alpha}}$ is the vector containing all the coefficients (and α_i would correspond to the $(i+1)^{\text{th}}$ position in the vector) and $\boldsymbol{\Phi}$ is the matrix containing all the basis functions

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_{0,d}(t_1) & \cdots & \phi_{m,d}(t_1) \\ \vdots & \ddots & \vdots \\ \phi_{0,d}(t_T) & \cdots & \phi_{m,d}(t_T) \end{bmatrix}. \tag{27}$$

One can aim to avoid over-fitting by penalizing the 2nd derivative of the function $x(t)$ (known as penalized splines), making our objective function

$$J(x) = \sum_{s=1}^{N} \left(y(t_s) - x(t_s)\right)^2 + \lambda \int \left(\frac{d^2 x}{dt^2}\right)^2 dt, \tag{28}$$

where λ controls the amount of trade-off between the data fit and penalty term. In this case, the coefficients α_i are estimated by

$$\hat{\boldsymbol{\alpha}} = \left(\boldsymbol{\Phi}^{\mathsf{T}} \boldsymbol{\Phi} + \lambda \mathbf{D}\right)^{-1} \boldsymbol{\Phi}^{\mathsf{T}} \mathbf{y}, \tag{29}$$

where $\mathbf{D}$ is the solution to the penalty in equation (28) (the integral of the square of the second derivative of x). It is possible to change the penalty term in equation (28) to some other penalty form (this is known as P-splines), where the $\mathbf{D}$ in equation (29) would be updated accordingly.

2.4. Smooth Functional Tempering

Here, we detail the method for parameter inference used in Campbell and Steele (2012). In their paper, the authors discuss two types of smooth functional tempering, one that needs to infer the initial conditions of the species concentrations and one that does not. This review uses the method that does not need to infer the initial conditions. If the initial conditions are unknown, then they must be inferred as an extra parameter in the inference procedure; however, the method described in this section effectively profiles over the initial conditions, dispensing with the need to infer them. This reduces the complexity of the procedure, which is more appealing. The reader can refer to the original publication should they wish to implement the former procedure. The choice of interpolation scheme for the concentrations $\mathbf{x}_i$ is B-splines. For an introduction to parallel tempering, see Section 2.2.

The posterior distribution of the parameters is

$$p_{\beta^{(j)}}(\boldsymbol{\theta}^{(j)}, \boldsymbol{\sigma}^{2(j)} | \mathbf{Y}, \mathbf{X}^{(j)}) \propto p(\boldsymbol{\theta}^{(j)}, \boldsymbol{\sigma}^{2(j)}) p(\mathbf{X}^{(j)} | \boldsymbol{\theta}^{(j)}, \boldsymbol{\lambda}^{(j)}) p(\mathbf{Y} | \mathbf{X}^{(j)}, \boldsymbol{\sigma}^{2(j)})^{\beta^{(j)}}, \tag{30}$$

where the superscript j denotes those variables associated with "temperature" $\beta^{(j)}$, the likelihood, $p(\mathbf{Y}|\mathbf{X}^{(j)}, \boldsymbol{\sigma}^{2(j)}) = N(\mathbf{X}^{(j)}, \boldsymbol{\sigma}^{2(j)})$,

is tempered in the same way as in equation (18), $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)$ and $p(\mathbf{X}^{(j)}|\,\boldsymbol{\theta}^{(j)}, \boldsymbol{\lambda}^{(j)})$ is

$$p(\mathbf{X}^{(j)}|\boldsymbol{\theta}^{(j)}, \boldsymbol{\lambda}^{(j)}) = \exp\left[-\sum_{i=1}^{n} \lambda_i^{(j)} ||\dot{\mathbf{x}}_i^{(j)} - f_i^{(j)}(\mathbf{X}^{(j)}, \boldsymbol{\theta}_i^{(j)}, \mathbf{t})||^2\right], \tag{31}$$

which is analogous to

$$p(\mathbf{X}^{(j)}|\boldsymbol{\theta}^{(j)}, \boldsymbol{\lambda}^{(j)}) = \exp\left[-\sum_{i=1}^{n} \lambda_i^{(j)} \sum_{t=1}^{T} \left(\dot{x}_i^{(j)}(t) - f_i^{(j)}(\mathbf{x}^{(j)}(t), \boldsymbol{\theta}_i^{(j)}, t)\right)^2\right]. \tag{32}$$

For details on tempering, see Section 2.2. In equation (31), ${\lambda_i}^{(j)}$ is the gradient mismatch parameter for species i corresponding to "temperature" $\beta^{(j)}$ (similar to the mismatch parameter ${\gamma_i}^{(j)}$ in Section 2.1). The ${\lambda_i}^{(j)}$ is chosen in advance and fixed to each "temperature" $\beta^{(j)}$ such that $0 < {\lambda_i}^{(1)} \leq \cdots \leq {\lambda_i}^{(M)} \leq \infty$, where values closer to 0 allow the gradients to be more different to one another and values closer to ∞ restrict them from being different.

Sampling from equation (30) is performed using MCMC.

2.5. Penalized Likelihood with Hierarchical Regularization

Ramsay et al. (2007) aim to conduct parameter inference in ODEs using a penalized likelihood approach and a hierarchical regularization in order to tune the gradient mismatch parameter and parameters of their interpolation scheme (splines). They perform parameter inference in a hierarchical three level approach. At level 1, they optimize the gradient mismatch parameter, in order to ensure the estimates of the coefficients of their interpolant are properly regularized by the mismatch to the ODEs. In their paper, they adjust the gradient mismatch parameter manually using numerical and visual heuristics, but suggest a way it could be achieved through generalized cross-validation, which we will detail. At level 2, the coefficients of the interpolant are optimized. While optimizing for the parameters in the final step, each time the ODE parameters and observational noise parameters are changed, they re-optimize the coefficients of the interpolant, by penalizing the differences between the gradients, which allows the ODEs to regulate the interpolant. At level 3, the ODE and observational noise parameters are estimated using a sum of squares criterion. This criterion is optimized directly for the ODE and observational noise parameters, but it is also optimized implicitly, since the sum of squares incorporates $\mathbf{x}_i$, which itself was optimized with respect to these parameters at level 2. A flow chart of these three levels can be found in **Figure 3**.

At level 1 of the three hierarchical levels, the gradient mismatch parameter is configured. To avoid the need for heuristics, Ramsay et al. (2007) suggest the use of generalized cross-validation, since the estimation of the state variables for some gradient mismatch parameter λ is usually a non-linear problem and so standard cross-validation methods are not applicable. Generalized cross-validation takes the form

$$F(\boldsymbol{\lambda}) = \frac{\sum_{i=1}^{n} ||\mathbf{y}_i - \mathbf{x}_i||^2}{\left[\sum_{i=1}^{n} \left\{T - \sum_{t=1}^{T} \frac{dx_i(t)}{dy_i(t)}\right\}\right]^2}, \tag{33}$$

where $\mathbf{y}_i$ is the data for species i, $\mathbf{x}_i$ is the interpolant corresponding to species i, n is the number of species and T is the number of timepoints. The derivatives in the denominator can be expressed as

$$\frac{dx_i(t)}{dy_i(t)} = \frac{\partial x_i(t)}{\partial \boldsymbol{\alpha}} \frac{d\boldsymbol{\alpha}}{dy_i(t)}, \tag{34}$$

where $\boldsymbol{\alpha}$ are the estimated coefficients of the splines interpolant [see equation (29)]. Calculating these derivatives takes the dependency of the data $\mathbf{y}$ and the ODE parameters $\boldsymbol{\theta}$ into account, since $\frac{d\boldsymbol{\alpha}}{d\mathbf{y}} = \frac{\partial \boldsymbol{\alpha}}{\partial \boldsymbol{\theta}} \frac{d\boldsymbol{\theta}}{d\mathbf{y}} + \frac{\partial \boldsymbol{\alpha}}{\partial \mathbf{y}}$. The estimates of $\boldsymbol{\lambda}$ will be calculated by minimizing equation (33) over values of $\boldsymbol{\lambda}$.

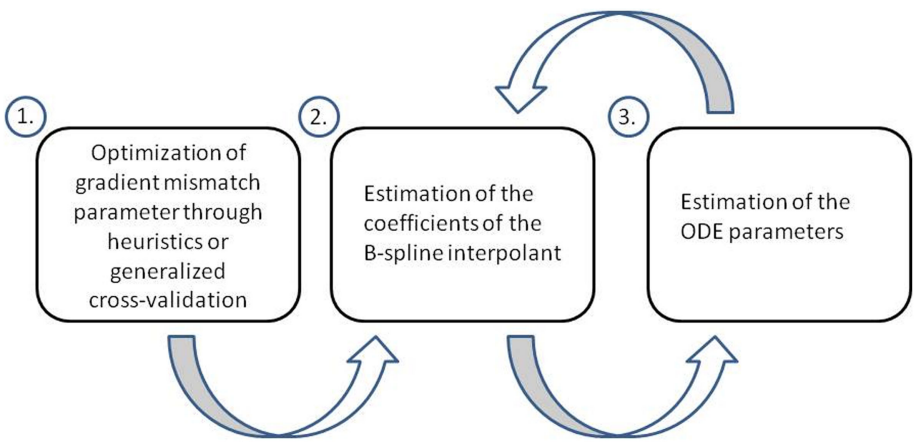

FIGURE 3 | Flow chart of the three level approach employed by Ramsay et al. (2007). At level 1, the gradient mismatch parameter is optimized either by visual or numerical heuristics or through generalized cross-validation. At level 2, the coefficients of the interpolant are estimated (splines in this method). At level 3, the ODE parameters are estimated. Levels 2 and 3 are iterated using a pseudo-delta method (see Section 2.5 for details).

The second level involves estimating the coefficients of the splines interpolant using the following criterion

$$J(\boldsymbol{\alpha}|\boldsymbol{\theta},\boldsymbol{\sigma},\boldsymbol{\lambda}) = \sum_{i=1}^{n} w_i||\mathbf{y}_i - \mathbf{x}_i||^2 + \sum_{i=1}^{n} \lambda_i \int \left[\frac{dx_i(t)}{dt} - f_i(\mathbf{x}(t),\boldsymbol{\theta}_i,t)\right]^2 dt, \quad (35)$$

where $\frac{d\mathbf{x}_i}{dt}$ is the gradient of the interpolant for species i and w_i are weights to normalize the sum of squares of different species (so that species on varying scales of measurement do not distort the sum of squares with very large or very small residuals that are simply a consequence of their magnitude or unit of measurement). Large values of λ_i mean that the gradients have to more closely match one another (since the difference between them will need to tend to 0, to compensate for the large penalty a large λ_i would produce), whereas small values would allow the gradients to differ more. The penalty term in equation (35) allows the mismatch between the gradients to regularize the estimates of the interpolant coefficients.

At the third level, the ODE parameters are optimized using the sum of squares criterion

$$S(\boldsymbol{\theta}|\boldsymbol{\lambda}) = \sum_{i=1}^{n} w_i||\mathbf{y}_i - \mathbf{x}_i||^2. \quad (36)$$

To optimize equation (36) with respect to $\boldsymbol{\theta}$, Ramsay et al. (2007) find the solution of the gradient

$$\frac{dS(\boldsymbol{\theta}|\boldsymbol{\lambda})}{d\boldsymbol{\theta}} = \frac{\partial S(\boldsymbol{\theta}|\boldsymbol{\lambda})}{\partial \boldsymbol{\theta}} + \frac{\partial S(\boldsymbol{\theta}|\boldsymbol{\lambda})}{\partial \boldsymbol{\alpha}}\frac{d\boldsymbol{\alpha}}{d\boldsymbol{\theta}} = 0. \quad (37)$$

Since the function $\boldsymbol{\alpha}(\boldsymbol{\theta})$ is not explicitly available, $\frac{d\boldsymbol{\alpha}}{d\boldsymbol{\theta}}$ is calculated by application of the implicit function theorem of differential calculus. This gives

$$\frac{d\boldsymbol{\alpha}}{d\boldsymbol{\theta}} = -\left(\frac{\partial^2 J(\boldsymbol{\alpha}|\boldsymbol{\theta},\boldsymbol{\sigma},\boldsymbol{\lambda})}{\partial \boldsymbol{\alpha}^2}\right)^{-1} \frac{\partial^2 J(\boldsymbol{\alpha}|\boldsymbol{\theta},\boldsymbol{\sigma},\boldsymbol{\lambda})}{\partial \boldsymbol{\alpha}\partial \boldsymbol{\theta}}. \quad (38)$$

2.6. Reproducing Kernel Hilbert Space

Here, we provide background for reproducing kernel Hilbert spaces (RKHS) that are used in González et al. (2013), and how they compare to Gaussian processes. RKHS interpolation is a useful tool in statistical learning, since a property of reproducing kernel Hilbert spaces, known as the representer theorem (details to follow), means that every function in an RKHS can be written as a linear combination of the kernel function evaluated at the training points. This provides a computationally fast process for interpolation, which is particularly useful in gradient matching, since the original purpose of gradient matching is to obtain a computational speed-up over methods involving calculating numerical solutions to the ODEs.

By Mercer's theorem [Mercer (1909)], we are able to represent a kernel that produces a positive definite covariance matrix in terms of eigenvalues λ_s and eigenfunctions ν_s

$$k(t_i,t_j) = \sum_{s=1}^{\infty} \lambda_s \nu_s(t_i)\nu_s(t_j). \quad (39)$$

These ν_s form an orthonormal basis for a function space

$$H = \{f : f(t) = \sum_{s=1}^{\infty} f_s \nu_s(t), \sum_{s=1}^{\infty} \frac{f_s^2}{\lambda_s} < \infty\}. \quad (40)$$

The inner product between two functions $f(t) = \sum_{s=1}^{\infty} f_s\nu_s(t)$ and $g(t) = \sum_{s=1}^{\infty} g_s\nu_s(t)$ in the space in equation (40) is defined as

$$\langle f,g\rangle_H \triangleq \sum_{s=1}^{\infty} \frac{f_s g_s}{\lambda_s}, \quad (41)$$

which Murphy (2012) shows implies that

$$\langle k(t_1,\cdot), k(t_2,\cdot)\rangle_H = k(t_1,t_2). \quad (42)$$

This is known as the reproducing property and the space of functions H is called a reproducing kernel Hilbert space. Now consider the minimization problem

$$J(f) = \frac{1}{2\sigma^2}\sum_{s=1}^{N}(y_s - f(t_s))^2 + \frac{1}{2}||f||_H^2, \quad (43)$$

where $J(f)$ is the objective function and $||f||_H$ is the norm in Hilbert space

$$||f||_H = \langle f,f\rangle_H = \sum_{s=1}^{\infty} \frac{f_s^2}{\lambda_s}. \quad (44)$$

The desired function used for interpolation should be simple and provide a good fit to the data. Complex functions with respect to the kernel in equation (39) will produce large norms, since they will need many eigenfunctions to represent them, and therefore be more heavily penalized in equation (43). Schöelkopf and Smola (2002) show that the desired function must have the following form

$$f(t) = \sum_{s=1}^{N} c_s k(t,t_s). \quad (45)$$

This is known as the representer theorem. To solve for $\mathbf{c}$, we combine equation (45) with equation (43), giving us

$$J(\mathbf{c}) = \frac{1}{2\boldsymbol{\sigma}^2}|\mathbf{y} - \mathbf{K}\mathbf{c}|^2 + \frac{1}{2}\mathbf{c}^{\mathrm{T}}\mathbf{K}\mathbf{c}, \quad (46)$$

where $\mathbf{K}$ is a matrix of kernel elements for all combinations of observed timepoints. Minimizing with respect to $\mathbf{c}$ gives us

$$\hat{\mathbf{c}} = (\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}. \quad (47)$$

Hence,

$$\hat{f}(t_*) = \sum_{s=1}^{N} \hat{c}_s k(t_*,t_s) = \mathbf{k}_*^{\mathrm{T}}(\mathbf{K} + \boldsymbol{\sigma}^2\mathbf{I})^{-1}\mathbf{y}, \quad (48)$$

where t_* is the timepoint at which one wants to make predictions and $\mathbf{k}_*$ is the vector of kernel elements for all combinations of t_* and t_s. This form is the same as a posterior mean of a Gaussian process predictive distribution.

2.7. Penalized Likelihood with RKHS

The goal of González et al. (2013) is to create a penalized likelihood function that incorporates the information of the ODEs, then using the properties of reproducing kernel Hilbert spaces to perform parameter estimation in a computationally fast manner. They consider ODEs of the form

$$\dot{\mathbf{x}}_i = g_i(\mathbf{Z}, \boldsymbol{\rho}_i, \mathbf{t}) - \delta_i \mathbf{x}_i, \tag{49}$$

or alternatively, represented in scalar form

$$\dot{x}_i(t) = g_i(\mathbf{z}(t), \boldsymbol{\rho}_i, t) - \delta_i x_i(t), \tag{50}$$

where $\mathbf{x}_i$ is the vector of mRNA concentrations for species i, δ_i is the degradation rate of the mRNA concentrations for species i, $\mathbf{Z}$ is the matrix containing the concentrations of all proteins [transcription factors (TFs)] at all timepoints, $\mathbf{z}(t)$ is the vector containing the concentrations of all proteins at timepoint t, $\boldsymbol{\rho}_i$ is a parameter vector that governs the amount of regulation that the TFs have on the i^{th} gene and $g_i(\mathbf{t}) = (g_i(t_1), \ldots, g_i(t_T))^{\mathsf{T}}$. Note the difference between equations (50) and (1). In equation (1), the regulatees can themselves act as regulators, corresponding to genes coding for transcription factors acting on other genes. In equation (50), regulators ($\mathbf{Z}$) and regulatees ($\dot{\mathbf{x}}$) are separated in what is effectively a bi-partite regulatory network structure. The ODE in equation (49) depends on the state variables $\mathbf{x}_i$ only by a linear decay term δ_i. Consider a differencing matrix $\mathbf{D}$, where

$$\mathbf{D} = \Upsilon \begin{bmatrix} -1 & 1 & 0 & \ldots & \ldots & 0 \\ -1 & 0 & 1 & 0 & \ldots & 0 \\ 0 & -1 & \ddots & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & \ldots & \ldots & -1 & 1 \end{bmatrix}, \tag{51}$$

and $\Upsilon = diag\left(\frac{1}{t_2-t_1}, \frac{1}{t_3-t_1}, \frac{1}{t_4-t_2}, \ldots, \frac{1}{t_T-t_{T-2}}, \frac{1}{t_T-t_{T-1}}\right)$. We can then approximate equation (49) as

$$\mathbf{D}\mathbf{x}_i = g_i(\mathbf{Z}, \boldsymbol{\rho}_i, \mathbf{t}) - \delta \mathbf{x}_i. \tag{52}$$

To demonstrate how $\mathbf{D}\mathbf{x}_i$ is computed, as an example let us consider $\mathbf{x}_i = (x(t_1), \ldots, x(t_5))^{\mathsf{T}}$ and $\mathbf{t} = (1, 2, \ldots, 5)^{\mathsf{T}}$. Then,

$$\begin{aligned} \mathbf{D}\mathbf{x}_i &= \begin{bmatrix} \frac{1}{2-1} & & & & \\ & \frac{1}{3-1} & & & \\ & & \frac{1}{4-2} & & \\ & & & \frac{1}{5-3} & \\ & & & & \frac{1}{5-4} \end{bmatrix} \\ &\quad \times \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \end{bmatrix} \\ &= \left[\frac{-x(1)+x(2)}{1}, \quad \frac{-x(1)+x(3)}{2}, \quad \frac{-x(2)+x(4)}{2}, \right. \\ &\quad \left. \frac{-x(3)+x(5)}{2}, \frac{-x(4)+x(5)}{1} \right]^{\mathsf{T}}. \end{aligned} \tag{53}$$

Writing $\mathbf{P} = \mathbf{D} + \delta_i \mathbf{I}$ ($\mathbf{I}$ here is the identity matrix) gives us the following penalty to be incorporated into the likelihood term

$$\boldsymbol{\Omega}(\mathbf{x}_i) = ||\mathbf{P}\mathbf{x}_i - g_i(\mathbf{Z}, \boldsymbol{\rho}_i, \mathbf{t})||^2. \tag{54}$$

Equation (52) implies that $\mathbf{P}\mathbf{x}_i - g_i(\mathbf{Z}, \boldsymbol{\rho}_i, \mathbf{t}) = 0$. Rather than solving this equation explicitly, it is used as a penalty term within a regression context, i.e., the $||f||_H^2$ term in equation (43) will be replaced by equation (54). However, equation (54) cannot be expressed as a norm of $\mathbf{x}_i$ within the RKHS framework, since $\mathbf{x}_i = \mathbf{0}$ does not necessarily imply that $\boldsymbol{\Omega}(\mathbf{x}_i) = 0$. The authors therefore transform the state variables $\mathbf{x}_i$ (and subsequently $\mathbf{y}_i$) in order to make them compatible. Consider instead

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{P}^{-1} g_i(\mathbf{Z}, \boldsymbol{\rho}_i, \mathbf{t}). \tag{55}$$

It is straightforward to see that multiplying both sides of equation (55) by $\mathbf{P}$ and taking squared norms gives us the exact form of equation (54) $\left(\|\mathbf{P}\tilde{\mathbf{x}}_i\|^2 = \left\|\mathbf{P}\mathbf{x}_i - g_i(\mathbf{Z}, \boldsymbol{\rho}_i, \mathbf{t})\right\|^2\right)$. Likewise, the data are transformed by

$$\tilde{\mathbf{y}}_i = \mathbf{y}_i - \mathbf{P}^{-1} g_i(\mathbf{Z}, \boldsymbol{\rho}_i, \mathbf{t}), \tag{56}$$

to correspond with $\tilde{\mathbf{x}}_i$. The penalty term in equation (54) now becomes

$$\boldsymbol{\Omega}(\tilde{\mathbf{x}}_i) = ||\mathbf{P}\tilde{\mathbf{x}}_i||^2 = \langle \mathbf{P}\tilde{\mathbf{x}}_i, \mathbf{P}\tilde{\mathbf{x}}_i \rangle = \tilde{\mathbf{x}}_i^{\mathsf{T}} \mathbf{P}^{\mathsf{T}} \mathbf{P} \tilde{\mathbf{x}}_i. \tag{57}$$

Equation (57) is now a proper norm, since when $\tilde{\mathbf{x}}_i = \mathbf{0}$, this implies $\boldsymbol{\Omega}(\tilde{\mathbf{x}}_i) = 0$. Denote $\mathbf{K} = (\mathbf{P}^{\mathsf{T}}\mathbf{P})^{-1}$. $\mathbf{K}$ is a matrix of kernel elements that define a unique RKHS. Hence,

$$\boldsymbol{\Omega}(\tilde{\mathbf{x}}_i) = ||\tilde{\mathbf{x}}_i||_H^2 = \mathbf{c}^{\mathsf{T}} \mathbf{K} \mathbf{c}, \tag{58}$$

[where $\mathbf{c}$ is given in equation (47), and equation 58 is used as the term in the far right of equation 46, see Section 2.6 for details]. By using equations (47) and (48), we obtain closed-form expressions for the transformed state variables [and the original expressions can be recovered using equation (55)]

$$\tilde{\mathbf{x}}_i = \mathbf{K}(\mathbf{K} + 2\lambda_i \boldsymbol{\Sigma})^{-1} \mathbf{y}_i, \tag{59}$$

where λ is a penalty parameter, and $\boldsymbol{\Sigma}$ is the covariance matrix of the data [which generalizes equation (47) since the observational error of our data may not be independent between species]. In practice, not all ODEs are of the form in equation (50), which only depends on the state variables by a linear decay term. Hence, the authors need to transform any ODE that is not of this form into 2 parts. Terms in part (1) will have a dependency on the state variables only by a linear decay term and can be modeled using the RKHS method and estimated by equation (59). Terms in part (2) cannot fit this framework and are modeled using splines. For example, consider $[\dot{V}]$ in the FitzHugh–Nagumo ODEs (for more details, see Section 4)

$$[\dot{V}] = \psi \left([V] - \frac{[V]^3}{3} + [R] \right), \tag{60}$$

where the square brackets denote the time-dependent concentration for that species, the dot over the V is shorthand for the temporal derivative $\frac{d}{dt}$ of V and ψ is a parameter. Since the state variables do not only depend on a linear decay term, equation (60) needs to be transformed. Part (1) will be expressed by $[\dot{V}] - \psi[V]$, where now the dependency on the state variables is only by a linear decay term and hence can be fitted using the RKHS method. Part (2) will be $\psi\left(-\frac{[\hat{V}]^3}{3} + [\hat{R}]\right)$, which is fitted using splines, where $[\hat{V}]$ and $[\hat{R}]$ are spline estimates for $[V]$ and $[R]$, respectively.

The penalized log-likelihood function can now be expressed by

$$l(\boldsymbol{\rho}_i, \delta_i, \boldsymbol{\Sigma}, \boldsymbol{\alpha}_i, \mathbf{c}|\tilde{\mathbf{y}}_i) = \sum_{i=1}^{n} \left[-\frac{1}{2}(\tilde{\mathbf{y}}_i - \tilde{\mathbf{x}}_i)^{\mathsf{T}} \Sigma^{-1} (\tilde{\mathbf{y}}_i - \tilde{\mathbf{x}}_i) - \frac{1}{2} ln|\Sigma| \right] - \sum_{i=1}^{n} \lambda_i \boldsymbol{\Omega}(\tilde{\mathbf{x}}_i), \quad (61)$$

where $\boldsymbol{\alpha}_i$ is the vector containing the coefficients from the spline interpolant for species *i*, see equation (26). Given that the gradient matching is dependent on the differencing operator, it is important to note that points further apart in time will produce continually poorer estimates of the gradient and thus poorer gradient matching. González et al. (2013) attempt to circumvent this issue by data augmentation. They infer the latent variables at additional unobserved timepoints with the expectation maximization (EM) algorithm, which emulates more datapoints, in order to obtain more accurate gradient estimates. Parameter estimation in an approximate penalized maximum likelihood sense can be carried out with standard non-linear optimization algorithms, such as quasi-Newton or conjugate gradients.

3. SUMMARY OF METHODS

This section provides a brief summary of the methods throughout the review, as described in Section 2. Since many methods and settings are used in this review for comparison purposes, for ease of reading, abbreviations are used. **Table 3** is a reference for those methods and an overview of the methods follows.

INF (Section 2.1): this method conducts parameter inference using adaptive gradient matching and Gaussian processes. The penalty mismatch parameter $\boldsymbol{\gamma}$ (where $\boldsymbol{\gamma}$ is the vector of mismatch penalty parameter values at different "temperatures") is inferred rather than tempered.

LB2 (Sections 2.1 and 2.2): this method conducts parameter inference using adaptive gradient matching and Gaussian processes. The penalty mismatch parameter $\boldsymbol{\gamma}$ is tempered in log base 2 increments, see **Table 1** for details.

LB10 (Sections 2.1 and 2.2): as with LB2, parameter inference is conducted using adaptive gradient matching and Gaussian processes; however, the penalty mismatch parameter $\boldsymbol{\gamma}$ is tempered in log base 10 increments, see **Table 1** for details.

C&S (Section 2.4): parameter inference is carried out using adaptive gradient matching and tempering of the mismatch parameter. The choice of interpolation scheme is B-splines.

RAM (Section 2.5): this technique uses a non-Bayesian optimization process for parameter inference. The method penalizes the difference between the gradients using splines and a hierarchical 3 level regularization approach is used to configure the tuning parameters.

GON (Section 2.7): parameter inference is conducted in a non-Bayesian fashion, implementing a reproducing kernel Hilbert space (RKHS) and penalized likelihood approach. Comparisons between RKHS and GPs have been previously explored conceptually [for example, see Rasmussen and Williams (2006) and Murphy (2012)], and in this review we analyze them empirically in the specific context of gradient matching. The RKHS gradient matching method in González et al. (2013) obtains the interpolant gradient using a differencing operator.

TABLE 3 | Abbreviations of the methods used throughout this review.

Abbreviation	Method	Reference
GON	Reproducing kernel Hilbert space and penalized likelihood	González et al. (2013)
RAM	Splines and hierarchical regularization	Ramsay et al. (2007)
INF	Inference of the gradient mismatch parameter using GPs	Dondelinger et al. (2013)
LB2	Tempered mismatch parameter using GPs in log base 2 increments	Macdonald and Husmeier (2015)
LB10	Tempered mismatch parameter using GPs in log base 10 increments	Macdonald and Husmeier (2015)
C&S	Tempered mismatch parameter using splines-based smooth functional tempering (SFT)	Campbell and Steele (2012)

TABLE 4 | Particular settings of Campbell and Steele (2012)'s method.

Abbreviation	Definition	Details
10C	10 Chains	When comparing methods, it was of interest to see how the performance depended on the number of parallel MCMC chains, as originally the authors used 4 chains
Obs20	20 Observations	Originally, the authors used 401 observations. This was reduced to a dataset size more usual with these types of experiments to observe the dependency of the methods on the amount of data
15K	15 Knots	The C&S method uses B-splines interpolation. The original tuning parameters from the authors' paper were changed to observe the sensitivity of the parameter estimation from these tuning parameters
P3	Polynomial order 3 (Cubic Spline)	The original polynomial order is 5. Again, this was changed to observe the sensitivity of the parameter estimation from these tuning parameters

Table 4 outlines particular settings with some of the methods in **Table 3**. The ranges of the penalty parameter for γ, for LB2 and LB10 methods, are given in **Table 1**. The increments are equidistant on the log scale. The M β_is from 0 to 1 are set by taking a series of equidistant M values and raising them to the power 5 [Friel and Pettitt (2008)].

4. DATA

4.1. FitzHugh–Nagumo

These equations model the voltage potential across the cell membrane of the axon of giant squid neurons [FitzHugh (1961); Nagumo et al. (1962)]. There are two "species": voltage (V) and recovery variable (R), and 3 parameters; α, β, and ψ. The square brackets denote the time-dependent concentration for that species and a dot over a symbol is shorthand for the temporal derivative $\frac{d}{dt}$ of that symbol:

$$[\dot{V}] = \psi\left([V] - \frac{[V]^3}{3} + [R]\right); \quad [\dot{R}] = -\frac{1}{\psi}\left([V] - \alpha + \beta * [R]\right). \tag{62}$$

An example of the signals produced from these ODEs can be found in **Figure 4**.

4.2. Protein Signaling Transduction Pathway

These equations model protein signaling transduction pathways in a signal transduction cascade, where the free parameters are kinetic parameters governing how quickly the proteins ("species") convert to one another [Vyshemirsky and Girolami (2008)]. There are 5 "species" (S, dS, R, RS, Rpp) and 6 parameters (k_1, k_2, k_3, k_4, V, K_m). The system describes the phosphorylation of a protein, $R \rightarrow Rpp$ [equation (67)], catalyzed by an enzyme S, via an active protein complex [RS, equation (66)], where the enzyme is subject to degradation [$S \rightarrow dS$, equation (64)]. The chemical kinetics are described by a combination of mass action kinetics [equations (63), (64), and (66)] and Michaelis–Menten kinetics [equations (65) and (67)]. A graphical representation of this system can be seen in **Figure 5**. The square brackets denote the time-dependent concentration for that species and a dot over a symbol is shorthand for the temporal derivative $\frac{d}{dt}$ of that symbol:

$$[\dot{S}] = -k_1 * [S] - k_2 * [S] * [R] + k_3 * [RS], \tag{63}$$

$$[\dot{dS}] = k_1 * [S], \tag{64}$$

$$[\dot{R}] = -k_2 * [S] * [R] + k_3 * [RS] + \frac{V * [Rpp]}{K_m + [Rpp]}, \tag{65}$$

$$[\dot{RS}] = k_2 * [S] * [R] - k_3 * [RS] - k_4 * [RS], \tag{66}$$

$$[\dot{Rpp}] = k_4 * [RS] - \frac{V * [Rpp]}{K_m + [Rpp]}. \tag{67}$$

An example of a typical signal produced from these ODEs can be found in **Figure 6**.

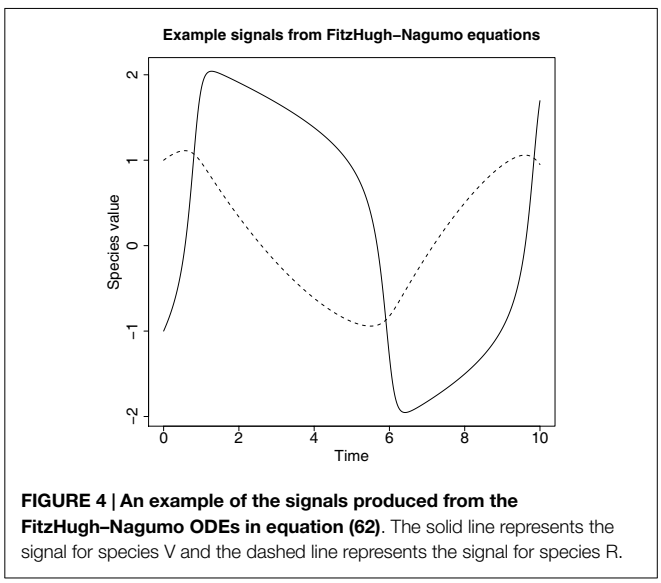

FIGURE 4 | An example of the signals produced from the FitzHugh–Nagumo ODEs in equation (62). The solid line represents the signal for species V and the dashed line represents the signal for species R.

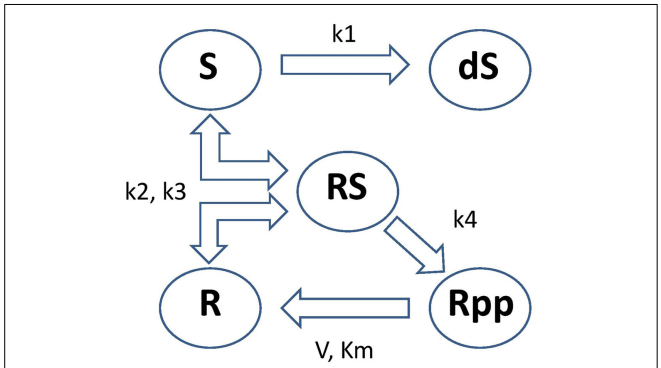

FIGURE 5 | Graphical representation of the protein signaling transduction pathway in equations (63)–(67). There are 5 "species" (S, dS, R, RS, Rpp) and 6 parameters (k_1, k_2, k_3, k_4, V, K_m). The system describes the phosphorylation of a protein, $R \rightarrow Rpp$ [equation (67)], catalyzed by an enzyme S, via an active protein complex [RS, equation (66)], where the enzyme is subject to degradation [$S \rightarrow dS$, equation (64)]. Figure adapted from Vyshemirsky and Girolami (2008).

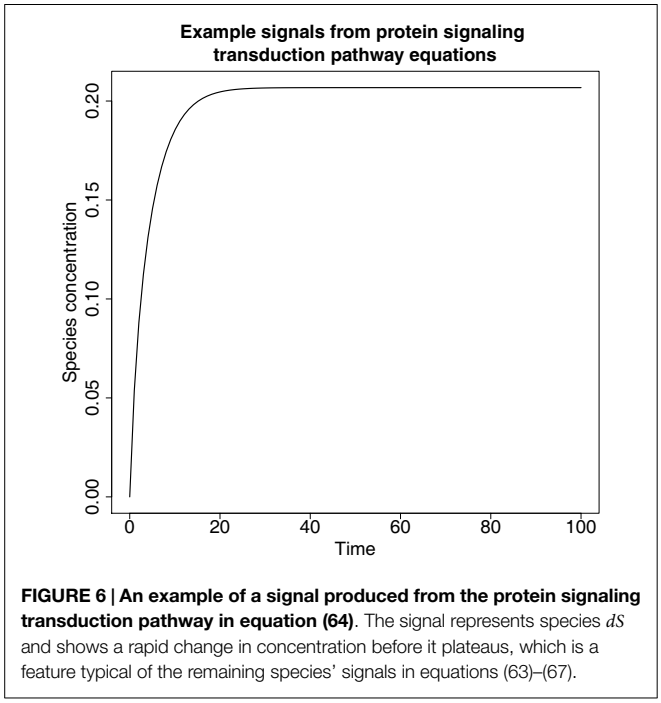

FIGURE 6 | An example of a signal produced from the protein signaling transduction pathway in equation (64). The signal represents species dS and shows a rapid change in concentration before it plateaus, which is a feature typical of the remaining species' signals in equations (63)–(67).

5. SIMULATION

For those methods for which software was unavailable at the time [Ramsay et al. (2007); González et al. (2013)], results were compared directly with the results from the original publications. To this end, test data were generated in the same way as described by the authors. For methods for which software was available at the time [Campbell and Steele (2012); Dondelinger et al. (2013); Macdonald and Husmeier (2015)], the evaluation was repeated twice, first on data equivalent to those used in the original publications, and again on new data generated with different (more realistic) parameter settings. For comparisons with Bayesian methods, the authors' specifications for the priors on the ODE parameters were used. For comparisons with non-Bayesian methods, the methods of Dondelinger et al. (2013) and Macdonald and Husmeier (2015) were applied with the parameter prior from Campbell and Steele (2012), since the ODE model was the same.

5.1. Reproducing Kernel Hilbert Space Method (Section 2.7)

The method was tested on the FitzHugh–Nagumo data (see Section 4) with the following parameters: $\alpha = 0.2$; $\beta = 0.2$, and $\psi = 3$. Starting from initial values of $(-1, -1)$ for the two "species", 50 timepoints were generated over the time course [0, 20], producing 2 periods, with iid Gaussian noise (SD = 0.1) added. Fifty independent datasets were generated in this way.

5.2. Splines and Hierarchical Regularization Method (Section 2.5)

This method was included in the study by González et al. (2013), and the results in this review are from the original paper. For a proper comparison, the methods of Dondelinger et al. (2013) and Macdonald and Husmeier (2015) were applied in the same way as in for the comparison with González et al. (2013).

5.3. Tempered Mismatch Parameter Using Splines-Based Smooth Functional Tempering (Section 2.4)

The method was tested on the FitzHugh–Nagumo system with the following parameter settings: $\alpha = 0.2$; $\beta = 0.2$, and $\psi = 3$, starting from initial values of $(-1, 1)$ for the two "species" [note the different starting values to the set-up in González et al. (2013)]. Four hundred and one observations were simulated over the time course [0, 20] (producing 2 periods) and Gaussian noise was added with SD {0.5, 0.4} to each respective "species". The original settings were used for inferring the ODE parameters: splines of polynomial order 5 with 301 knots; four parallel tempering chains associated with gradient mismatch parameters {10, 100, 1000, 10,000}; parameter prior distributions for the ODE parameters: $\alpha \sim N(0,\ 0.4^2)$, $\beta \sim N(0,\ 0.4^2)$, and $\psi \sim \chi^2_2$.

In addition to comparing the methods of Dondelinger et al. (2013) and Macdonald and Husmeier (2015) with these original settings, the following modifications were made to test the robustness of the procedures with respect to these (rather arbitrary) choices. The number of observations was reduced from 401 to 20 over the time course [0, 10] (producing 1 period) to reflect more closely the amount of data typically available from current systems biology projects. For these smaller datasets, the number of knots for the splines was reduced to 15 (keeping the same proportionality of knots to datapoints as before), and a different polynomial order was tested: 3 instead of 5. Due to the high computational costs of the Campbell and Steele (2012) method (roughly $1\frac{1}{2}$ weeks for a run), only 3 MCMC simulations on 3 independent datasets could be run. The respective posterior samples were combined, to approximately marginalize over datasets, and thereby remove their potential particularities. For a fair comparison, the tempering schedule in Campbell and Steele (2012) was applied to the methods of Dondelinger et al. (2013) and Macdonald and Husmeier (2015) such that 4 parallel chains were used rather than 10.

5.4. Inference of the Gradient Mismatch Parameter Using GPs (Section 2.1)

The methods of Dondelinger et al. (2013) and Macdonald and Husmeier (2015) were applied in the same way as in the original publication of Dondelinger et al. (2013), selecting the same kernels and parameter/hyperparameter priors. Data were generated from the protein signal transduction pathway, described in Section 4, with the following settings; ODE parameters: ($k_1 = 0.07$, $k_2 = 0.6$, $k_3 = 0.05$, $k_4 = 0.3$, $V = 0.017$, $K_m = 0.3$); initial values of the species: ($S = 1$, $dS = 0$, $R = 1$, $RS = 0$, $Rpp = 0$); 15 timepoints covering one period, {0, 1, 2, 4, 5, 7, 10, 15, 20, 30, 40, 50, 60, 80, 100}. Multiplicative iid Gaussian noise of SD = 0.1 was used to distort the signals, in order to reflect observational error that would be obtained in experiments. For Bayesian inference, a $\Gamma(4, 0.5)$ prior was used for the ODE parameters. For the GP, we used the same kernel as in Dondelinger et al. (2013); see below for details. In addition to this ODE system, these methods were also applied to the set-ups previously described for the FitzHugh–Nagumo model.

5.5. Choice of Kernel

For the GP, a suitable kernel needs to be chosen, which defines a prior distribution in function space. Two kernels are considered in this review [to match the authors' set-ups in Dondelinger et al. (2013)], the radial basis function (RBF) kernel

$$k(t_i, t_j) = \sigma^2_{RBF} \exp\left(-\frac{(t_i - t_j)^2}{2l^2}\right), \tag{68}$$

with hyperparameters σ^2_{RBF} and l^2, and the sigmoid variance kernel

$$k(t_i, t_j) = \sigma^2_{sig} arcsin \frac{a + (bt_i t_j)}{\sqrt{(a + (bt_i t_i) + 1)(a + (bt_j t_j) + 1)}}, \tag{69}$$

with hyperparameters σ^2_{sig}, a and b [Rasmussen and Williams (2006)].

To choose initial values for the hyperparameters, a standard GP regression model (i.e., without the ODE part) is fitted using maximum likelihood. The interpolant is then inspected to decide

whether it adequately represents the prior knowledge of the signal. For the data generated from the FitzHugh–Nagumo model, the RBF kernel provides a good fit to the data. For the protein signaling transduction pathway, the non-stationary nature of the data is not represented properly with the RBF kernel, which is stationary [Rasmussen and Williams (2006)], in confirmation of the findings in Dondelinger et al. (2013). Following Dondelinger et al. (2013), the sigmoid variance kernel was used, which is non-stationary [Rasmussen and Williams (2006)] and this provided a considerably improved fit to the data.

5.6. Other Settings

Finally, the values for the variance mismatch parameter of the gradients, γ, needs to be configured for the method in Macdonald and Husmeier (2015). Log base 2 and log base 10 increments were used (initializing at 1), since studies that indicate reasonable values are limited [see Calderhead et al. (2008) and Friel and Pettitt (2008)]. All parameters were initialized with a random draw from the respective priors (apart from GON and RAM, which did not use priors).

6. RESULTS

We present the results in the same way the authors of the methods we are comparing presented them in the original papers. For the methods we had obtained the authors' code for, we also present the root mean square (RMS) values in function space. First, the signal was reconstructed with the sampled parameters and then the true signal was subtracted (signal created with true parameters and no observational noise added). The RMS was calculated on these residuals. It is important to assess the methods on this criterion as well as looking at the parameter uncertainty, as some parameters might only be weakly identifiable, corresponding to ridges in the likelihood landscape. In other words, large uncertainty in parameter estimates may not necessarily imply a poor performance by a method, if the reconstructed signals for all groups of sampled parameters were close to the truth.

All distributions of the results in this section are displayed graphically as boxplots, which display whiskers that extend from the lower ($Q1$) and upper ($Q3$) quartiles of the box, to boundaries defined by $Q1 - 1.5(Q3 - Q1)$ and $Q3 + 1.5(Q3 - Q1)$. All values outside these boundaries are considered outliers and drawn as a circle.

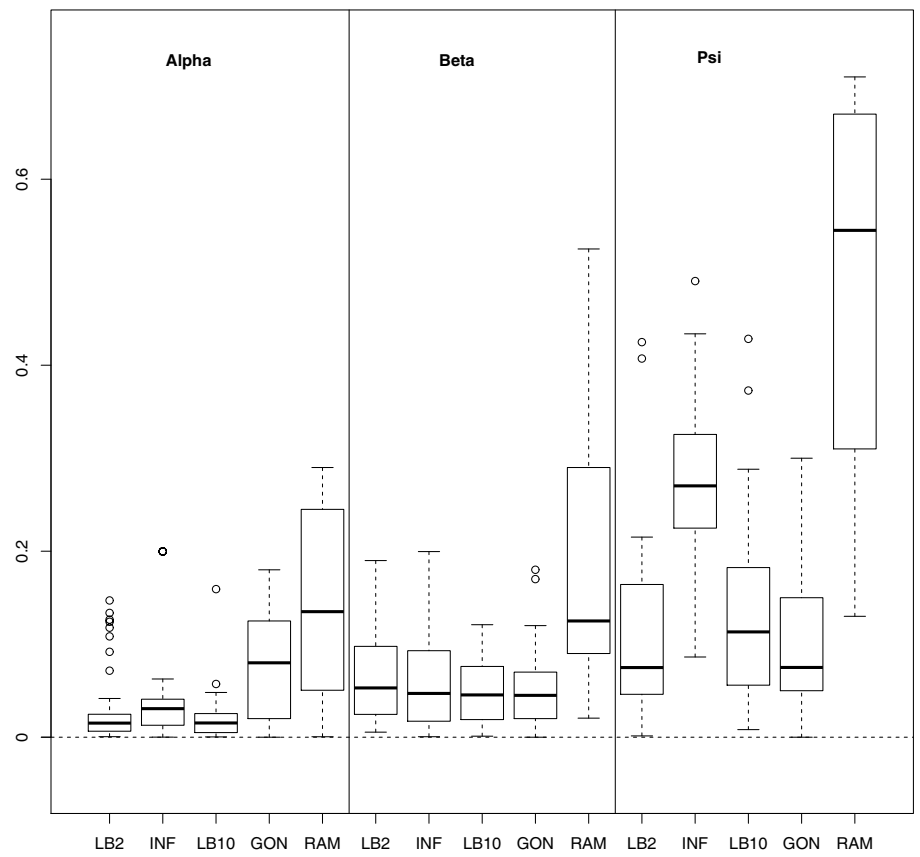

FIGURE 7 | Boxplots of the distributions of the absolute differences of an estimate to the true parameter over 50 datasets. The three sections from left to right represent the parameters α, β, and ψ from equation (62). Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method, GON's method [boxplot reconstructed from González et al. (2013)], and RAM's method [boxplot reconstructed from González et al. (2013)]. For an explanation of the boxplot form, see the beginning of Section 6. Figure reconstructed from Macdonald and Husmeier (2015).

6.1. Reproducing Kernel Hilbert Space (Section 2.7) and Hierarchical Regularization (Section 2.5) Methods

For this configuration, to judge the performance of the methods, we used the same concept as in GON to examine our results. For each parameter, the absolute value of the difference between an estimate and the true parameter ($|\hat{\theta}_i - \theta_i|$) was computed and the distribution across the datasets was examined. For the LB2, LB10, and INF methods, the median of the sampled parameters was used since it is a robust estimator. Looking at **Figure 7**, the LB2, LB10, and INF methods do as well as the GON method for 2 parameters (INF doing slightly worse for ψ) and outperform it for 1 parameter. All methods outperform the RAM method.

6.2. Tempered Mismatch Parameter Using Splines-Based Smooth Functional Tempering (Section 2.4)

For this set-up, the entire posterior distributions were examined. The posterior distributions were averaged over datasets in order to present the overall performance of each method, not confounded by the particular observational error that was added to a dataset. The C&S method shows good performance over all parameters in the one case where the number of observations is 401, the number of knots is 301, and the polynomial order is 3 (cubic spline), since the bulk of the average posterior distributions of the sampled parameters surrounds the true parameters in **Figures 8** and **10** and is close to the true parameter in **Figure 9**. However, these settings require a great deal of "hand-tuning" or time expensive cross-validation and would be very difficult to set when using real data. The sensitivity of the splines-based method can be seen in the other settings, where the results deteriorate. It is also important to note that when the dataset size was reduced, the cubic spline performed very badly. This inconsistency makes these methods very difficult to apply in practice. The LB2, LB10, and INF methods consistently outperform the C&S method with the bulk of the average posterior distributions overlapping or being closer to the true parameters. On the set-up with 20 observations, for 4 chains and 10 chains, the INF method produced largely different estimates over the datasets, as depicted by the wide boxplots and long tails. The long tails in all of these distributions are due to the combination of estimates from different datasets.

By examining **Figure 11**, we can see how the methods perform in function space. The RMS values for some of the C&S set-ups were very large, so for graphical viewing purposes, we

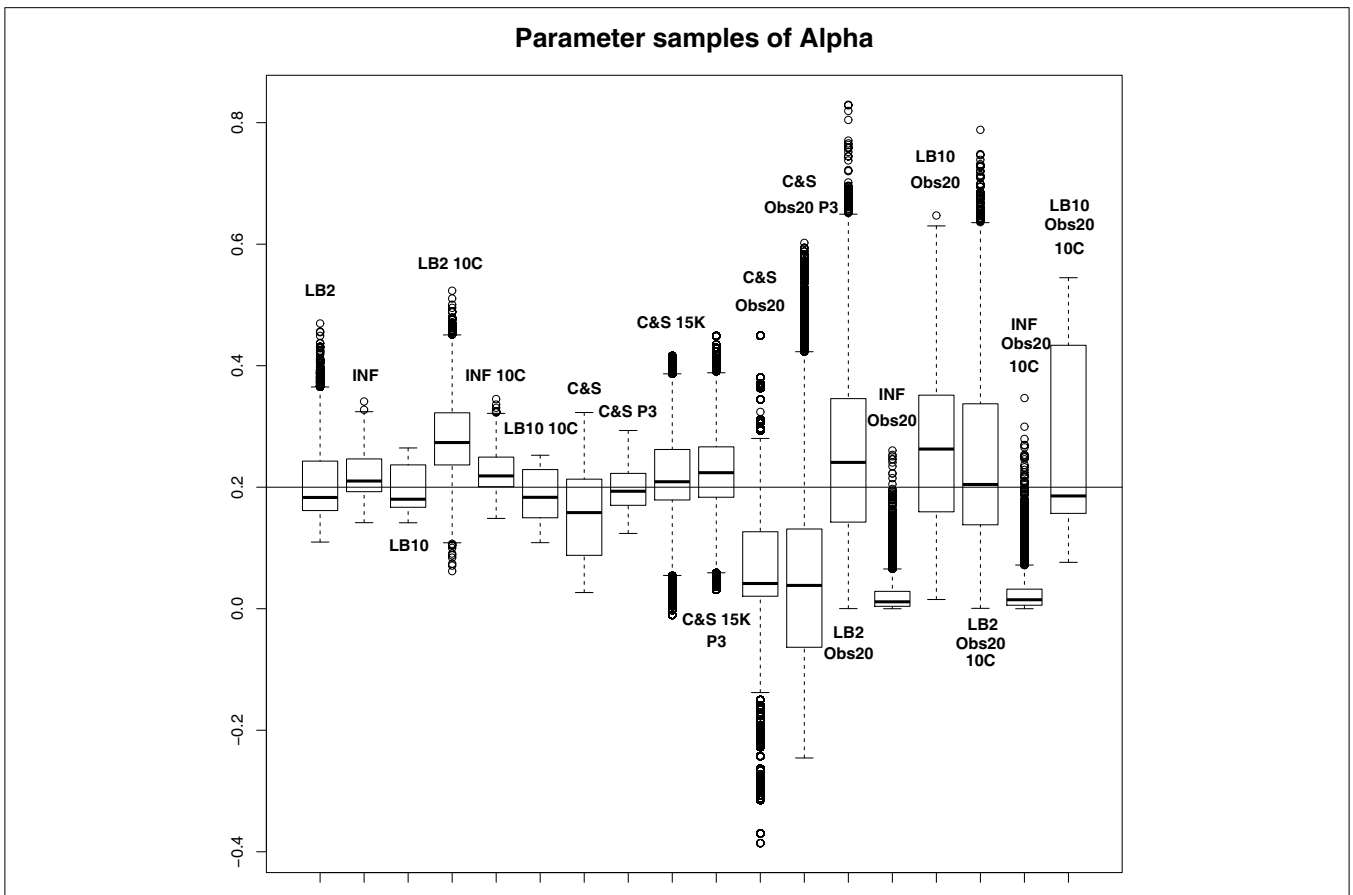

FIGURE 8 | Average posterior distributions of parameter α from equation (62) over 3 datasets. From left to right: LB2, INF, LB10, LB2 10C, INF 10C, LB10 10C, C&S, C&S P3, C&S 15K, C&S 15K P3, C&S Obs20, C&S Obs20 P3, LB2 Obs20, INF Obs20, LB10 Obs20, LB2 Obs20 10C, INF Obs20 10C, and LB10 Obs20 10C. The solid line is the true parameter. For definitions, see **Tables 3** and **4**. For an explanation of the boxplot form, see the beginning of Section 6. Figure reconstructed from Macdonald and Husmeier (2015).

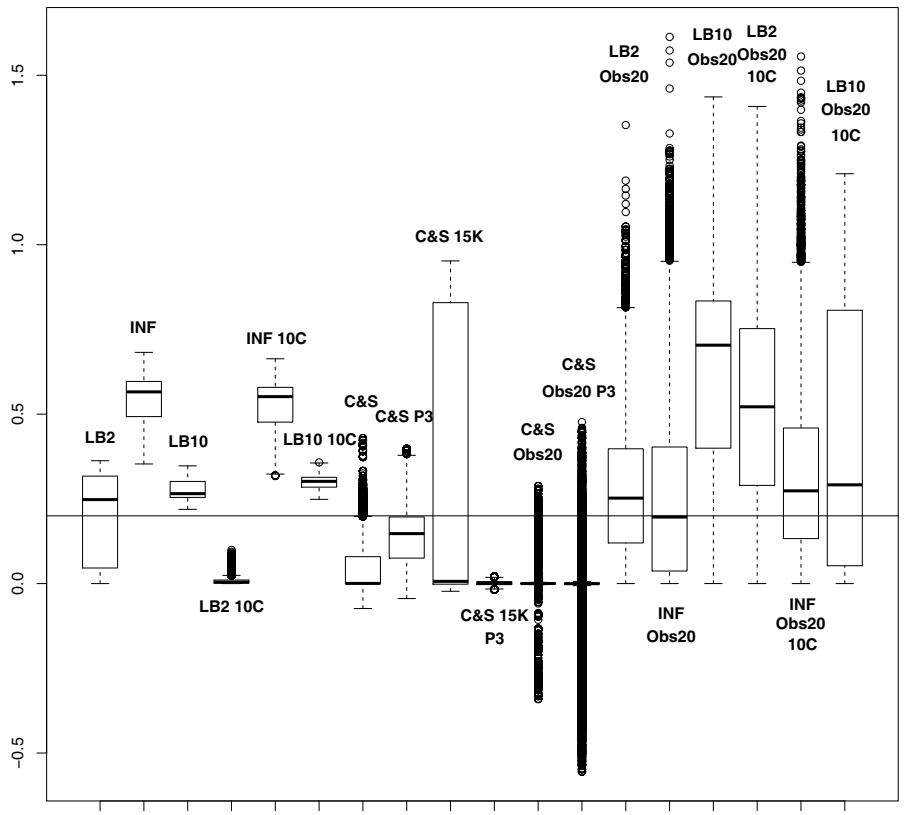

FIGURE 9 | Average posterior distributions of parameter β from equation (62) over 3 datasets. From left to right: LB2, INF, LB10, LB2 10C, INF 10C, LB10 10C, C&S, C&S P3, C&S 15K, C&S 15K P3, C&S Obs20, C&S Obs20 P3, LB2 Obs20, INF Obs20, LB10 Obs20, LB2 Obs20 10C, INF Obs20 10C, and LB10 Obs20 10C. The solid line is the true parameter. For definitions, see **Tables 3** and **4**. For an explanation of the boxplot form, see the beginning of Section 6. Figure reconstructed from Macdonald and Husmeier (2015).

applied a squashing function

$$f(RMS) = \frac{RMS}{1 + RMS}, \tag{70}$$

where $f(RMS) = RMS$ for $RMS << 1$, and $f(RMS) = 1$ for $RMS \to \infty$. The RMS values have been monotonically transformed and values closer to 0 show better performance, whereas values closer to 1 show poorer performance. These results reinforce what we saw from the parameter estimates. The C&S performs well only in the one case where there was a large number of datapoints (401) and a cubic spline was used. The other set-ups for C&S perform very poorly, including the case where the cubic spline was used with a smaller dataset size. The LB2, INF, and LB10 methods perform well and similarly across the different set-ups, with LB10 performing slightly better in some scenarios.

6.3. Inference of the Gradient Mismatch Parameter Using GPs (Section 2.1)

In order to see how the tempering method in Macdonald and Husmeier (2015) performs in comparison to the INF method, we can examine the results from the protein signaling transduction pathway (see Section 4), as well as comparing the results in the previous set-ups. The distributions of the posterior parameter samples minus the true values for the protein signaling transduction pathway are shown in **Figure 12**. The INF method was unable to converge properly for some of the datasets. In order to present the average performance of the methods, for INF, LB2, and LB10, the root mean square (RMS) of the difference between the posterior parameter samples and the true values was calculated across all datasets. The results from the dataset which produced the median RMS are shown for each method.

By examining **Figure 12**, we can see that for each parameter, the bulk of the distributions is close to the true value and so the methods are performing reasonably. Overall, there does not appear to be a significant difference between the INF, LB2, and LB10 methods for this model. **Figure 13** shows the distribution of RMS values for INF, LB2, and LB10 methods in terms of deviance from the true time series. All three methods perform similarly to one another, with RMS values close to zero.

For the set-up in Sections 2.7 and 2.5: **Figure 14** shows the Expected Cumulative Distribution Functions (ECDFs) of the absolute difference of the posterior parameter samples to the true values, for INF, LB2, and LB10. Included are the p-values

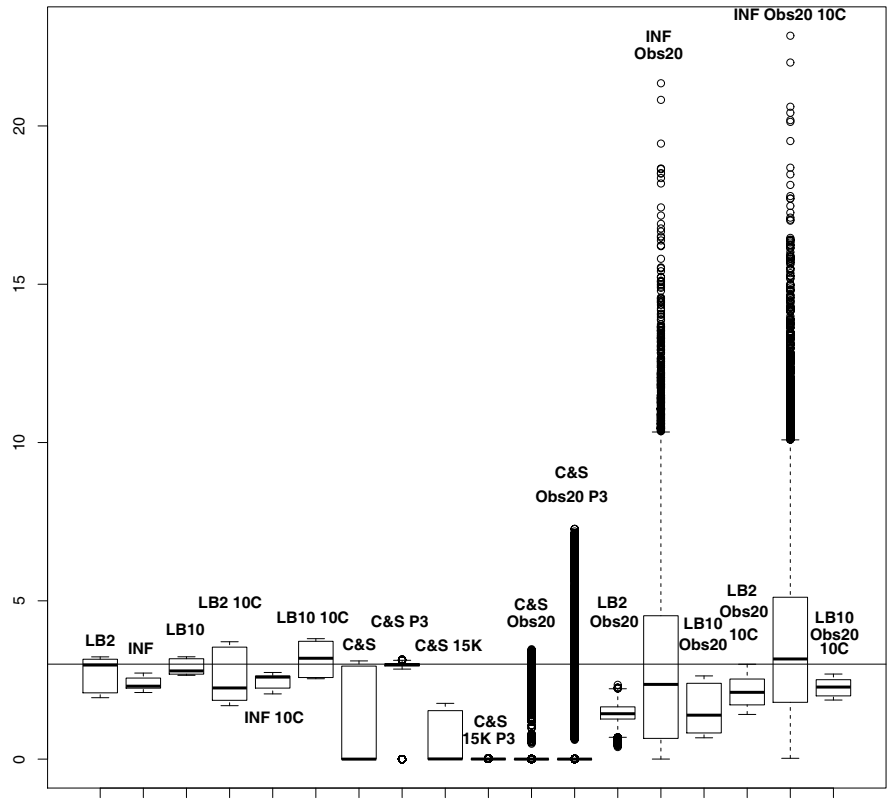

FIGURE 10 | Average posterior distributions of parameter ψ from equation (62) over 3 datasets. From left to right: LB2, INF, LB10, LB2 10C, INF 10C, LB10 10C, C&S, C&S P3, C&S 15K, C&S 15K P3, C&S Obs20, C&S Obs20 P3, LB2 Obs20, INF Obs20, LB10 Obs20, LB2 Obs20 10C, INF Obs20 10C, and LB10 Obs20 10C. The solid line is the true parameter. For definitions, see **Tables 3** and **4**. For an explanation of the boxplot form, see the beginning of Section 6. Figure reconstructed from Macdonald and Husmeier (2015).

for 2-sample, 1-sided Kolmogorov–Smirnov tests. If a distribution's ECDF is significantly higher than another, this constitutes as better parameter estimation. A higher curve means that a method has more values that lie in the lower range of absolute error.

Figure 14 shows that both the LB2 and LB10 methods outperform the INF method, shown by p-values of less than the standard significance level of 0.05. Therefore, we conclude that the CDFs for LB2 and LB10 are significantly higher than those for INF. Since we are dealing with absolute errors, this means that the parameter estimates from the LB2 and LB10 methods are closer to the true parameters than the INF method. The LB2 and LB10 methods show no significant difference to each other.

For the set-up in Section 2.4: The LB2 and LB10 methods do well over all the parameters and dataset sizes, with most of the mass of the distributions surrounding or being situated close to the true parameters. The LB2 does better than the LB10 for 4 parallel chains (distributions overlapping the true parameter for all three parameters) and the LB10 outperforms the LB2 for 10 parallel chains (distribution overlapping true parameter in **Figure 8**, being closer to the true parameter in **Figure 9**, and narrower and more symmetric around the true parameter in **Figure 10**). The INF method's bulk of parameter sample distributions is located close to the true parameters for all dataset sizes. However, the decrease in uncertainty is at the expense of bias. When reducing the dataset to 20 observations, for 4 and 10 chains, the inference deteriorates and is outperformed by the LB2 and LB10 methods. This could be due to the parallel tempering scheme constraining the mismatch between the gradients in chains closer to the posterior, allowing for better estimates of the parameters.

7. DISCUSSION

We have carried out a comparative evaluation of various state-of-the-art gradient matching methods. These methods are based on different statistical modeling and inference paradigms: non-parametric Bayesian statistics with Gaussian processes (INF, LB2, and LB10), hierarchical regularization using splines interpolation (RAM), splines-based smooth functional tempering (C&S), and penalized likelihood based on reproducing kernel Hilbert spaces (GON). We have also compared the antagonistic paradigms of Bayesian inference (INF) versus parallel tempering (LB2 and

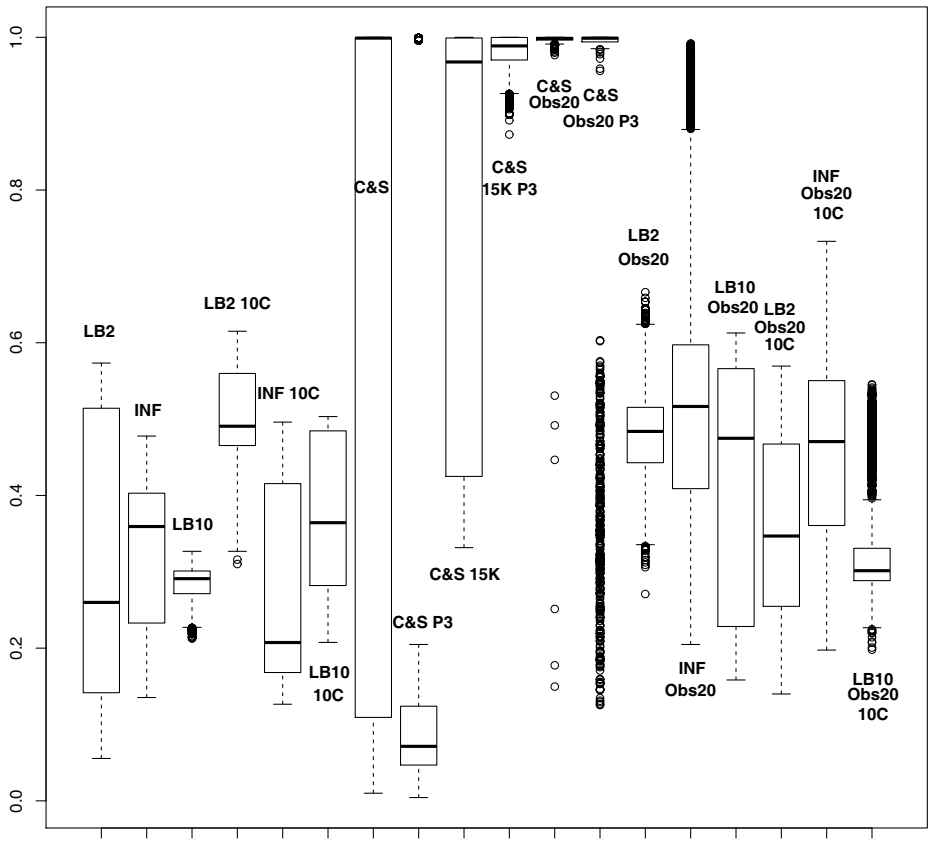

FIGURE 11 | Average posterior distributions of RMS values in function space, calculated on the residuals of the true signal (signal produced with true parameters and no observational error) minus the signal produced with the posterior sampled parameters. For viewing purposes, a squashing function has been applied to the RMS values, see equation (70). Larger values show poorer reconstructed signals. From left to right: LB2, INF, LB10, LB2 10C, INF 10C, LB10 10C, C&S, C&S P3, C&S 15K, C&S 15K P3, C&S Obs20, C&S Obs20 P3, LB2 Obs20, INF Obs20, LB10 Obs20, LB2 Obs20 10C, INF Obs20 10C, and LB10 Obs20 10C. For definitions, see **Tables 3** and **4**. For an explanation of the boxplot form, see the beginning of Section 6.

LB10) of slack parameters in the specific context of adaptive gradient matching. We discuss aspects of the methodology and empirical findings separately.

7.1. Methodology

The GON method, due to the RKHS framework, is fast to implement. This is an attractive property, since the main motivation for gradient matching methods was to obtain a computational speed-up over techniques that calculate the numerical solution to the ODEs. This method hinges on obtaining better estimates of the interpolant gradient by use of the EM algorithm, so great care needs to be had when implementing this step. Care also needs to be exercised in fitting the splines estimates of the species for terms of the ODEs that cannot be fit with the RKHS approach. This step is not optimized within the penalized likelihood of equation (61) and so poor splines estimates could deteriorate the results. The 3 level hierarchy of the RAM method, for first configuring the tuning parameters and then for performing parameter estimation, is sensible. Since gradient matching methods rely on a good estimate of the interpolant, focusing on the tuning parameters should achieve more robust parameter estimates of the ODEs. This 3 level approach, however, does increase the computational complexity and the RAM method does not achieve a good speed-up over the numerical solution methods. The C&S method's use of parallel tempering of both the likelihood and gradient mismatch parameter is intuitive, and allows the MCMC to explore with a reduced chance of getting stuck on local optima. This method uses B-splines interpolation, which can be difficult in practice to configure the tuning parameters for. The INF method has the advantage that the interpolant hyperparameters can be inferred from the data, since it uses Gaussian process interpolation. The inference approach to the gradient mismatch parameter, as opposed to tempering, however, could be overflexible and drive the parameter to values where the coupling between the gradients is too weak. The LB2 and LB10 methods avoid this by using a tempering scheme to drive the parameter to the theoretically correct value (0 corresponding to no mismatch between the gradients). However, currently, there is no way of optimizing the specific step size and increment schedule.

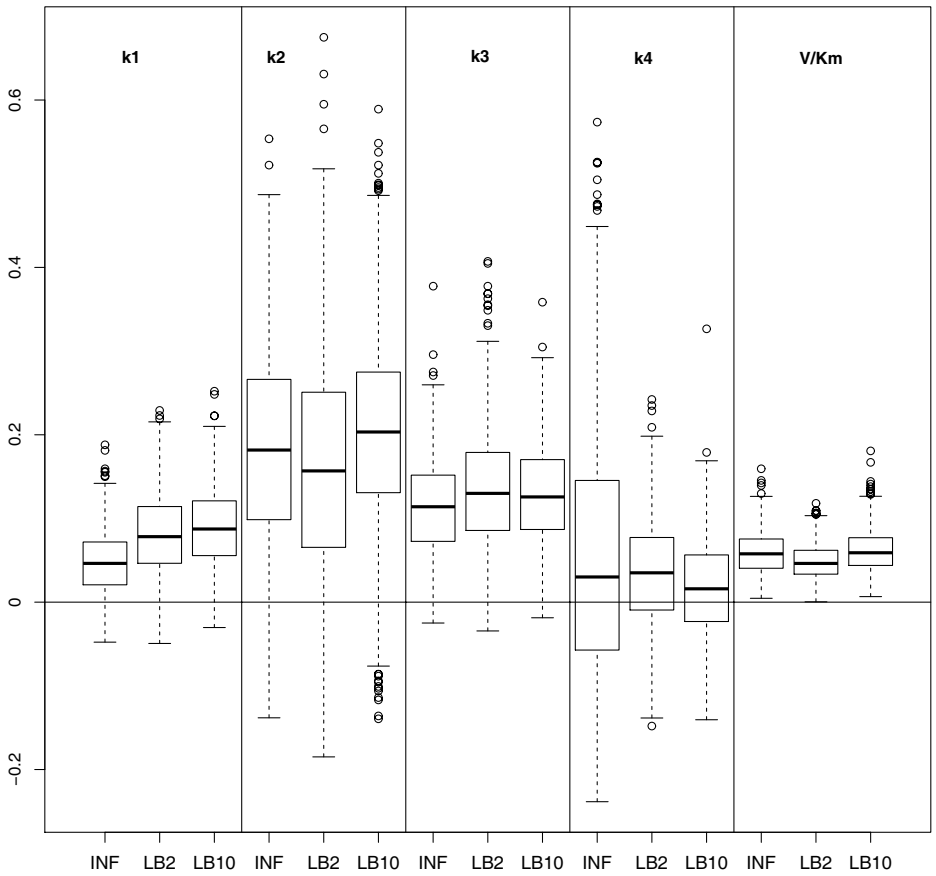

FIGURE 12 | Results from the dataset that showed the average RMS of the posterior parameter samples minus the true values for the INF, LB2, and LB10 methods. The posterior distributions are of the sampled parameters from equations (63)–(67) minus their true values. The horizontal line shows zero difference. For an explanation of the boxplot form, see the beginning of Section 6. Figure reconstructed from Macdonald and Husmeier (2015).

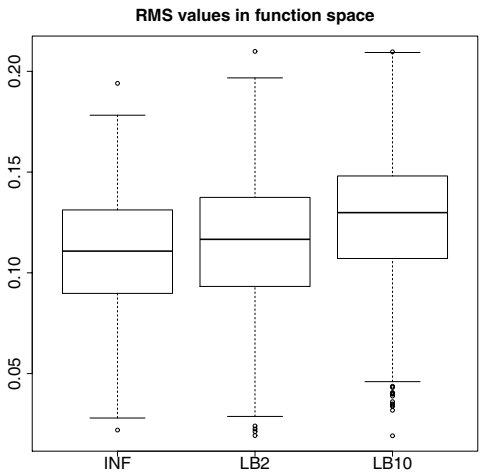

FIGURE 13 | Posterior distributions of RMS values for the ODE model in equations (63)–(67). The RMS values are calculated on the residuals of the true signal (signal produced with true parameters and no observational error) minus the signal produced from the sampled parameters. For an explanation of the boxplot form, see the beginning of Section 6.

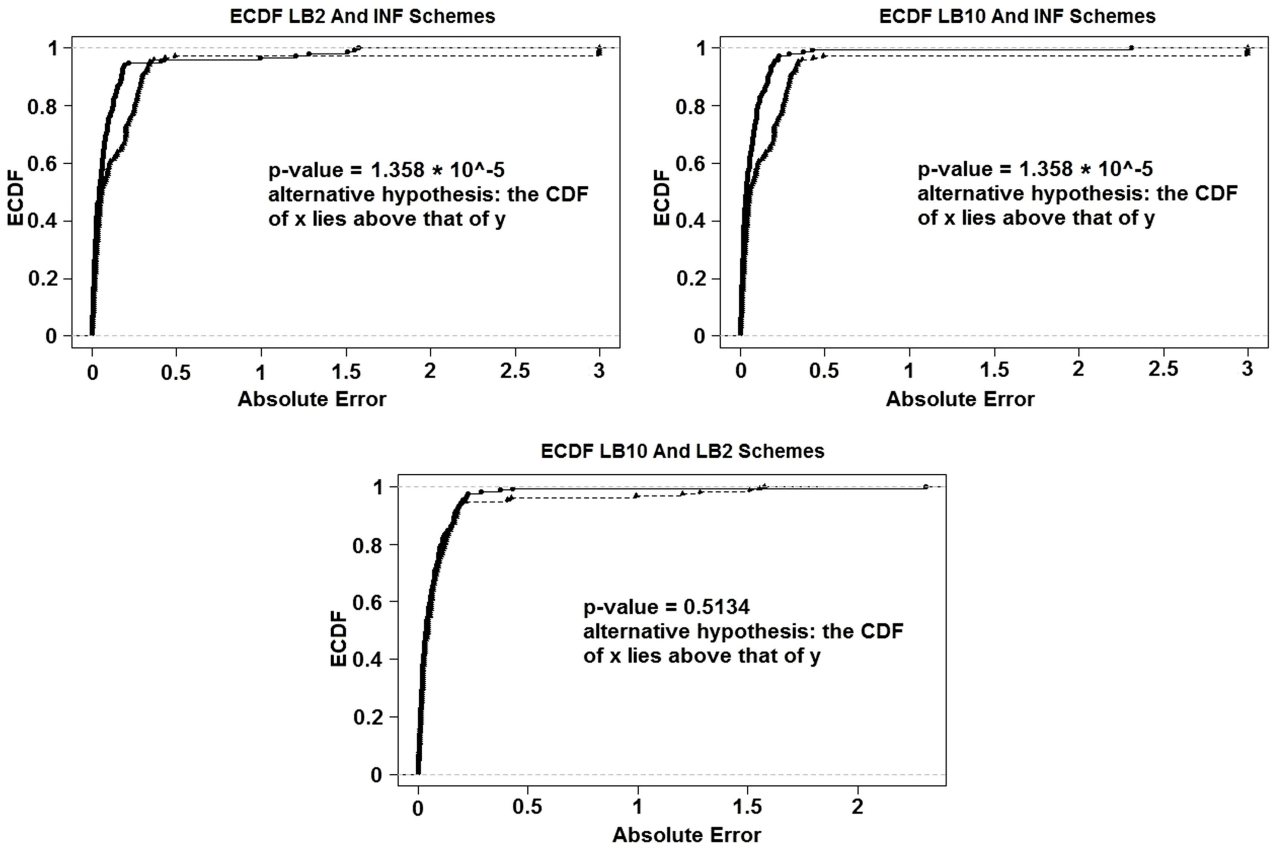

FIGURE 14 | ECDFs of the absolute errors of the parameter estimation. Top left – ECDFs for LB2 and INF, top right – ECDFs for LB10 and INF, and bottom – ECDFs for LB10 and LB2. Included are the p-values for 2-sample, 1-sided Kolmogorov–Smirnov tests. For definitions, see **Tables 3** and **4**. Figure reconstructed from Macdonald and Husmeier (2015).

7.2. Empirical Findings

The GON method produces estimates that are close to the true parameters in terms of absolute uncertainty. This, however, was for the case with small observational noise (Gaussian iid noise SD = 0.1), and it would be interesting to see how the parameter estimation accuracy is affected by the increase of noise. The RAM method performs worse than the rest of the methods it was compared to, across all parameters. The C&S method does well only in the one case, where the number of observations is very high (higher than what would be expected in these types of experiments) and the tuning parameters are finely adjusted (which in practice is very difficult and time-consuming). When the number of observations was reduced, all settings for this method deteriorated significantly. It is important also to note that the settings that we found to be optimal were slightly different than in the original paper, which highlights the sensitivity and lack of robustness of the splines-based method. The INF method shows a reasonable performance in terms of consistently producing results close to the true parameters, across all the set-ups we have examined. However, this technique's decrease in uncertainty is at the expense of bias. The LB2 and LB10 methods show the best performance across the set-ups. The parameter accuracy is unbiased across the different ODE models and the different settings of those models. The parallel tempering seems to be quite robust, performing similarly across the various set-ups. We have explored four different schedules for the parallel tempering scheme (as shown in **Table 1**). Overall, the performance of parallel tempering has been found to be reasonably robust with respect to a variation of the schedule.

ACKNOWLEDGMENTS

This work was supported by EPSRC (EP/L020319/1). We are grateful to Dr. Catherine Higham and Dr. Caroline Haig for helpful discussions and feedback on the manuscript.

REFERENCES

Barenco, M., Tomescu, D., Brewer, D., Callard, R., Stark, J., and Hubank, M. (2006). Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biol.* 7, R25. doi:10.1186/gb-2006-7-3-r25

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Singapore: Springer.

Calderhead, B., and Girolami, M. (2009). Estimating Bayes factors via thermodynamic integration and population MCMC. *Comput. Stat. Data Anal.* 53, 4028–4045. doi:10.1016/j.csda.2009.07.025

Calderhead, B., Girolami, M., and Lawrence, N. (2008). "Accelerating Bayesian inference over non-linear differential equations with Gaussian processes," in *Neural Information Processing Systems (NIPS)*, 22.

Campbell, D., and Steele, R. (2012). Smooth functional tempering for nonlinear differential equation models. *Comput. Stat.* 22, 429–443. doi:10.1007/s11222-011-9234-3

Dondelinger, F., Filippone, M., Rogers, S., and Husmeier, D. (2013). "ODE parameter inference using adaptive gradient matching with Gaussian processes," in *Journal of Machine Learning Research Workshop and Conference Proceedings (JMLR WCP): The 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Vol. 31, 216–228.

FitzHugh, R. (1961). Impulses and physiological states in models of nerve membrane. *Biophys. J.* 1, 445–466. doi:10.1016/S0006-3495(61)86902-6

Friel, N., and Pettitt, A. (2008). Marginal likelihood estimation via power posteriors. *J. R. Stat. Soc. Series B Stat. Methodol.* 70, 589–607. doi:10.1111/j.1467-9868.2007.00650.x

González, J., Vujačić, I., and Wit, E. (2013). Inferring latent gene regulatory network kinetics. *Stat. Appl. Genet. Mol. Biol.* 12, 109–127. doi:10.1515/sagmb-2012-0006

Hastie, T., Tibshirani, R., and Freidman, J. (2009). *The Elements of Statistical Learning*. New York: Springer.

Liang, H., and Wu, H. (2008). Parameter estimation for differential equation models using a framework of measurement error in regression models. *J. Am. Stat. Assoc.* 103, 1570–1583. doi:10.1198/016214508000000797

Macdonald, B., and Husmeier, D. (2015). "Computational inference in systems biology," in *Bioinformatics and Biomedical Engineering: Third International Conference, IWBBIO 2015. Proceedings, Part II*. Series: Lecture Notes in Computer Science (9044) eds. F. Ortuño and I. Rojas (Granada: Springer), 276–288.

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philos. Trans. R. Soc. Lond. A* 209, 415–446. doi:10.1098/rsta.1909.0016

Mohamed, L., Calderhead, B., Filippone, M., Christie, M., and Girolami, M. (2012). "Population MCMC methods for history matching and uncertainty quantification," in *Computational Geosciences* (Netherlands: Springer International publishing), 423–436.

Murphy, K. (2012). *Machine Learning: A Probabilistic Perspective*. Cambridge: MIT Press.

Murray, I., and Adams, R. (2010). "Slice sampling covariance hyperparameters of latent Gaussian models," in *Advances in Neural Information Processing Systems (NIPS)*, 23.

Nagumo, J., Arimoto, S., and Yoshizawa, S. (1962). An active pulse transmission line simulating a nerve axon. *Proc. Inst. Radio Eng.* 50, 2061–2070.

Ramsay, J., Hooker, G., Campbell, D., and Cao, J. (2007). Parameter estimation for differential equations: a generalized smoothing approach. *J. R. Stat. Soc. Series B Stat. Methodol.* 69, 741–796. doi:10.1111/j.1467-9868.2007.00610.x

Rasmussen, C., and Williams, C. (2006). *Gaussian Processes for Machine Learning*. Cambridge: MIT Press.

Schöelkopf, B., and Smola, A. (2002). *Support Vector Machines, Regularisation, Optimisation and Beyond*. Cambridge: MIT Press.

Vyshemirsky, V., and Girolami, M. (2008). Bayesian ranking of biochemical system models. *Bioinformatics* 24, 883–889. doi:10.1093/bioinformatics/btn475

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.