BIOINFORMATICS ORIGINAL PAPER Vol. 25 no. 10 2009, pages 1300–1306
doi:10.1093/bioinformatics/btp139

Systems biology

Gaussian process regression bootstrapping: exploring the effects of uncertainty in time course data

Paul D. W. Kirk* and Michael P. H. Stumpf*

Centre for Bioinformatics, Division of Molecular Biosciences, Imperial College London, London SW7 2AZ, UK

Received on November 21, 2008; revised on February 3, 2009; accepted on March 7, 2009

Advance Access publication March 16, 2009

Associate Editor: Limsoon Wong

ABSTRACT

Motivation: Although widely accepted that high-throughput biological data are typically highly noisy, the effects that this uncertainty has upon the conclusions we draw from these data are often overlooked. However, in order to assign any degree of confidence to our conclusions, we must quantify these effects. *Bootstrap* resampling is one method by which this may be achieved. Here, we present a parametric bootstrapping approach for time-course data, in which Gaussian process regression (GPR) is used to fit a probabilistic model from which replicates may then be drawn. This approach implicitly allows the time dependence of the data to be taken into account, and is applicable to a wide range of problems.

Results: We apply GPR bootstrapping to two datasets from the literature. In the first example, we show how the approach may be used to investigate the effects of data uncertainty upon the estimation of parameters in an ordinary differential equations (ODE) model of a cell signalling pathway. Although we find that the parameter estimates inferred from the original dataset are relatively robust to data uncertainty, we also identify a distinct second set of estimates. In the second example, we use our method to show that the topology of networks constructed from time-course gene expression data appears to be sensitive to data uncertainty, although there may be individual edges in the network that are robust in light of present data.

Availability: Matlab code for performing GPR bootstrapping is available from our web site:
http://www3.imperial.ac.uk/theoreticalsystemsbiology/data-software/

Contact: paul.kirk@imperial.ac.uk, m.stumpf@imperial.ac.uk

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

The use of data obtained from high-throughput technologies such as microarrays has become standard in systems biology. There are many ways in which these data are exploited, such as reverse engineering putative pathways and networks directly from the data (e.g. Lèbre, 2007; Opgen-Rhein and Strimmer, 2007), or inferring the values of unknown parameters in mechanistic models (e.g. Barenco *et al.*, 2006; Swameye *et al.*, 2003). The methods used to obtain the data are often subject to significant levels of measurement noise, and so we might expect repetitions of the experiments to yield quantitatively different datasets. However, the costs associated with high-throughput experiments usually mean that the number of technical replicates is restricted, and so it is difficult to quantify the effects of data uncertainty upon the inferences we draw. Clearly, if our aim is to attach biological meaning to our results (for example, by proposing putative pathways), then we need to have some degree of confidence that any conclusions we make are robust to the uncertainty in the data. That is, we need to be sure that what we infer (whether it be the rate constants of a biochemical reaction, the topology of a gene regulatory network or any other unknown quantity or reverse engineered model) is not specific to the particular noisy dataset that we happened to observe.

Bootstrapping is a well-known resampling method that may be used to assess properties (such as the standard error) of an inferred quantity or statistical estimator (Efron, 1979; Efron and Tibshirani, 1993). The process that generated the data is estimated by an approximating distribution from which samples may be drawn. Bootstrap datasets are then obtained from this distribution, and the statistical estimator is calculated for each. This induces a sampling distribution over the estimator, from which we may assess, for example, its variance amongst all of the bootstrap datasets. Previous biological applications of bootstrapping include, to name a few examples, placing confidence intervals on phylogenies (Felsenstein, 1985), assessing the reliability of conclusions drawn from clustering expression data (Kerr and Churchill, 2001), and constructing 'robust' estimates of gene networks (Imoto *et al.*, 2005).

We here consider a parametric bootstrap for time-course data in which the time-dependent process that generated the data is modelled using Gaussian process regression (GPR). In recent years, this Bayesian non-linear regression technique has grown in popularity, and has been applied in several systems biology contexts (Gao *et al.*, 2008; Lawrence *et al.*, 2007; Yuan, 2006). To our knowledge, GPR has not previously been used as a method for bootstrapping time-course data. However, it would seem to be ideally suited to this task, since it provides a method for fitting a plausible probabilistic model that captures the time dependence of the data, and from which it is easy to draw bootstrap samples.

We demonstrate GPR bootstrapping using two examples from the systems biology literature: estimating the parameters of an ordinary differential equation model for the STAT5 signalling pathway (Swameye *et al.*, 2003); and inference of gene regulatory networks in *Arabidopsis thaliana* (Smith *et al.*, 2004).

Below, we first provide an overview of GPR and how it may be used in general as a bootstrapping method (Section 2), and then we describe how the approach may be applied (Section 3).

*To whom correspondence should be addressed.

In Section 4, we summarize our findings for two examples, and discuss the implications in Section 5. We conclude by highlighting the importance of bootstrapping in general as a method for assessing the effects of data uncertainty.

2 APPROACH

GPR is a Bayesian non-linear regression method, which has been used to good effect in a number of studies (Gao *et al.*, 2008; Lawrence *et al.*, 2007; Yuan, 2006). Formally, a Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian (normal) distribution (Rasmussen and Williams, 2006). A GP is defined by a mean function and a covariance function, which specify the mean vectors and covariance matrices for each finite collection of the random variables. GP theory is discussed in more detail by MacKay (1998) and Rasmussen and Williams (2006), but for completeness and convenience we present an overview of standard GPR theory in addition to our own contribution of how it may be used to perform a parametric bootstrap.

2.1 Regression

In a regression problem, we are interested in elucidating the relationship between a collection of covariates or inputs $x_1, \ldots, x_p$, and a continuous dependent output variable, z. We assume that $x_1, \ldots, x_p, z$ are all real-valued, and we write the collection of covariates as a p-component column vector, $\mathbf{x} = [x_1, \ldots, x_p]^\top \in \mathbb{R}^p$. It is assumed that there is an unknown deterministic function, f, which wholly describes the relationship between z and $\mathbf{x}$, so that $z = f(\mathbf{x})$. Our aim is therefore to find the function, f.

In practice, the methods by which measurements of z are obtained introduce experimental noise. We hence define a random variable, y, to represent the experimentally observable version of z. We assume that y may be written as $y = z + \epsilon$, where ϵ is a noise term. For convenience, we also assume that $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and that ϵ is independent of $\mathbf{x}$. For the time being, we consider the case in which the variance, σ^2, is known, but shall return later to the problem of how it may be estimated.

To summarize, we have,

$$y(\mathbf{x}) = f(\mathbf{x}) + \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2). \tag{1}$$

One way to approach the regression problem is to impose a fixed parametric form on f [such as $f(\mathbf{x}) = \sum_{i=1}^{M} \beta_i \phi_i(\mathbf{x})$, where the β_i are parameters, the ϕ_i are a set of basis functions and $M \in \mathbb{N}$], and then to estimate its parameters from a set of experimentally obtained observations using methods such as ordinary least squares. An alternative is to recognize that the function, f, is unknown, and hence is itself a source of uncertainty; GPR provides us with a means by which to do this.

GPR belongs to a class of approaches known as *non-parametric Bayesian methods*. Such methods can be viewed broadly as providing probability models on function spaces (Müller and Quintana, 2004). Apart from GPs, the other well-known non-parametric Bayesian methods are those based on Dirichlet processes (DPs). These were introduced by Ferguson (1973) and Antoniak (1974), and provide a framework for the probabilistic modelling of unknown probability distributions. That is, rather than assuming that a given sample has been drawn from a probability distribution of known parametric form (but with unknown parameters), DP-based approaches model the uncertainty in the probability distribution itself. In contrast, GP approaches provide a framework for the probabilistic modelling of unknown functions rather than unknown distributions.

2.2 GP priors

In GPR, we assume a GP prior for $f(\mathbf{x})$ with mean function m and covariance function k. This means the following:

- For any input, $\mathbf{x}_* \in \mathbb{R}^p$, we regard the value taken by $f(\mathbf{x})$ at $\mathbf{x} = \mathbf{x}_*$ to be a random variable. The notation $f(\mathbf{x}_*)$ should now be understood to denote this random variable.
- Given a finite collection of covariate vectors, $\mathbf{x_1}, \ldots, \mathbf{x_n}$, the random variables $f(\mathbf{x_1}), \ldots, f(\mathbf{x_n})$ are assumed to be jointly distributed according to a multivariate Gaussian with mean $\mathbf{m} = [m(\mathbf{x_1}), \ldots, m(\mathbf{x_n})]^\top$ and covariance matrix $(K)_{ij} = k(\mathbf{x_i}, \mathbf{x_j})$.

We thus write $f(\mathbf{x}) \sim \mathcal{GP}(m, k)$.

Note that if we assume the regression model of Equation (1), the GP prior over $f(\mathbf{x})$ induces a GP prior over the observable outputs $y(\mathbf{x})$. That is, assuming Equation (1) and that $f(\mathbf{x}) \sim \mathcal{GP}(m, k)$, it follows that,

$$y(\mathbf{x}) \sim \mathcal{GP}(m, l), \tag{2}$$

where $l(\mathbf{x_i}, \mathbf{x_j}) = k(\mathbf{x_i}, \mathbf{x_j}) + \sigma^2 \delta(\mathbf{x_i}, \mathbf{x_j})$. Here, $\delta(\mathbf{x_i}, \mathbf{x_j})$ is the standard Kronecker delta function.

2.3 From prior to posterior

We now suppose that, having assumed a GP prior $f(\mathbf{x}) \sim \mathcal{GP}(m, k)$, we proceed to obtain a set of output measurements $y_1, \ldots, y_r$ at the covariate vectors $\mathbf{x_1}, \ldots, \mathbf{x_r}$. We are interested in determining how we may update our GP prior in light of these observed data. We show below that, given any finite collection $\mathbf{x_1}^*, \ldots, \mathbf{x_s}^*$ of covariate vectors, the joint conditional probability of the function values $f(\mathbf{x_1}^*), \ldots, f(\mathbf{x_s}^*)$ given the observations is again described by a multivariate normal. We hence obtain a GP posterior over $f(\mathbf{x})$.

We view $y_1, \ldots, y_r$ as realizations of the random variable $y(\mathbf{x})$ at the inputs $\mathbf{x}$. We know from Equation (2) that,

$$\left[y(\mathbf{x_1}), \ldots, y(\mathbf{x_r})\right]^\top \sim \mathcal{N}(\mathbf{m}_o, K_o), \tag{3}$$

where $\mathbf{m}_o = [m(\mathbf{x_1}), \ldots, m(\mathbf{x_r})]^\top$ and $\left(K_o\right)_{ij} = k(\mathbf{x_i}, \mathbf{x_j}) + \sigma^2 \delta(\mathbf{x_i}, \mathbf{x_j})$. For notational brevity, we henceforth write $[\mathbf{y}(\mathbf{x})]^\top$ to mean $\left[y(\mathbf{x_1}), \ldots, y(\mathbf{x_r})\right]^\top$.

Let $\mathbf{x_1}^*, \ldots, \mathbf{x_s}^*$ be another finite collection of covariate vectors. From our assumption of a GP prior over f, together with Equation (3), it is straightforward to see that,

$$\left[\mathbf{y}(\mathbf{x}), f(\mathbf{x_1}^*), \ldots, f(\mathbf{x_s}^*)\right]^\top \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{m}_o \\ \mathbf{m}_*\end{bmatrix}, \begin{pmatrix}K_o & K_* \\ K_*^\top & K_{**}\end{pmatrix}\right), \tag{4}$$

where, $\mathbf{m}_* = [m(\mathbf{x_1}^*), \ldots, m(\mathbf{x_s}^*)]^\top$, $\left(K_{**}\right)_{ij} = k(\mathbf{x_i}^*, \mathbf{x_j}^*)$ and $\left(K_*\right)_{ij} = k(\mathbf{x_i}, \mathbf{x_j}^*)$.

From Equations (3) and (4), and using standard properties of Gaussian distributions (von Mises, 1964), it follows that the function values $f(\mathbf{x_1}^*), \ldots, f(\mathbf{x_s}^*)$ conditioned on the observed outputs $\mathbf{y}$ are also jointly distributed according to a multivariate normal.

Specifically,

$$\left[f(\mathbf{x_1}^*),\ldots,f(\mathbf{x_s}^*)\right]^\top \Big| \left([\mathbf{y}(\mathbf{x})]^\top = \mathbf{y}\right) \sim \mathcal{N}(\mathbf{m}_{\text{post}}, K_{\text{post}}), \tag{5}$$

where $\mathbf{y} = [y_1,\ldots,y_r]^\top$, and,

$$\mathbf{m}_{\text{post}} = \mathbf{m}_* + K_*^\top \left(K_o + \sigma^2 I_r\right)^{-1} (\mathbf{y} - \mathbf{m}_o), \tag{6}$$

$$K_{\text{post}} = K_{**} - K_*^\top \left(K_o + \sigma^2 I_r\right)^{-1} K_*. \tag{7}$$

Here, I_r is the $r \times r$ identity matrix.

Since Equation (5) is true for any $s \in \mathbb{N}$, it follows that the function outputs, $f(\mathbf{x})$, conditioned on the observations, $\mathbf{y}$, define a GP, which is referred to as the GP posterior.

2.4 Using the GP posterior

Equation (5) provides the joint posterior distribution of the function values $f(\mathbf{x_1}^*),\ldots,f(\mathbf{x_s}^*)$, given the GP prior and the observations, $\mathbf{y}$. Since the mean of a Gaussian distribution is also its mode, the maximum *a posteriori* prediction of $[f(\mathbf{x_1}^*),\ldots,f(\mathbf{x_s}^*)]^\top$ is simply the mean vector $\mathbf{m}_{\text{post}}$. Thus, GPR allows the prediction of $f(\mathbf{x})$ at any finite collection of covariate vectors, $\mathbf{x_1}^*,\ldots,\mathbf{x_s}^*$. The covariance matrix, K_{post}, describes the variability of the distribution about the mean, and hence may be used to place confidence intervals around this prediction. Figure 1A illustrates the use of GPR to make predictions and specify confidence intervals.

In this article, we are concerned not only with fitting a regressor to the dataset, but also with sampling from the regression model in order to obtain bootstrap datasets. This is similar to the work of Kerr and Churchill (2001), who also generate bootstrap samples by first fitting a model to a set of time-course data (in their case, an ANOVA model). The advantages of GPR are its non-linearity, that it implicitly allows us to model the uncertainty in the underlying function, f, and that it is relatively easy to apply. Generating samples from our GP regressor is also fairly simple. We know that the joint posterior distribution of any finite collection, $f(\mathbf{x_1}^*),\ldots,f(\mathbf{x_s}^*)$, is a multivariate normal [as given in Equation (5)], and hence we may simulate samples using standard methods for such distributions (Press *et al.*, 2007).

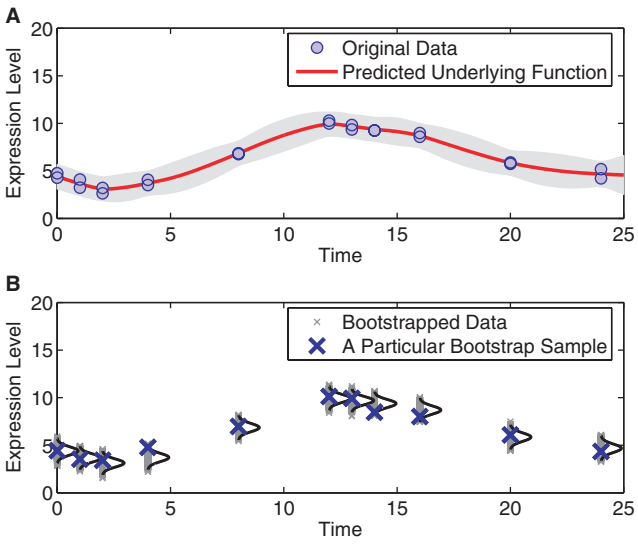

Fig. 1. Using a GP regressor to fit gene expression time-course data and draw bootstrap samples [data taken from the 'arth800' dataset of the R package 'GeneNet' (Schäfer *et al.*, 2006)]. (**A**) The red line shows the mean of the posterior process, while the grey region is a 99% credible interval around this mean. (**B**) The marginal distribution over the observable output values at each time point is univariate Gaussian, centred at the mean of the posterior process. The crosses represent samples drawn from the posterior, with blue crosses used to highlight one particular draw.

Since we are concerned with the generation of plausible datasets, rather than just plausible samples of the underlying function values, it follows that we are actually interested in $y(\mathbf{x})$ rather than $f(\mathbf{x})$. However, if we can sample function outputs, $f(\mathbf{x})$, and if we know (or can estimate) the variance σ^2, then we can use Equation (1) in order to obtain samples of $y(\mathbf{x})$. Thus, in practice, we proceed by first sampling $[f(\mathbf{x_1}^*),\ldots,f(\mathbf{x_s}^*)]^\top$ from the multivariate normal described by Equations (5), (6) and (7), and then adding Gaussian noise sampled from $\mathcal{N}(0,\sigma^2 I_s)$.

In this study, we generate bootstrap samples at the same points as those at which the data were observed (i.e. we choose $s = r$ and set $\mathbf{x_1}^* = \mathbf{x_1},\ldots,\mathbf{x_s}^* = \mathbf{x_r}$). Figure 1B provides an example of a number of bootstrap samples obtained using a GP regressor fitted to a gene expression time course.

2.5 The mean and covariance functions

In order to specify a GP prior, it is clearly necessary to provide a mean function, m, and a covariance function, k. The covariance function is the more important of these, as it describes how we believe the value of the function outputs, $f(\mathbf{x})$, covary with one another, and hence allows us to express our beliefs about fundamental properties of f, such as how rapidly it changes. For the sake of simplicity and parsimony, the mean function is often chosen to be zero, and this is the approach we adopt here. This does not present a serious limitation: as we can see from the regressor in Figure 1A, the mean of the posterior process (represented by the red line) is certainly not constrained to be zero, and we are able to obtain a good fit to the data. Of course, other mean functions may be chosen to express stronger prior beliefs about the underlying function. There are many possible choices for the covariance function, k, and we here consider two of the more popular options, the *squared exponential* covariance function,

$$k_{\text{SE}}(\mathbf{x_i},\mathbf{x_j}) = \sigma_g^2 \exp\left(-\frac{1}{2l_1}|\mathbf{x_i} - \mathbf{x_j}|^2\right), \tag{8}$$

where $|\cdot|$ denotes the Euclidean distance; and a standard *Matérn* covariance function,

$$k_M(\mathbf{x_i},\mathbf{x_j}) = \sigma_f^2 \left(1 + \frac{\sqrt{3}|\mathbf{x_i} - \mathbf{x_j}|}{l_2}\right) \exp\left(-\frac{\sqrt{3}|\mathbf{x_i} - \mathbf{x_j}|}{l_2}\right). \tag{9}$$

Here, the constants σ_g, σ_f, l_1 and l_2 are *hyperparameters*. Although its smoothness properties have been criticized as unrealistic (Stein, 1999), the squared exponential covariance function remains the most frequently used 'default' choice for GPR, largely because of its simplicity. There are many examples of covariance function (Rasmussen and Williams, 2006, ch. 4), which allow the GP prior to be tailored to specific scenarios. In this article, we employ k_{SE} and k_M, as they are simple, yet sufficiently flexible to allow a good fit to the data.

The hyperparameters of the covariance function provide us with another means to encode our prior beliefs about the nature of f. We can see, for example, that if l_1 (or l_2) is very large, then $f(\mathbf{x_i})$ and $f(\mathbf{x_j})$ will only tend to vary together if $|\mathbf{x_i}-\mathbf{x_j}|$ is small: the value of the function at $\mathbf{x_i}$ will only affect the value at $\mathbf{x_j}$ if $\mathbf{x_i}$ and $\mathbf{x_j}$ are close together. Ideally, we would either use prior knowledge to specify the hyperparameters, or adopt a fully Bayesian approach and integrate them out. However, we are rarely able to express our prior beliefs so precisely, and while a full Bayesian approach is possible (using, for example, Markov chain Monte Carlo (MCMC)), the associated computational expense is often undesirable. This is certainly the case here: in the example presented in Section 3.2 (which we expect may represent a typical application), we are required to fit regressors to 800 gene expression time-course datasets, so we wish to minimize the costs of fitting the GPR model. An alternative and computationally cheaper method is to estimate the hyperparameters in order to maximize the (log) likelihood of the observed data. We also use this approach to estimate the variance of the noise term, σ^2, in Equation (1). From Equation (3) and the definition of a multivariate normal, the likelihood of $\mathbf{y}$ is given by,

$$p(\mathbf{y}(\mathbf{x})=\mathbf{y}|\theta,\sigma^2)=\frac{\exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{m_0})^\top K_o(\theta,\sigma^2)^{-1}(\mathbf{y}-\mathbf{m_0})\right)}{\sqrt{(2\pi)^r \det\left(K_o\left(\theta,\sigma^2\right)\right)}},$$

where θ is the vector of the covariance function's hyperparameters, $\det(\cdot)$ denotes the determinant, and we write $K_o(\theta,\sigma^2)$ to make explicit the dependence of K_o on the hyperparameters. Taking logs and removing constant terms, we deduce that the maximum likelihood values for θ and σ^2 are given by,

$$\hat{\theta},\hat{\sigma}^2=\underset{\theta,\sigma^2}{\operatorname{argmax}}\left\{-\frac{1}{2}\log\left(\det\left(K_o\left(\theta,\sigma^2\right)\right)\right)-\frac{1}{2}(\mathbf{y}-\mathbf{m_0})^\top K_o\left(\theta,\sigma^2\right)^{-1}(\mathbf{y}-\mathbf{m_0})\right\}.$$

This optimization can be approached using standard methods for optimization (Press *et al.*, 2007), such as the Nelder–Mead simplex method or gradient descent.

3 APPLICATIONS

In order to demonstrate the potential applications of the GPR bootstrap, we consider two examples from the literature: estimating the parameters of a model of the STAT 5 signalling pathway, and inferring a gene network.

3.1 Parametric ODE modelling of signalling pathways

The JAK-STAT pathway is a well-studied signalling pathway that describes a mechanism by which signals carried by cytokines may be transduced to the cell nucleus via STAT activation, dimerization and relocation to the nucleus (Aaronson and Horvath, 2002; Horvath, 2000). Swameye *et al.* (2003) suggested a number of parametric ODE models to describe the JAK2-STAT5 signalling pathway, the parameters of which were estimated from experimental data. We consider one of the proposed models (taken from Swameye *et al.*, 2003, Supplementary Material), and—using data from the original experiments—apply our GPR bootstrapping approach in order to assign confidence intervals to the parameter estimates.

3.1.1 The Model The model we consider is as follows,

$$\frac{\mathrm{d}v_1}{\mathrm{d}t}=-r_1v_1D+2r_4v_4 \qquad \frac{\mathrm{d}v_2}{\mathrm{d}t}=r_1v_1D-v_2^2 \qquad (10)$$
$$\frac{\mathrm{d}v_3}{\mathrm{d}t}=-r_3v_3+0.5v_2^2 \qquad \frac{\mathrm{d}v_4}{\mathrm{d}t}=r_3v_3-r_4v_4.$$

Here, v_1, v_2 and v_3 represent the concentrations of (respectively) unphosphorylated STAT5, phosphorylated monomeric STAT5 and phosphorylated dimeric STAT5 in the cytoplasm. The variable v_4 denotes the concentration of STAT5 in the nucleus, and D is an experimentally determined quantity (which varies over time) related to the amount of Epo-induced phosphorylation of the EpoR (Swameye *et al.*, 2003). The r_i's are parameters (see Swameye *et al.*, 2003; Supplementary Material). The initial values of v_2, v_3 and v_4 at time $t=0$ are assumed to be zero (since it is supposed that all STAT5 in the cell is initially cytoplasmic and unphosphorylated), while the initial concentration of unphosphorylated cytoplasmic STAT5, $v_1(t=0)$, is treated as an unknown parameter.

The quantities v_1, v_2, v_3 and v_4 were not measured individually. Instead, the amount of phosphorylated STAT5 in the cytoplasm, y_1, and the total amount of cytoplasmic STAT5 (phosphorylated and unphosphorylated), y_2, were recorded. These can be written in terms of the v_i's as follows,

$$y_1=r_5(v_2+2v_3)$$
$$y_2=r_6(v_1+v_2+2v_3),$$

where r_5 and r_6 are two unknown scaling parameters, which must also be estimated. In total, there are thus six unknown parameters in this model [r_1, r_3, r_4, r_5, r_6 and $v_1(0)$].

3.1.2 GPR bootstrapping and parameter estimation Swameye *et al.* (2003) measured y_1 and y_2 at a number of discrete time points in order to obtain several sets of experimental data. We focus on just one of these datasets (the 'DATA1_hall' set, available from the original authors at http://webber.physik.uni-freiburg.de/∼jeti/), which we use together with our GPR bootstrapping approach in order to obtain 1500 bootstrapped datasets. To define our GP prior, we choose a zero mean function and the squared exponential function of Equation (8).

In order to learn the hyperparameters and fit the GP regressor to the dataset, we make use of the `gpml` suite of Matlab functions accompanying Rasmussen and Williams (2006), available from http://www.gaussianprocess.org/gpml/.

For each of our bootstrapped datasets, we estimate the unknown parameters of the ODE system presented in Equation (10) using the stochastic ranking evolutionary strategy (SRES) of Runarsson and Yao (2000), as implemented in the libSRES C library (Ji and Xu, 2006). This allows us to find the parameter values which minimize the sum of squared differences between the data and the predictions made by the ODE model. This optimization problem is susceptible to the usual difficulty of becoming stuck in a local minimum. The evolutionary nature of SRES goes some of the way toward mitigating this difficulty, but to reduce the impact of becoming stuck in a local minimum yet further, we also run the algorithm for a large number of iterations and rerun eight times for each dataset (taking as our final estimate the 'best' amongst these eight runs). Before considering the bootstrapped data, we use SRES to estimate parameter values from the original dataset. The values so obtained are: $v_1(0)=0.996$, $r_1=2.43$, $r_3=0.256$, $r_4=0.303$, $r_5=1.27$ and $r_6=0.944$. For this 'optimal' set of parameters the model provides a reasonable fit to the observed data which is comparable with the fit obtained in the original paper (Swameye *et al.*, 2003). The aim of our bootstrapping approach is to determine whether or not these parameter estimates are robust to the uncertainty in the data.

3.2 Gene network inference

When considering how our GPR bootstrapping approach may be applied in order to investigate the effects of data uncertainty on the reverse engineering

of gene regulatory networks we consider only relevance networks (Butte *et al.*, 2000) and graphical Gaussian models (GGMs). However, our method could just as easily be applied in order to assess the effects of data uncertainty on network inference methods (such as Lèbre, 2007) more generally. We consider temporal expression data for the 800 *A.thaliana* genes from (Smith *et al.*, 2004) which are provided in the 'arth800' dataset of the R package 'GeneNet' (Schäfer *et al.*, 2006).

3.2.1 Gene relevance networks Butte *et al.* (2000) introduced the idea of a *gene relevance network*—a type of graphical model in which vertices represent genes and in which we draw an edge between genes g_1 and g_2 if and only if the expressions of g_1 and g_2 are correlated. Thus, relevance networks provide us with a means to represent (linear) dependencies between genes. Correlations are calculated between genes in a pairwise fashion; it is decided whether or not to draw an edge between g_1 and g_2 without reference to the presence or absence of edges between any other genes. To determine whether or not to place an edge between genes g_1 and g_2, we first calculate the (in our case, Pearson) correlation between their gene expression time courses, square this value to obtain a score s, and then place an edge if $s > r$ for some prespecified threshold value r.

3.2.2 Graphical Gaussian models GGMs are used to represent dependencies between genes that have been detected by *partial* correlations. In contrast to relevance networks (where a missing edge between two genes indicates marginal independence), the absence of an edge between genes g_1 and g_2 in a GGM means that g_1 and g_2 are *conditionally* independent. We use the R package 'GeneNet' in order to infer GGMs from time-course gene expression data (according to Opgen-Rhein and Strimmer, 2007), and make use of the package's capability to calculate an empirical posterior probability, $p_e(g_1, g_2)$ (Schäfer and Strimmer, 2005), for the existence of the edge between g_1 and g_2. If $p_e(g_1, g_2) > \tau$, where τ is some prespecified threshold (cut-off) value for the probability, then an edge is drawn between g_1 and g_2.

3.2.3 Bootstrapping the data We apply our GPR bootstrapping approach to the *A.thaliana* data. This dataset comprises measurements taken for 800 genes at 11 times, with two measurements at each time point (Fig. 1A illustrates the data for one of the genes). We proceed as in the previous example, but this time make use of the following covariance function,

$$k(t_i, t_j) = k_{\mathrm{SE}}(t_i, t_j) + k_M(t_i, t_j), \tag{11}$$

where k_{SE} and k_{M} are as previously described. Using the method of Section 2 we obtain 1000 bootstrap datasets, each one consisting of two measurements at 11 time points for 800 genes.

4 RESULTS

4.1 Parametric ODE modelling of signalling pathways

For each of our 1500 bootstrapped datasets, we find the 'optimal' set of parameters using SRES. This induces a joint sampling distribution over the optimal parameters for the ODE model, whose marginals are represented by the histograms in Figure 2.

Note that the joint sampling distribution is conceptually very different to the joint posterior parameter distribution that might be sought using a Bayesian approach: in the former, we find a single parameter estimate for each of a large number of sampled datasets, whereas in the latter we first specify a prior distribution over the parameters and then seek to update this in light of observed data.

Figure 2 shows that the marginal sampling distributions are generally quite narrow. This can be quantified by calculating the coefficient of variation, c_v, for each of the parameters, $c_v(v_1(0)) = 0.0338$, $c_v(r_1) = 0.175$, $c_v(r_3) = 1.77$, $c_v(r_4) = 0.214$,

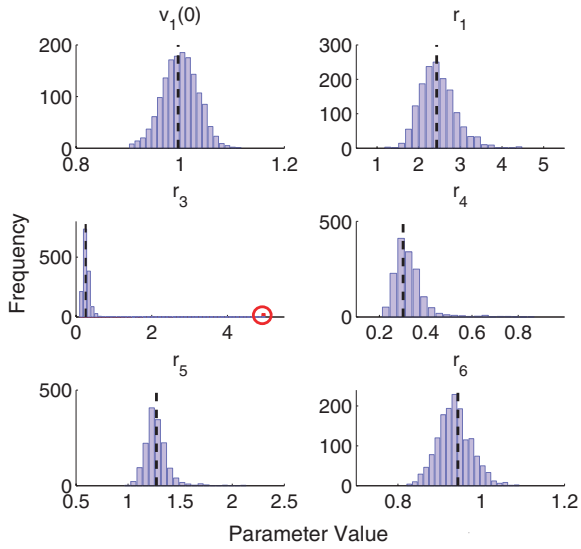

Fig. 2. Histograms showing the marginal sampling distributions over the optimal parameter values. Although these distributions are generally quite narrow and centred about the parameter estimates obtained from the original dataset (vertical black dashed lines), note that for r_3 there is a small amount of probability mass located at $r_3 \approx 5$ (shown as a red bar and ringed by a red circle).

$c_v(r_5) = 0.0914$ and $c_v(r_6) = 0.0434$. The coefficient of variation for r_3 is significantly greater than for the other parameters. This is due to the influence of bootstrap samples for which the optimal estimate for r_3 was ~ 5 (see the red bar in Fig. 2). Indeed, across all of the bootstrap samples, there appear to be two distinct sets of estimated optimal parameter values. The first (much larger) set comprises estimates centred around the values obtained from the original dataset. The second set comprises estimates for which $r_3 \approx 5$, and contains only 28 elements. Although obtained for just a small number of the bootstrap samples ($\approx 2\%$), the parameter estimates in this second set still provide a good fit to the original data (see Supplementary Fig. 1). To quantify this, the average mean square error (MSE) obtained using parameter estimates from the second set was 0.10 for the fit to the y_1 values, and 0.031 for the fit to the y_2 values. By comparison, the MSE obtained using the parameter values estimated from the original data was 0.026 for the fit to the y_1 values, and 0.0043 for the y_2 values. This suggests that parameter estimates from the second set provide (on average) a slightly worse fit overall, but do a marginally better job of fitting y_2 than those derived from the original dataset.

4.2 Gene network inference

We start by considering the inferred GGMs. Let $N_O(\tau)$ denote the network inferred from the original dataset using threshold τ, and similarly let $N_B^{(i)}(\tau)$ be the network inferred from the i-th bootstrap dataset. To assess how similar the bootstrapped networks are to $N_O(\tau)$, we calculate the proportion, $\rho^{(i)}(\tau)$, of edges in the original network that also appear in $N_B^{(i)}(\tau)$. We hence obtain a sampling distribution over $\rho^{(i)}(\tau)$ for a given τ. In addition to considering the data sampled using GPR bootstrapping, we also performed a non-parametric bootstrap of the data, and calculated a

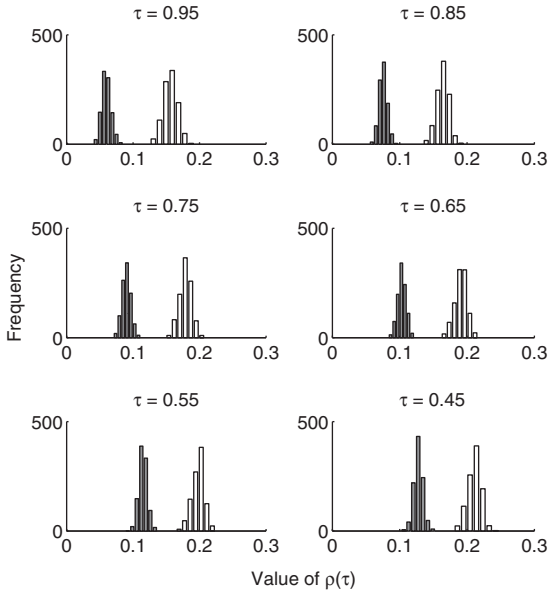

Fig. 3. Plots showing the sampling distributions over $\rho^{(i)}(\tau)$ for different values of τ. Black: GPR bootstrap. White: non-parametric bootstrap.

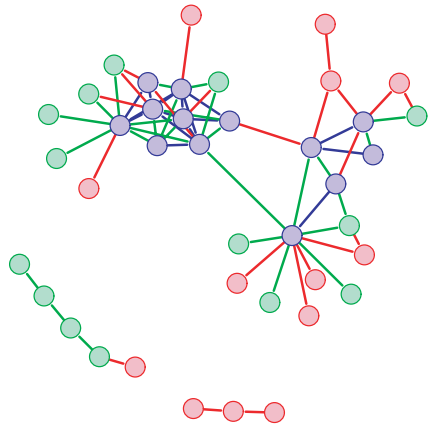

Fig. 4. A high confidence subnetwork constructed from the *A.Thaliana* data set. Edges drawn in blue appear in 100% of the bootstrap data samples; green edges appear in 99%; and red edges appear in 98%. Vertices have the same colour as whichever of their edges appears in the largest number of bootstrap samples.

sampling distribution over $\rho^{(i)}(\tau)$ based upon 1000 (non-parametric) bootstrap datasets. Histograms describing the sampling distributions obtained for different values of τ are presented in Figure 3.

In each case, the sampling distribution is approximately normal. For smaller values of τ, the mean of the sampling distribution over $\rho^{(i)}(\tau)$ is greater than for larger values. This is unsurprising: as τ gets smaller, the value of the empirical posterior probability required for an edge gets lower (until at the extreme case, $\tau = 0$, edges are drawn between *all* vertices, regardless of the data). Thus, for smaller values of τ, the sensitivity to the data is reduced. This has the effect of making the inferred network more robust to perturbations in the data, but unfortunately also makes the network increasingly meaningless. For more meaningful values of τ (say, of around 0.85 or above), the degree of similarity between the bootstrapped networks and $N_O(\tau)$ [as measured by the mean value of $\rho^{(i)}(\tau)$] is disappointingly low. This suggests that the original inferred GGM is highly sensitive to uncertainty in the data.

We repeated the above analysis for relevance networks, and obtained similar results (see Supplementary Material). To summarize, the mean values of $\rho^{(i)}(r)$ for r values of 0.95, 0.85, 0.75, 0.65, 0.55 and 0.45 were (respectively) 0.062, 0.18, 0.29, 0.42, 0.51 and 0.61. Yet again, these values are low, indicating that the topology of relevance networks is sensitive to uncertainty in the data (note that r and τ are not directly comparable, so it is difficult to compare the relative tolerance to data uncertainty of relevance networks and GGMs). Although the overall topology of the network seems to be sensitive to data uncertainty, there are individual edges that demonstrate a much higher degree of tolerance. For example, taking r to be 0.85, we may look for edges that appear in 100% of the networks obtained from the bootstrapped datasets. If we do this, then we find 16 edges connecting 13 vertices, as shown in Figure 4. We can use this approach more generally to construct networks that have a required level of tolerance to data uncertainty, omitting any edges that do not appear in at least q% of the bootstrap samples. In this way, we can construct 'high confidence' relevance networks.

5 DISCUSSION

Our results highlight the necessity of accounting for data uncertainty when trying to draw conclusions from experimentally obtained data. In the case of the parametric ODE model of the JAK2-STAT5 signalling pathway, we showed that in addition to the parameter estimates obtained from the original dataset, there is a second set of plausible estimates which (had stochastic effects provided us with a slightly different dataset) we may well have concluded were the 'correct' values. We believe that the presence of the distinct second set of plausible parameter estimates indicates that the error function (i.e. the sum of squared differences between the data and the model predictions) that was minimized in order to fit the model to the original data possesses a second (local) minimum. As well as demonstrating the importance of taking into account the noise in experimental datasets, our results could also be viewed as an endorsement for Bayesian methods, which do not seek to identify a single optimal parameter set, but instead approximate the whole posterior parameter distribution.

The results from our network inference investigation perhaps provide an even better illustration of the effect that data uncertainty can have upon inference. Our approach demonstrates that the inference of an edge between two vertices is highly sensitive to the level of noise in the data, and hence it is likely that the false positive rate for each individual edge is high. We also showed how GPR bootstrapping may be used to construct 'high confidence' relevance networks for which we would expect the false positive rate to be lower.

6 CONCLUSION

Determining the effects of uncertainty in experimental data is imperative if we are to have any degree of confidence in the conclusions that we draw or the models that we reverse engineer from biological data. GPR bootstrapping is a widely applicable

and easily implemented approach that allows us to investigate and quantify these effects. We have illustrated the use of GPR bootstrapping using two examples, and discussed the impact that data uncertainty has upon inference. Although we have here concentrated on time-course data, our approach could easily be applied in situations where the independent variable is something other than time. Given the current levels of noise in post-genomic data, approaches such as GPR bootstrapping are vital in order to allow us to make the most of currently available information and to provide us with a means to assess the conclusions we draw.

Funding: Wellcome Trust (080713/Z/06/Z).

Conflict of Interest: none declared.

REFERENCES

Aaronson,D.S. and Horvath,C.M. (2002) A road map for those who don't know JAK-STAT. *Science*, **296**, 1653–1655.

Antoniak,C.E. (1974) Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Ann. Stat.*, **2**, 1152–1174.

Barenco,M. *et al.* (2006) Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biol*., **7**, R25.

Butte,A.J. *et al.* (2000) Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proc. Natl Acad. Sci. USA*, **97**, 12182–12186.

Efron, B. (1979) Bootstrap methods: Another look at the jackknife. *Ann. Stat.*, **7**, 1–26.

Efron,B. and Tibshirani,R.J. (1993) *An Introduction to the Bootstrap*. Chapman and Hall, New York.

Felsenstein,J. (1985) Confidence limits on phylogenies: an approach using the bootstrap. *Evolution*, **39**, 783–791.

Ferguson,T.S. (1973) A Bayesian analysis of some nonparametric problems. *Ann. Stat.*, **1**, 209–230.

Gao,P. *et al.* (2008) Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities. *Bioinformatics*, **24**, i70–i75.

Horvath,C.M. (2000) STAT proteins and transcriptional responses to extracellular signals. *Trends Biochem. Sci.*, **25**, 496–502.

Imoto,S. *et al.* (2005) Residual bootstrapping and median filtering for robust estimation of gene networks from microarray data. In *Computational Methods in Systems Biology*, Vol. 3082, Springer-Verlag, Berlin, pp. 149–160.

Ji,X. and Xu,Y. (2006) libsres: a c library for stochastic ranking evolution strategy for parameter estimation. *Bioinformatics*, **22**, 124–126.

Kerr,M.K. and Churchill,G.A. (2001) Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments. *Proc. Natl Acad. Sci. USA*, **98**, 8961–8965.

Lawrence,N.D. *et al.* (2007) Modelling transcriptional regulation using Gaussian processes. In Schölkopf,B. *et al.* (eds) *Advances in Neural Information Processing Systems 19.* MIT Press, Cambridge, MA, pp. 785–792.

Lèbre,S. (2007) Inferring dynamic genetic networks with low order independencies. *arXiv.org*, **arXiv:0704.2551v4**, 1–36.

MacKay,D.J.C. (1998) Introduction to Gaussian processes. In Bishop,C.M. (ed.) *Neural Networks and Machine Learning*, NATO ASI Series. Springer, Berlin, pp. 133–165.

Müller,P. and Quintana,F.A. (2004) Nonparametric Bayesian data analysis. *Stat. Sci.*, **19**, 95–110.

Opgen-Rhein,R. and Strimmer,K. (2007) From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data. *BMC Syst. Biol.*, **1**, 37.

Press,W.H. *et al.* (2007) *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, New York.

Rasmussen,C.E. and Williams,C.K.I. (2006) *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, Cambridge, MA.

Runarsson,T.P. and Yao,X. (2000) Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.*, **4**, 284–294.

Schäfer, J. and Strimmer, K. (2005) An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, **21**, 754–764.

Schäfer,J. *et al.* (2006) Reverse engineering genetic networks using the GeneNet package. *R News*, **6**, 50–53.

Smith,S.M. *et al.* (2004) Diurnal changes in the transcriptome encoding enzymes of starch metabolism provide evidence for both transcriptional and posttranscriptional regulation of starch metabolism in Arabidopsis leaves. *Plant Physiol.*, **136**, 2687–2699.

Stein,M.L. (1999) *Interpolation of Spatial Data : Some Theory for Kriging (Springer Series in Statistics)*. Springer, New York.

Swameye,I. *et al.* (2003) Identification of nucleocytoplasmic cycling as a remote sensor in cellular signaling by databased modeling. *Proc. Natl Acad. Sci. USA*, **100**, 1028–1033.

von Mises,R. (1964) *Mathematical Theory of Probability and Statistics*. Academic Press, New York.

Yuan,M. (2006) Flexible temporal expression profile modelling using the Gaussian process. *Comput. Stat. Data Anal.*, **51**, 1754–1764.