A Bayes Linear Approach to Systems Biology

I R Vernon, M Goldstein

September 23, 2010

Abstract

As post-genomic biology becomes more predictive, the inference of rate parameters that feature in both genetic and biochemical networks becomes increasingly important. Here we present a novel methodology for inference of such parameters in the case of stochastic networks, based on concepts from the area of computer models combined with Bayes Linear variance learning methodology.

We apply these techniques to a simple, analytically tractable Birth-Death process model, followed by a more complex stochastic Prokaryotic Auto-regulatory Gene Network.

1 Introduction

Traditionally, chemical reaction networks have been modelled by sets of ODE's. However, for intracellular reaction networks, especially those concerning gene transcription, the discrete number of molecules involved and the inherently stochastic behaviour of the network becomes important. These networks can be accurately model by stochastic processes (namely continuous-time Markov processes), that possess many unknown rate constants representing all the various reactions involved.

The goal of this study is to be able to perform inference on the underlying rate parameters x that feature in such stochastic models of systems biology, using the available data which is often incomplete, measured infrequently and has substantial measurement error. While some models, such as the Birth-Death model considered in section 6, are fast enough so that we could understand them to a high degree of accuracy simply by simulating the process thousands of times for many different values of the inputs x, many more realistic networks take far longer to run and have large numbers of input parameters. This means that such an exhaustive exploration of the model quickly becomes impossible. While an MCMC approach has very many attractive features in this context, it too is limited at a level of complexity which is currently significantly lower than the majority of biological systems of interest. Because of this we are interested in techniques for inference that will 'scale up' to models of significantly higher complexity, and believe the methods presented in this paper should achieve this goal.

We present in this article, novel methodology based around the concept of emulation: a popular technique in the Computer Model area (see for example Craig et al. (1997), Craig et al. (2001), Rougier (2009), Kennedy and O'Hagan (2001), O'Hagan (2006)). Specifically, we demonstrate how to construct emulators for the variance and mean surfaces of a stochastic computer model, using Bayes Linear variance learning. We then show how such emulators can be combined with implausibility measures to cut out areas of the input or parameter space that are highly inconsistent with the observed data. These techniques are applied to a simple Birth-Death process model (that has the benefits of analytical tractability) and then to the more challenging Prokaryotic Auto-regulatory Gene Network model (see wilk).

The report is organized as follows: in section (2) we develop the techniques required to perform a Two Stage Bayes Linear Update on a variance and then mean surface. Section (3) describes the emulation process where we attempt to mimic the behaviour of both the mean and variance of the stochastic model output at various time points as a function of the inputs x. In section (4) we discuss the all important implausibility measure that identifies regions of the parameter space that are unlikely to correspond to some observed process. In section 5 we described the stochastic kinetic biochemical reaction networks that motivate this work, and introduce the two models used. In section 6 we apply our techniques to the Birth-Death model and compare to analytical results. Section 7 contains the main application: that of the Prokaryotic Auto-regulatory Gene Network model, and we conclude in section 8.

2 Two Stage Bayes Linear Update

We consider a stochastic computer model, of which the stochastic kinetic biochemical reaction networks are an example, which has a choice on input (or rate) parameters x and produces at time T an output $Y(t)$. As the model is stochastic, repeated evaluation with the same inputs x would yield a sequence of different values $Y_1(t), Y_2(t), ..., Y_k(t)$. Essentially, in this section we will use a Bayes Linear strategy to emulate (and therefore mimic) the mean and the variance of this sequence of outputs $\{Y_k(t)\}$. We begin by developing a representation of the mean and variance of this type of process $Y_k(t)$, within the Bayes Linear context. As we will mainly be analysing different time points univariately, we will often drop the explicit time dependence of Y_k.

2.1 Representation of the Mean and Variance Surfaces

Setting the random quantity $Y = Y(t)$, we explicitly show the dependence on the input or rate parameters x by writing $Y(x)$. We then say that for fixed x, the sequence $Y(x) = Y_1(x), Y_2(x), ...$ forms an infinite exchangeable sequence of discrete random quantities with $\mathrm{E}(Y_k(x)) = \mu(x)$, $\mathrm{Var}(Y_k(x)) = \sigma^2(x)$, and $\mathrm{Cov}(Y_k(x), Y_j(y)) = \gamma(x, y)$. We can then use the exchangeability representation to write:

$$Y_k(x) \;=\; \mathcal{M}(Y,x) \;+\; \mathcal{R}_k(Y,x), \quad k = 1, 2, ..., \tag{1}$$

where the sequence $\mathcal{R}_1(Y,x), \mathcal{R}_2(Y,x), ...,$ is uncorrelated and we have,

$$\begin{aligned}
\mathrm{E}(\mathcal{R}_k(Y,x)) &= 0 && (2)\\
\mathrm{Var}(\mathcal{R}_k(Y,x)) &= \sigma^2(x) \;-\; \gamma(x) \;=\; V_R(x) && (3)\\
\mathrm{E}(\mathcal{M}(Y,x)) &= \mu(x) && (4)\\
\mathrm{Var}(\mathcal{M}(Y,x)) &= \gamma(x,x) \;=\; \gamma(x) && (5)\\
\mathrm{Cov}(\mathcal{M}(Y,x), \mathcal{M}(Y,x')) &= \gamma(x,x'), && (6)
\end{aligned}$$

with all other covariances equal to zero. As we are dealing with stochastic models, and are planning to update the variance surface first, before updating the mean surface, we now need a representation for the population variance. Thus, let

$$[\mathcal{R}_k(Y,x)]^2 = (Y_k(x) - \mu(x))^2 = V_k(x), \tag{7}$$

and we make the judgement that the sequence $V_1(x), V_2(x), ...$ is also second-order exchangeable. We therefore have the representation

$$[\mathcal{R}_k(Y,x)]^2 \;=\; V_k(x) \;=\; \mathcal{M}(V,x) \;+\; \mathcal{R}_k(V), \tag{8}$$

where similar to before, the sequence $\mathcal{R}_1(V,x), \mathcal{R}_2(V,x), ...,$ is uncorrelated and we have,

$$\begin{aligned} \mathrm{E}(\mathcal{R}_k(V,x)) &= 0 && (9) \\ \mathrm{Var}(\mathcal{R}_k(V,x)) &= V_{R(V)}(x) && (10) \\ \mathrm{E}(\mathcal{M}(V,x)) &= V_R(x) && (11) \\ \mathrm{Var}(\mathcal{M}(V,x)) &= V_M(x) && (12) \\ \mathrm{Cov}(\mathcal{M}(V,x), \mathcal{M}(V,x')) &= C(x,x') && (13) \end{aligned}$$

Note that each $\mathcal{R}_k(V,x)$ is uncorrelated with $\mathcal{M}(V,x)$, and that $\mathcal{M}(V,x)$ represents the population variance.

2.2 Updating the Variance

The problem we now face is that as $\mathcal{M}(Y,x)$ is always unknown, we cannot measure $V_k(x)$ directly. Say for example we create a design and run our model at several different values of $x = x_1, x_2, ...,$ and at each of these values we run the model n times where $n = n_1, n_2, ...,$ then the data we obtain (which is sufficient in terms of Bayes Linear Updating of the variances and the means), are the sample means $(\bar{y}(x_1), \bar{y}(x_2), ..., \bar{y}(x_d))$ and sample variances $S = (s^2_{n_1}(x_1), s^2_{n_2}(x_2), ..., s^2_{n_d}(x_d))$.

We therefore need to build a representation of $s^2_n(x)$ in order to be able to update the expectation and variance of $\mathcal{M}(V,x)$. With the usual definition,

$$s^2_n(x) \quad = \quad \frac{1}{n-1}\Sigma^n_{k=1}(Y_k(x) - \bar{Y}_n(x))^2 \qquad (14)$$

we can then derive:

$$s^2_n(x) \quad = \quad \mathcal{M}(V,x) \; + \; T(x), \qquad (15)$$

where from equation (8),

$$T(x) \quad = \quad \frac{1}{n}\Sigma_k \mathcal{R}_k(V,x) \; - \; \frac{2}{n(n-1)}\Sigma_{k<j}\mathcal{R}_k(Y,x)\mathcal{R}_j(Y,x). \qquad (16)$$

A reasonable assumption to make at this point is that the residuals $\mathcal{R}_j(Y,x)$ satisfy the following forth-order uncorrelated properties:

$$\mathrm{Cov}(\mathcal{M}(V,x), \mathcal{R}_k(Y,x)\mathcal{R}_j(Y,x)) \; = \; \mathrm{Cov}(\mathcal{R}_i(V,x), \mathcal{R}_k(Y,x)\mathcal{R}_j(Y,x)) \; = \; 0, \qquad (17)$$

and that if $k > j, w > u$, and provided $k \neq w$ and $j \neq u$:

$$\mathrm{Cov}(\mathcal{R}_k(Y,x)\mathcal{R}_j(Y,x), \mathcal{R}_w(Y,x)\mathcal{R}_u(Y,x)) \; = \; 0. \qquad (18)$$

The following important properties of $T(x)$ can now be derived from (16,17, 18):

$$\begin{aligned} \mathrm{E}(T) &= 0 && (19) \\ V_T(x) \; = \; \mathrm{Var}(T(x)) &= \frac{1}{n}V_{R(V)}(x) \; + \; \frac{2}{n(n-1)}[V_M(x) + V^2_R(x)] && (20) \\ \mathrm{Cov}(\mathcal{M}(V,x), T(y)) &= 0. && (21) \end{aligned}$$

As we intend to update $M(V,x)$ using the collection of sample variances of our model runs $S = (s^2_{n_1}(x_1), s^2_{n_2}(x_2), ..., s^2_{n_d}(x_d))$, we need expressions for $\mathrm{E}(S)$, $\mathrm{Var}(S)$ and $\mathrm{Cov}(\mathcal{M}(V,x), S)$. Using equations (15,17,18) these are found to be:

$$\mathrm{E}(S) \quad = \quad (V_R(x_1), V_R(x_2), ..., V_R(x_d)) \qquad (22)$$

$$\mathrm{Var}(S) \quad = \quad \begin{pmatrix} V_M(x_1) + V_T(x_1) & \mathrm{Cov}[\mathcal{M}(V,x_1), \mathcal{M}(V,x_2)] & ... & \mathrm{Cov}[\mathcal{M}(V,x_1), \mathcal{M}(V,x_d)] \\ ... & V_M(x_2) + V_T(x_2) & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & V_M(x_d) + V_T(x_d) \end{pmatrix}$$

$$\mathrm{Cov}(\mathcal{M}(V,x), S) = (\mathrm{Cov}(\mathcal{M}(V,x), \mathcal{M}(V,x_1), ..., \mathrm{Cov}(\mathcal{M}(V,x), \mathcal{M}(V,x_d)).$$

The form of $\text{Cov}[\mathcal{M}(V, x_1), \mathcal{M}(V, x_2)]$ and indeed $\text{E}(\mathcal{M}(V, x))$ will be determined in section (3) where we discuss the emulation of the variance surface and the decisions that feed into this process. Leaving these aside, it now remains to specify the form of $V_M(x)$ (the variance of $\mathcal{M}(V, x)$) and $V_{R(V)}(x)$ (the variance of $\mathcal{R}_k(V, x)$) in order to have all that is needed to perform a Bayes Linear Update on the expectation of $\mathcal{M}(V, x)$.

2.3 Prior Choices for $V_M(x)$ and $V_{R(V)}(x)$

$V_{R(V)}(x)$ is a forth order quantity and reflects our judgement as to the shape of the distribution of Y. In order to understand this quantity and to make sensible decisions about it we assume that the population variance acts as a scale parameter in that:

$$\mathcal{R}_i(Y, x) \quad = \quad \sqrt{\mathcal{M}(V, x)} Z_i(x), \tag{23}$$

where $Z_1(x), Z_2(x), ...$ are independent, have expectation zero and variance one, and are independent of the value of $\mathcal{M}(V, x)$. This implies that $\mathcal{R}_i(V, x) = \mathcal{M}(V, x)(Z_i^2(x) - 1)$ which means that $V_{R(V)}(x)$ can be written in terms of the kurtosis of Z_i as:

$$\begin{aligned} V_{R(V)}(x) \quad = \quad \text{Var}(\mathcal{R}_i(V, x)) \quad &= \quad (V_M(x) + V_R^2(x)) \, \text{Var}(Z_i^2(x)) & (24) \\ &= \quad (V_M(x) + V_R^2(x))(\text{Kur}(Z_i(x)) - 1). & (25) \end{aligned}$$

We are free at this stage to make an assessment of the kurtosis of Z_i, for example (if we assume its dependence on x is negligible) we can set $\text{Kur}(Z_i)$ equal to 3 if we believe Y has a normal distribution, $3(\nu - 2)/(\nu - 4)$ for a t-distribution (with $\nu > 4$ degrees of freedom), or 1.8 for a uniform distribution (this in fact is the lower bound on the kurtosis of any regular unimodal symmetric distribution).

In the current case where we have a collection of runs of our systems biology model, we also have another possibility: to assess the kurtosis using information from the models runs themselves. In section (3) we will make initial assessments of kurtosis and use the data as a check to test if our prior value seems reasonable (and to test the assumption of negligible x dependence). Note that the correct approach would be to Bayes Linear Update the kurtosis.

The final quantity we need is $V_M(x) = \text{Var}(\mathcal{M}(V, x))$. This represents the weight we give to our prior value of $V_R(x) = \text{E}(\mathcal{M}(V, x))$, compared to that of the data in the Bayes Linear Update. For a particular value of x, higher values of $V_M(x)$ will mean that we give little weight to our prior and the update will be dominated by $s_n^2(x)$. If there is no s_n^2 measured exactly at x the situation is more complicated (as is discussed in section (3)) but the principle is still the same.

A possible approach is to set $V_M(x) = c(x)V_R(x)^2$ and to examine the effect of the update due to different values of c(x). This is in fact the same as the 'equivalent sample size' approach in which we propose that the prior information at x is worth a notional sample size $m(x)$. The approaches are linked by the equation (with $\kappa(x) = [(n(x) - 1)\text{Var}(Z_i^2) + 2]/n(x))$),

$$c(x) \quad = \quad \frac{\kappa(x)n(x)}{m(x)(n(x) - 1) - \kappa(x)n(x)} \tag{26}$$

In the simple case of updating a variance V_R using a single sample variance s_n^2 where there is no x dependence, the above concepts are clearly defined. However, in the current case we will be updating a variance surface defined over x, using multiple sample variances evaluated at different values of x. This makes the assessment of $V_M(x)$ more difficult, however, possible simple choices are: setting $c(x)$ equal to a constant c such that $0 < c < 1$; setting $m(x)$ equal to a constant m such that $m < \min(n(x))$; or setting $V_M(x)$ equal to a constant related to the emulation process described below which is the strategy we choose.

With the above prior information specified, we are now able to update the expectation and variance of the collection $\mathcal{M}(V,x_g)$ (where $x_g = (x_{g1},...,x_{ge})^T$) having observed $S = (s^2_{n_1}(x_1), s^2_{n_2}(x_2), ..., s^2_{n_d}(x_d))^T$ (with $x_s = (x_1,...,x_d)^T$), using the Bayes Linear Update formula:

$$\begin{aligned}
\mathrm{E}_S[\mathcal{M}(V,x_g)] &= \mathrm{E}[\mathcal{M}(V,x_g)] \;+\; \mathrm{Cov}[\mathcal{M}(V,x_g),S]\ \mathrm{Var}^{-1}[S]\ (S \;-\; \mathrm{E}(S)) && (27)\\
&= V_R(x_g) \;+\; \mathrm{Cov}[\mathcal{M}(V,x_g),\mathcal{M}(V,x_s)]\ \mathrm{Var}^{-1}[S]\ (S \;-\; V_R(x_s)) && (28)\\
\mathrm{Var}_S[\mathcal{M}(V,x_g)] &= \mathrm{Var}[\mathcal{M}(V,x_g)] \;-\; \mathrm{Cov}[\mathcal{M}(V,x_g),S]\ \mathrm{Var}^{-1}[S]\ \mathrm{Cov}[S,\mathcal{M}(V,x_g)] && (29)\\
&= V_M(x_g) \;-\; \mathrm{Cov}[\mathcal{M}(V,x_g),\mathcal{M}(V,x_s)]\ \mathrm{Var}^{-1}[S]\ \mathrm{Cov}[\mathcal{M}(V,x_s),\mathcal{M}(V,x_g)] && (30)
\end{aligned}$$

This is the technique we will use to update the variance surface and it is very similar to the mean update that is described in the next section.

2.4 Updating the Mean

In the Two Stage Bayes Linear Update we first update the variance surface represented by $\mathcal{M}(v,x)$ with respect to the sample variances $S = (s^2_{n_1}(x_1), s^2_{n_2}(x_2), ..., s^2_{n_d}(x_d))^T$, and hence obtain the updated expectation of $\mathcal{M}(V,x)$ which we shall write as:

$$\mathrm{E}_S(\mathcal{M}(v,x)) = V_R^*(x), \qquad (31)$$

with the * representing an updated quantity. We now go on to update the mean surface represented by $\mathcal{M}(Y,x)$ with respect to the collection of sample means $D = (\bar{y}_{n_1}(x_1), \bar{y}_{n_2}(x_2),\ ...,\bar{y}_{n_d}(x_d))$, but now using the new $V_R^*(x)$. As before we have the representation:

$$Y_k(x) \;=\; \mathcal{M}(Y,x) \;+\; \mathcal{R}_k(Y,x), \quad k = 1,2,..., \qquad (32)$$

where we will obtain the terms $\mathrm{E}(\mathcal{M}(Y,x)) = \mu(x)$ and $\mathrm{Cov}(\mathcal{M}(Y,x_1),\mathcal{M}(Y,x_2)) = \gamma(x_1,x_2)$ after discussion of the emulation process below. As we only measure the sample means we need to use equation (32) to represent them in the form:

$$\bar{y}_{n_i}(x_i) \;=\; \mathcal{M}(Y,x) \;+\; \frac{1}{n_i}\Sigma_{k=1}^{n_1}\mathcal{R}_k(Y,x). \qquad (33)$$

This is the equivalent equation to that of (15) and (16) which give the slightly more complicated representation of $s^2_{n_i}(x_i)$. Before we can commence with the update of $\mathcal{M}(Y,x)$ by the collection of the sample means D, we first need to calculate $\mathrm{E}(D)$, $\mathrm{Var}(D)$ and $\mathrm{Cov}(\mathcal{M}(Y,x),D)$. Using equation (33) combined with the uncorrelated and zero expectation properties of $\mathcal{R}(Y,x)$ we find that (writing $\gamma(x_1,x_1) \equiv \gamma(x_1)$):

$$\begin{aligned}
\mathrm{E}(D) &= (\mu(x_1),\mu(x_2),...,\mu(x_d)) && (34)\\
\mathrm{Var}(D) &= \begin{pmatrix} \gamma(x_1)+\frac{1}{n_1}V_R^*(x_1) & \gamma(x_1,x_2) & ... & \gamma(x_1,x_d)\\ . & \gamma(x_2)+\frac{1}{n_2}V_R^*(x_2) & ... & .\\ . & . & ... & .\\ . & . & ... & \gamma(x_d)+\frac{1}{n_d}V_R^*(x_d) \end{pmatrix}\\
\mathrm{Cov}(\mathcal{M}(Y,x),D) &= (\gamma(x,x_1),...,\gamma(x,x_d)).
\end{aligned}$$

Now all that remains is to consider the form of $\gamma(x)$: the variance of $\mathcal{M}(Y,x)$, which is the weight given to our prior value of $\mu(x) = \mathrm{E}(\mathcal{M}(Y,x))$. $\gamma(x)$ is equivalent to the $V_M(x)$ discussed in the previous section. As before there are several possible methods we could use to assess $\gamma(x)$, however the situation here is significantly simpler: if we say our prior information about $\mu(x)$ has again weight $m(x)$ then the equivalent sample size approach gives the obvious result that,

$$\gamma(x) \;=\; \frac{1}{m(x)}\,V_R^*(x). \qquad (35)$$

Of course we are now still free to choose the size of $m(x)$ and indeed the specific form of its x-dependence, but we can now choose this to be consistent with the assessment of $V_M(x)$ (and the choice of $m(x)$) in the previous section.

Again with the above prior information now specified, we are able to update the expectation and variance of the collection $\mathcal{M}(Y, x_g)$ (where $x_g = (x_{g1}, ..., x_{ge})^T$) having observed $D = (\bar{y}_{n_1}(x_1), \bar{y}_{n_2}(x_2), ..., \bar{y}_{n_d}(x_d))$ (with $x_s = (x_1, ..., x_d)^T$), using the Bayes Linear Update formula:

$$\begin{aligned}
\mathrm{E}_D[\mathcal{M}(Y, x_g)] &= \mathrm{E}[\mathcal{M}(Y, x_g)] \;+\; \mathrm{Cov}[\mathcal{M}(Y, x_g), D]\, \mathrm{Var}^{-1}[D]\, (D \;-\; \mathrm{E}(D)) && (36)\\
&= \mu(x_g) \;+\; \gamma(x_g, x_s)\, \mathrm{Var}^{-1}[D]\, (D \;-\; \mu(x_s)) && (37)\\
\mathrm{Var}_D[\mathcal{M}(Y, x_g)] &= \mathrm{Var}[\mathcal{M}(Y, x_g)] \;-\; \mathrm{Cov}[\mathcal{M}(Y, x_g), D]\, \mathrm{Var}^{-1}[D]\, \mathrm{Cov}[D, \mathcal{M}(V, x_g)] && (38)\\
&= \mu(x_g) \;+\; \gamma(x_g, x_s)\, \mathrm{Var}^{-1}[D]\, \gamma(x_s, x_g). && (39)
\end{aligned}$$

We now have the necessary techniques to be able to perform updates upon a variance surface, followed by a mean surface. These are required for our treatment of the systems biology models as we will emulate both the variance and mean surfaces of the model, for which such updating is essential. The emulation process is described in the next section.

3 Emulation of Mean and Variance

Many methods of performing inference on the rate parameters of a stochastic network model, such as MCMC, can be both computer intensive and generally do not scale up well to networks of realistic size. For this reason we have developed a more tractable Bayes Linear approach to the problem and in this section we describe one of its main features: the emulation of the mean and variance surfaces. As was mentioned in the previous section, if we do a design of runs of our model at several different values of $x = x_1, x_2, ...$, and at each of these values we run the model n times where $n = n_1, n_2, ...,$, we will only have information about the sample mean and variance at each of the points $x = x_1, x_2, ...,$. Assuming that we have not totally covered our input space with a large numbers of runs, we therefore need to use emulation to represent our beliefs about the output of the process at a new point x. As describing the probability density of the output $Y = Y(T)$ given x is overly complicated, we restrict ourselves to looking at the expectation and variance of the output (and our corresponding uncertainties about these quantities) given x. These were denoted previously as $\mathcal{M}(Y, x)$ and $\mathcal{M}(V, x)$ and are defined by equations (1) and (8).

We describe our beliefs about the behavior of $\mathcal{M}(Y, x)$ and $\mathcal{M}(V, x)$ (defined by equations (1) and (8)), via two emulators:

$$\begin{aligned}
\mathcal{M}(V, x) &= \beta_V^T g_V(x) \;+\; \epsilon_V(x), && (40)\\
\mathcal{M}(Y, x) &= \beta_Y^T g_Y(x) \;+\; \epsilon_Y(x), && (41)
\end{aligned}$$

where the $g_V(x)$ and $g_Y(x)$ are known simple functions of x that we feel have a linear effect on $\mathcal{M}(V, x)$ and $\mathcal{M}(Y, x)$ respectively, and the $\epsilon(x)$ terms are residual terms from the simple linear fit, with zero expectation and variance structures that will be discussed below. The emulators are not an attempt at a physical model for the relation between the inputs and outputs of the process, instead they are a convenient tool that allow us to represent our subjective beliefs about the mean and variances of the process and their dependence upon the input parameters given by x. Note that as we will be dealing with the relatively low dimensional Birth-Death process (and then a reduced dimension Prokaryotic model) we write equation (40) in terms of both input parameters λ and μ as $x = (\lambda, \mu)^T$. When dealing with more complicated models that have many input parameters we would assess which subset of these are the most significant for each output in question (called the active inputs) and write

(40) in terms of only those. The point being that as the model gets more complicated, our emulators do not increase significantly in complexity, although they will gain an extra discrepancy term related to the ignored extra input parameters.

We choose the following form for both the functions $g_V(x)$ and $g_Y(x)$:

$$g_V(x) \;=\; g_Y(x) \;=\; (1, \lambda, \mu, \lambda\mu, \lambda^2, \mu^2), \tag{42}$$

a decision made to ensure an accurate linear fit in the emulators. The $\epsilon(x)$ terms express our belief that $\mathcal{M}(V,x)$ and $\mathcal{M}(Y,x)$ cannot be described purely by quadratic surfaces. Clearly the $\epsilon(x)$ terms must be strongly correlated for neighbouring values of x and so we impose a prior covariance structure over $\epsilon_V(x)$ of the form:

$$\mathrm{Cov}(\epsilon_V(x), \epsilon_V(x')) \;=\; \sigma^2_{\epsilon_V} \;\exp\left[-\theta_V (x-x')^T (x-x')\right], \tag{43}$$

with a similar structure for $\epsilon_Y(x)$. The terms $\sigma^2_{\epsilon_V}$ and θ_V (and $\sigma^2_{\epsilon_Y}$ and θ_Y) are constants to be chosen (motivated by the emulation construction process). Once these choices have been made, the first emulator then gives:

$$\begin{aligned} V_R(x) &= \mathrm{E}(\mathcal{M}(V,x)) && (44)\\ V_M(x) &= \mathrm{Var}(\mathcal{M}(V,x)) && (45)\\ C(x,x') &= \mathrm{Cov}(\mathcal{M}(V,x), \mathcal{M}(V,x')) && (46) \end{aligned}$$

while the second emulator gives:

$$\begin{aligned} \mu(x) &= \mathrm{E}(\mathcal{M}(Y,x)) && (47)\\ \gamma(x) &= \mathrm{Var}(\mathcal{M}(Y,x)) && (48)\\ \gamma(x,x') &= \mathrm{Cov}(\mathcal{M}(Y,x), \mathcal{M}(Y,x')) && (49) \end{aligned}$$

and we would now have everything we need to update the variance surface followed by the mean surface.

There are choices at this stage over the specific manner in which we construct the emulator. For example, should we use the equivalent sample size approach described in the previous section (to find e.g. $V_M(x)$ first, then obtain $\sigma^2_{\epsilon_V}$), or should we obtain such information about $\sigma^2_{\epsilon_V}$ and $\sigma^2_{\epsilon_Y}$ from linear modelled fitting of the run data? In line with previous emulation experience (Vernon et al. (2010)), we choose the latter approach and choose the remaining unspecified quantities to be consistent with simple linear model fits of S and D. See Vernon et al. (2010) for further discussion of emulation strategies.

We now have all we need and can perform the Bayes Linear Update of $\mathcal{M}(V,x)$ by the sample variance data S using the techniques given in section (2.2). Hence we obtain the updated expectation and variance of the variance surface, with respect to S:

$$\begin{aligned} V_R^*(x) &= \mathrm{E}_S(\mathcal{M}(V,x)), && (50)\\ V_M^*(x) &= \mathrm{Var}_S(\mathcal{M}(V,x)), && (51) \end{aligned}$$

where the * implies updated value. In a similar manner we now do the second stage of the two stage Bayes Linear Analysis, and use $V_R^*(x)$ to update $\mathcal{M}(Y,x)$ and obtain the updated expectation and variance for the mean surface:

$$\begin{aligned} \mu^*(x) &= \mathrm{E}_D(\mathcal{M}(Y,x)), && (52)\\ \gamma^*(x) &= \mathrm{Var}_D(\mathcal{M}(Y,x)). && (53) \end{aligned}$$

These can now be used in implausibility measures discussed in the next section.

4 Implausibility Measures

A useful concept for identifying areas of the input parameter space that are highly unlikely to give rise to a certain set of observations, is that of an Implausibility Measure (Craig et al. (1997) and Vernon et al. (2010)). We can apply this concept to both the variance and mean of the stochastic computer model as follows. Let $f(x)$ represent either $\mathcal{M}(V, x)$ or $\mathcal{M}(Y, x)$, and say we have observed measurements of the mean or variance process given by z and we are interested in which values of x could be consistent with z.

We say that z can be written as:

$$z \quad = \quad y_T \ + \ e \tag{54}$$

where y_T is the actual real world process mean or variance, e is an observational error term which is uncorrelated with y_T prior to observing z. We then say that y_T is linked to our model $f(x)$ by:

$$y_T \quad = \quad f(x^+) \ + \ d, \tag{55}$$

where x^+ represents the choice of input parameters we would make if we had perfect knowledge of the real biological process, and d represents the difference between our model of the process and the real biological process (known as the model discrepancy). There are some subtleties regarding the definition of (55) which we will ignore here (see Goldstein and Rougier (2009) for details).

As we have now linked observations in the real world to our model $f(x)$ and therefore to x we can now define the Implausibility Measure:

$$\mathcal{I}(x)^2 \quad = \quad \frac{\mathrm{E}_D(f(x) - y_T)^2}{\mathrm{Var}_D(f(x) - y_T)} \tag{56}$$

$$\quad = \quad \frac{(\mathrm{E}_D(f(x)) - z)^2}{\mathrm{Var}_D(f(x)) + \mathrm{Var}(e) + \mathrm{Var}(d)}. \tag{57}$$

Remember that $f(x)$ here represents either the variance $\mathcal{M}(V, x)$ or the mean $\mathcal{M}(Y, x)$, and D represents either S or D respectively. Note that high values of $\mathcal{I}(x_1)$ imply that it is very unlikely that we would find that $f(x)$ is close to y_T in the vicinity of $x = x_1$, whether low values of $\mathcal{I}(x)$ are due to two reasons: either we expect that running the simulator at x will give a close match between $f(x)$ and y_T, or because we are so uncertain about $f(x) - y_T$ that there could be a good match at x but we can't tell.

The implausibility measure is an extremely useful tool as applied to computer models as it can be used to cut out all implausible areas of the input parameter space, and it has been used is several areas of application (Craig et al. (1996); Cumming and Goldstein (2009); Bower et al. (2009); Vernon et al. (2010)). It can also be used to design the next set of simulator runs in a sequential design. We will use this measure extensively in sections 6 and 7 where we apply it to the Birth-Death process model and to the Prokaryotic network model respectively.

4.1 Fully Bayesian Approach: MCMC Discussion

The nature of the problem of performing inference on rate parameters of systems biology models, suggests that a fully Bayesian Markov-chain Monte Carlo approach would be worthwhile. Indeed the MCMC methods currently employed for problems of this type have many attractive features and can sensibly deal with issues such as partially observed data (where not all of the chemical species are measured) and measurement error. These techniques require the use of Diffusion Models to represent the system. Although the behavior of such models can be quite different from the true discrete model, diffusion models are accurate enough for the purpose of performing inference.

Other issues such as observations only being made at infrequent discrete, regular time intervals can also be overcome through the use of a technique known as data augmentation which for each MCMC step simulates additional data points between the observed data. This is done to ensure the accuracy of the Euler approximation which requires a small distance between data points, but presents its own challenges related to the mixing of the MCMC algorithm.

The main problem with approaches of this form is that they are computationally extremely intensive even for moderately sized systems, and this complexity grows rapidly with system size. As the sizes of networks that are of current interest to Biologists are an order of magnitude larger than those that can be handled using such MCMC techniques, more tractable approaches such as the Bayes Linear emulation approach presented here are urgently required for use in this area.

5 Stochastic Kinetic Biochemical Network Models

5.1 Chemical Reaction Networks

Consider the simple chemical reaction

$$Y_1 + Y_2 \longrightarrow Y_3 \tag{58}$$

which represents a single molecule of species Y_1 combining with one molecule of species Y_2 to produce a single molecule of Y_3. The rate or *hazard* of this reaction will be constant in time (as shown by Gillespie (1977)), and the law of mass action asserts that the hazard will be proportional to Y_1Y_2, that is the product of the numbers of molecules of each chemical species. The proportionality constant is known as the reaction rate constant, and is denoted x. Learning about such rate constants x (which are viewed as the inputs to a stochastic computer model) is the main aim of this article.

The general form of a reaction network, which we will study two examples of, involves k species of chemical $Y_1, Y_2, ..., Y_k$ and r reactions $R_1, R_2, ..., R_r$, occurring in thermal equilibrium inside some fixed volume (see Golightly and Wilkinson (2005)). We can write this set of reactions as:

$$\begin{aligned}
R1 : u_{11}Y_1 + u_{12}Y_2 + ... + u_{1k}Y_k \quad &\longrightarrow \quad v_{11}Y_1 + v_{12}Y_2 + ... + v_{1k}Y_k && (59)\\
R2 : u_{21}Y_1 + u_{22}Y_2 + ... + u_{2k}Y_k \quad &\longrightarrow \quad v_{21}Y_1 + v_{22}Y_2 + ... + v_{2k}Y_k && (60)\\
\vdots \quad &\;\;\vdots \quad\;\; \vdots && (61)\\
Rr : u_{r1}Y_1 + u_{r2}Y_2 + ... + u_{rk}Y_k \quad &\longrightarrow \quad v_{r1}Y_1 + v_{r2}Y_2 + ... + v_{rk}Y_k && (62)
\end{aligned}$$

The arrow $\longrightarrow$ represents the conversion of Reactants to Products and may summarise more complicated intermediate steps in the reaction (which may be of no interest at the current level of modeling). Here u_{ij} is the *stoichiometry* of the jth reactant of the ith reaction, and similarly v_{ij} is the stoichiometry of the jth product of the ith reaction. Each of the possible reactions R_i has an associated rate constant x_i and a reaction hazard $h_i(Y, x_i)$ with $Y = (Y_1, Y_2, ..., Y_k)^T$ being the current state of the system. The form of the hazards are determined by mass action kinetics, and depend on the order of the reaction R_i as is described in detail in appendix A. For example, the hazard corresponding to equation (58) would be $h(Y, x) = xY_1Y_2$ (see Wilkinson (2006)).

A popular method of describing such a network is by defining a Petri Net N, which is a list of five objects that define the state of the system, and all the possible reactions. Here $N = \{C, T, U, V, Y\}$, where C is a vector of chemical names, $T = (R_1, ..., R_r)^T$ is a vector of reactions, U and V are $r \times k$ matrices that give the number of molecules used and produced in each reaction, having elements u_{ij} and v_{ij} respectively, and Y is a vector of the number of molecules that are currently in the system (the state vector). Note that the actual effect of each reaction is encapsulated by the net effect reaction matrix $A = V - U$. Knowledge of N and the form of the hazards $h_i(Y, x_i)$ is all that is required to simulate realisations from such reaction networks as we describe in the next section.

5.2 Continuous Time Discrete State Markov Process Model

The behaviour of such reaction networks described in section 5.1, is often analysed in terms of the "Chemical Master Equation". This is a differential equation for the probability that the system is in a state Y at time t, which we denote $P(Y;t)$, and is given by:

$$\frac{\partial}{\partial t}P(Y;t) = \sum_{i=1}^{r}\{h_i(Y - A_i, x_i)P(Y - A_i;t) - h_i(Y, x_i)P(Y;t)\} \tag{63}$$

This can be derived from consideration of the number of ways the system can arrive in state Y at time t (see Wilkinson (2006) and Gillespie (1977) for details). The master equation is an exact description of the system, however, it can only be solved analytically in a small number of cases (one of which is the Birth-Death process described in section 5.3). To analyse the larger, more realistic networks of interest to biologists, discrete event simulation algorithms such as the Gillespie algorithm are used. Details of the full Gillespie algorithm are given in appendix B, but its basic structure is as follows.

As the hazard for a reaction of type i is $h_i(Y, x_i)$, we know that the hazard for a reaction of *some* type is given by:

$$h_0(Y, x) = \sum_{i=1}^{r} h_i(Y, x_i) \tag{64}$$

and therefore the time to the next reaction will be $\text{Exp}(h_0(Y, x))$. Also, this reaction will be of random type i, with probability proportional to the ith hazard and hence given by $h_i(Y, x_i)/h_0(Y, x)$. Therefore, to simulate the time and type of the next reaction is simple, requiring only standard techniques. See appendix B, Golightly and Wilkinson (2005), Gillespie (1977) and references therein for more detailed summaries.

In essence the Gillespie algorithm, with a suitably defined network, is the stochastic computer model referred to throughout this article. Its outputs (realisations of the state of the system at various times $Y(t)$) and their dependence on the inputs or rate constants x, are the subject of the Bayes Linear two stage emulation strategy described in sections 2 and 3. Specifically, we emulate the expectation and variance of $Y(t)$ in order to compare with observed data using the implausibility measures described in section 4.

5.3 Example: Birth-Death Process

We have chosen to initially examine the simple birth-death model in order to help develop our techniques. This model is basic in that it has only one species of 'chemical' the number of which at time T we write as $Y(T)$. Hence Y is discrete, taking only integer values but is dependent on a continuous time parameter T. This system has the simple rules that Y can change due to two types of reaction:

$$Y \rightarrow Y + 1, \quad \text{with rate } \lambda Y, \tag{65}$$

$$Y \rightarrow Y - 1, \quad \text{with rate } \mu Y. \tag{66}$$

We write the rate parameters as $x = (\lambda, \mu)^T$ and represent the above reactions by the vector of reaction hazards $h(y, x) = (\lambda, \mu y)$ and the reaction matrix $A = (1, -1)^T$ (or

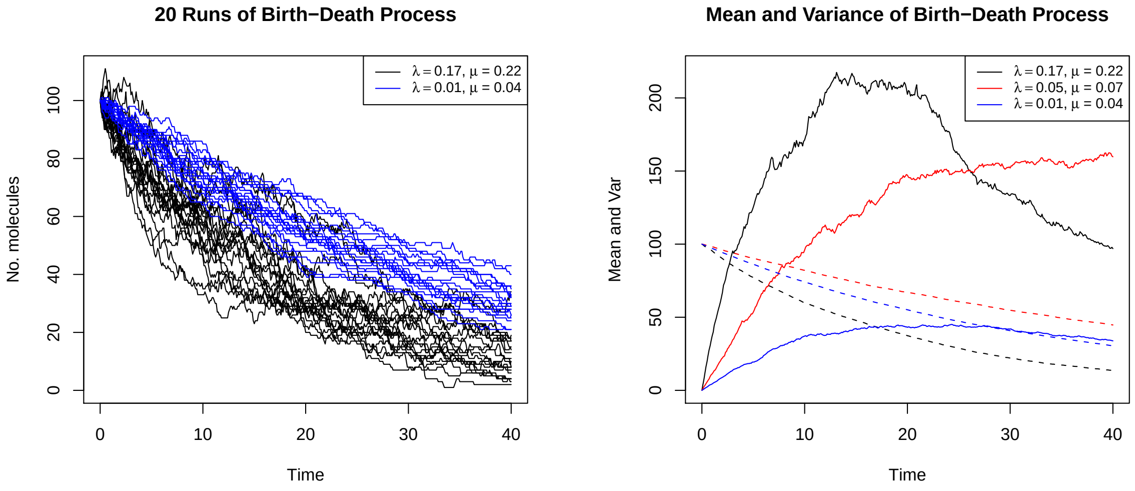

Figure 1: The Birth-Death Process.

equivalently the stoichiometry matrix $S = A^T = (1, -1)$). We take as initial conditions $Y(0) = 100$. This system can easily be simulated using the Gillespie algorithm described in appendix B, and figure 1 (left panel) shows 20 realisations of the process for two choices of the inputs λ and μ, as the blue and black lines. The right panel of figure 1 shows the sample means and variances of 500 realisations of the process, for three different choices of the inputs. It can be seen that the variances contain very different information about the inputs compared to the means (which only depend on the difference $\lambda - \mu$). This will be seen in the implausibility plots of section 6.2. Note that using the Master Equation (equation 5.2) we can derive and solve differential equations for both the mean and variance of $Y(T)$, and we use these for emulator diagnostics in section 6.1.1.

5.4 Example: Prokaryotic Auto-regulatory Gene Network

The prokaryotic auto-regulatory gene network has been often studied as it represents a network which exhibits gene self-repression: a common feature in many biological networks. Here transcription of a gene g results in the production of an mRNA molecule r. The mRNA molecule causes production of a protein P (a process referred to as *Translation*). Two P molecules form the dimer P_2, which represses the gene by binding to a regulatory site upstream of the gene. This can be written as:

$$\begin{aligned} g + P_2 &\Longleftrightarrow g.P_2 && \text{Repression} && (67)\\ g &\longrightarrow g + r && \text{Transcription} && (68)\\ r &\longrightarrow r + P && \text{Translation} && (69)\\ 2P &\Longleftrightarrow P_2 && \text{Dimerisation} && (70)\\ r &\longrightarrow \emptyset && \text{mRNA degradation} && (71)\\ P &\longrightarrow \emptyset && \text{Protein degradation} && (72) \end{aligned}$$

This is a necessarily simplified version of a true prokaryotic system, but is very useful to study as it exhibits many of the features found in real systems. See Wilkinson (2006) for descriptions of other related processes: e.g. Translation, Degradation and Transport.

The Prokaryotic Auto-regulation network has Petri Net:

$$P = \begin{pmatrix} g.P_2 \\ g \\ r \\ P \\ P_2 \end{pmatrix}, \quad T = \begin{pmatrix} \text{Repression} \\ \text{Reverse Repression} \\ \text{Transcription} \\ \text{Translation} \\ \text{Dimerisation} \\ \text{Dissociation} \\ \text{mRNA degradation} \\ \text{Protein degradation} \end{pmatrix} \tag{73}$$

$$Pre = \begin{pmatrix} 0&1&0&0&1 \\ 1&0&0&0&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&2&0 \\ 0&0&0&0&1 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \end{pmatrix}, \quad Post = \begin{pmatrix} 1&0&0&0&0 \\ 0&1&0&0&1 \\ 0&1&1&0&0 \\ 0&0&1&1&0 \\ 0&0&0&0&1 \\ 0&0&0&2&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \end{pmatrix}, \quad M = \begin{pmatrix} 10 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{74}$$

If we combine the above Petri Net with stochastic kinetic rules stating the rate of each reaction, we then have everything needed to simulate the network, which can be performed using the Gillespie algorithm given in appendix B. The Reaction Matrix A is defined as

$$A = Post - Pre = \begin{pmatrix} 1&-1&0&0&-1 \\ -1&1&0&0&1 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&-2&1 \\ 0&0&0&2&-1 \\ 0&0&-1&0&0 \\ 0&0&0&-1&0 \end{pmatrix} \tag{75}$$

A has rows that represent the effect of individual reactions on the state of the network. Similarly, the Stoichiometry Matrix S is defined as $S = A^T$.

Figure 2 shows a single realisation of the Prokaryotic Network, simulated over 200 seconds with inputs or rate parameters set to $x = (1, 10, 0.01, 10, 1, 1, 0.1, 0.1)$. Note that the P_2 molecule still exhibits highly stochastic behaviour even though the numbers of P_2 molecules are often high. It should also be understood that $g.P_2$ and g are related by a conservation law as will be discussed further below.

6 Application to the Birth-Death Model

6.1 Variance and Mean Emulation

Here we apply the techniques of sections 2 and 3 to the Birth-Death model, in order to construct emulators of the variance and mean surface. As this model is analytically solvable, we then compare the emulators with the exact results.

We examine the area of input space defined by: $0 < \lambda < 0.08$ and $0.04 < \mu < 0.13$ and begin by using the Gillespie algorithm described above to perform a $d = 15$ input point maximin latin hypercube design, with $n = 40$ repetitions at each point. We use these runs to construct mean and variance emulators for the model output at 4 different time points $T = (2, 8, 18, 30)$. Figure 3 shows the expectation of the variance emulator (that is $V_R^*(x) = \mathrm{E}_S(\mathcal{M}(V, x))$) for time point $T = 8$ (top-left panel) and

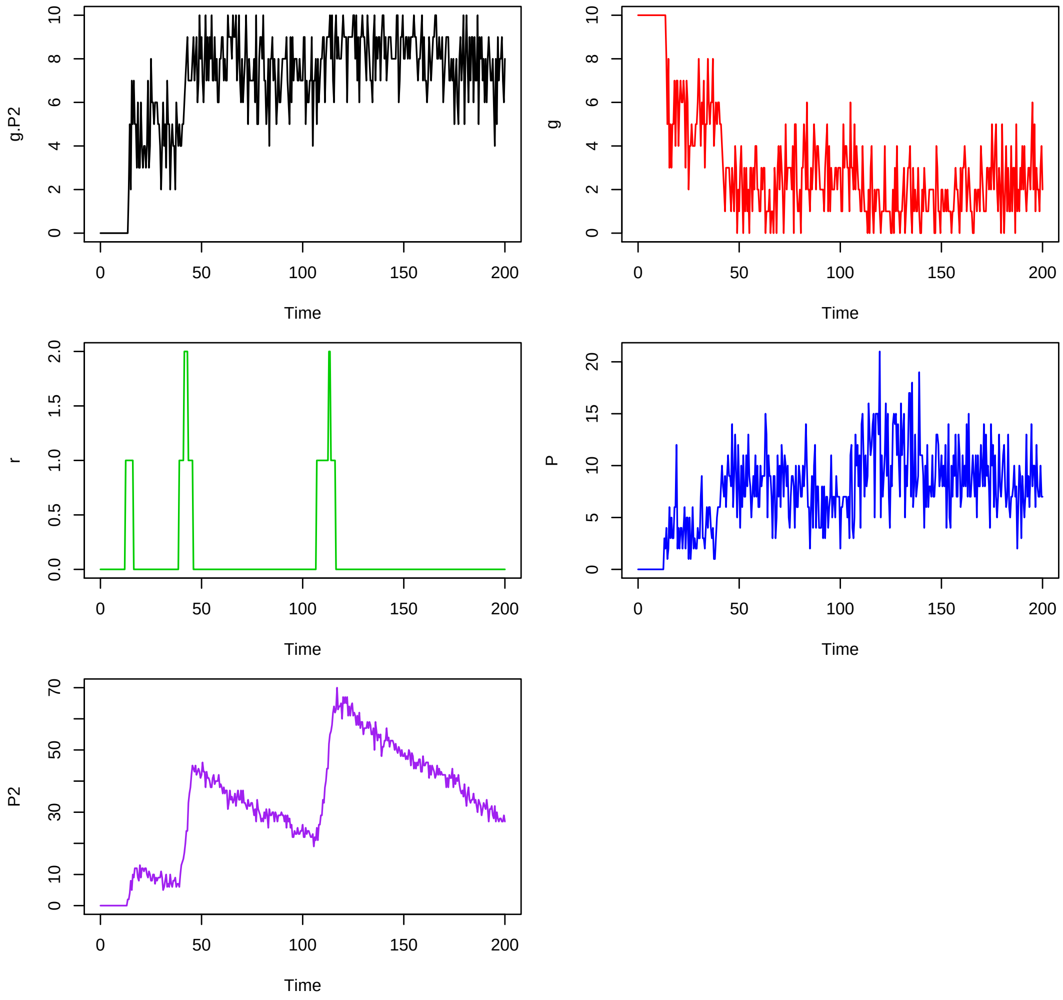

Figure 2: A single realisation of the Prokaryotic Network, simulated over 200 seconds with inputs (or rate) parameters set to $x = (1, 10, 0.01, 10, 1, 1, 0.1, 0.1)$. The panels show the numbers of $g.P_2$, g, r, P and P_2 molecules respectively. Note that the P_2 molecule still exhibits highly stochastic behaviour even though the number of P_2 molecules are often high. This model is emulated and inference performed with respect to its inputs in section 7.

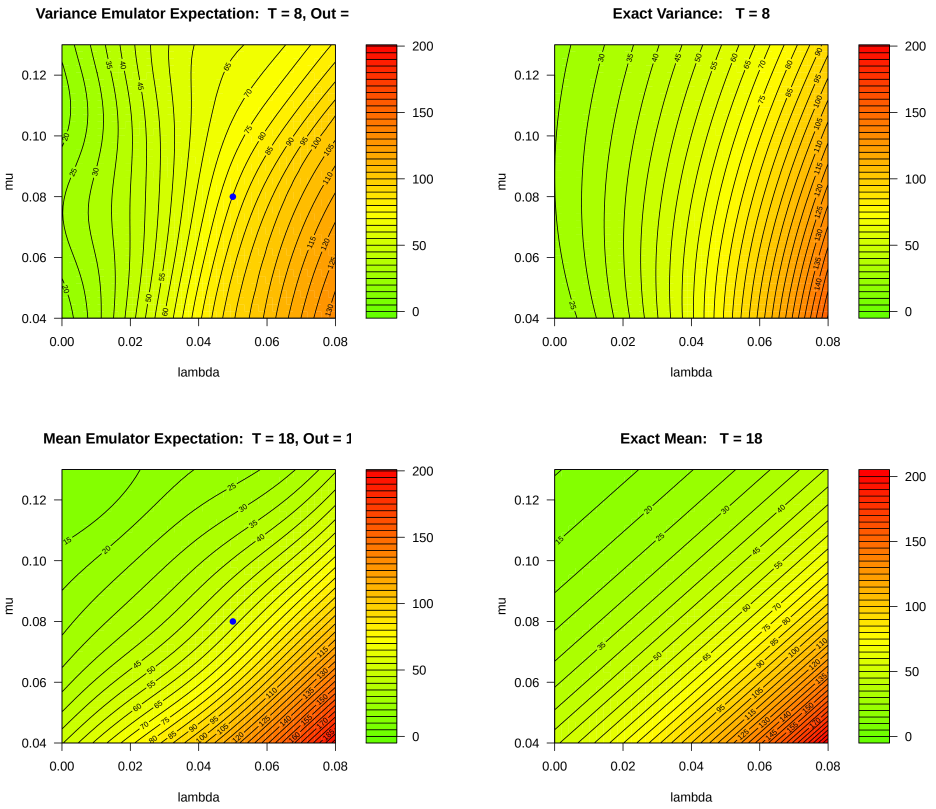

Figure 3: Example emulators compared with exact solutions. The expectation of the variance emulator $\mathrm{E}_S(\mathcal{M}(V,x))$ is shown for time point $T = 8$ (top-left panel) and the analytically exact solution for the variance surface found from equation 5.2 (top-right panel). Also shown is the expectation of the mean emulator $\mathrm{E}_D(\mathcal{M}(Y,x))$ for time point $T = 18$ (bottom left panel) and the corresponding exact solution found from the Master Equation (bottom right panel)

the analytically exact solution for the variance surface found from equation 5.2 (top-right panel). Also shown is the expectation of the mean emulator (that is $\mu^*(x) = \mathrm{E}_D(\mathcal{M}(Y,x))$) for time point $T = 18$ (bottom left panel) and the corresponding exact solution found from the Master Equation (bottom right panel).

In both cases, and for the remaining emulators at the other time points, it is seen that their expectations do mimic the behaviour of the exact solution to a reasonable degree of accuracy. We then perform more rigorous diagnostics described in the next section.

6.1.1 Diagnostics

As we have the exact solution for the mean and variance surface for the Birth-Death model, derived from equation 5.2, we can perform exhaustive diagnostics over the whole input space. Writing the exact solutions as $V_{exact}(x)$ and $\mu_{exact}(x)$ for the variance and mean respectively, we examine the following diagnostic quantities $D_v(x)$

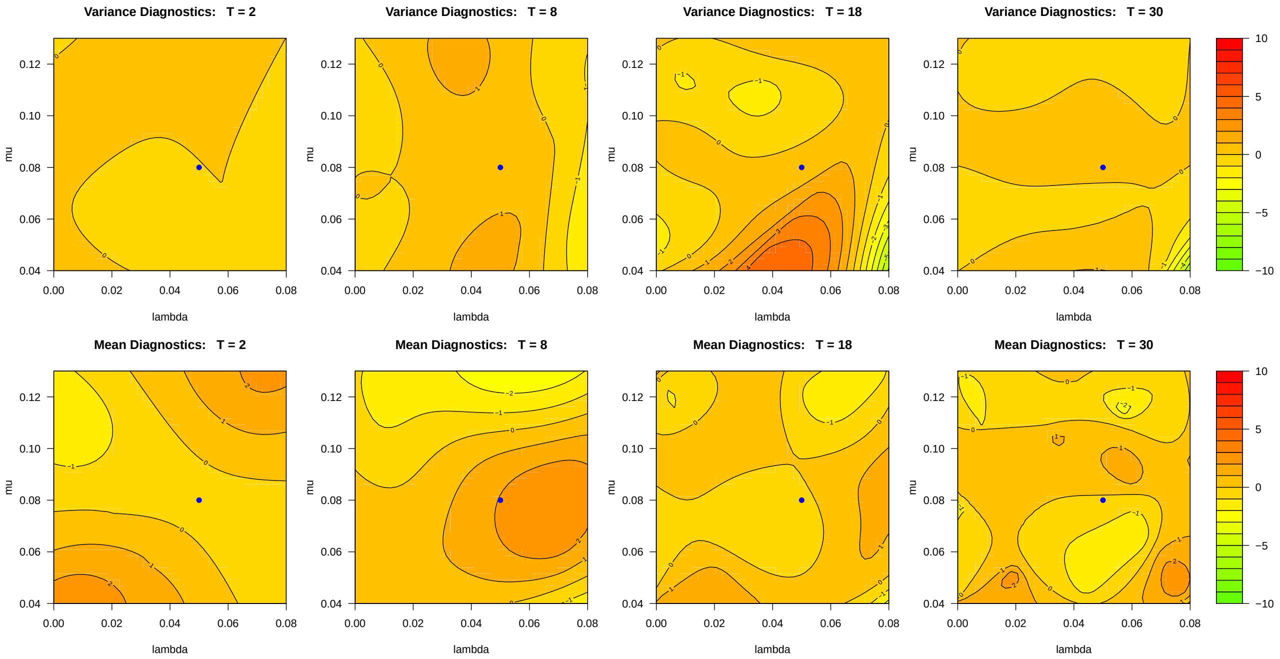

Figure 4: Contour plots of the diagnostic measures for the variance and mean emulators $D_v(x)$ and $D_m(x)$ respectively (top row and bottom row), for each of the four time points $T = (2, 8, 18, 30)$ considered. Note that, in general, the diagnostics behave well and are mainly smaller than expected.

and $D_m(x)$:

$$D_v(x) \;=\; \frac{\mathrm{E}_S(\mathcal{M}(V,x)) - V_{exact}(x)}{\sqrt{\mathrm{Var}_S(\mathcal{M}(V,x))}} \;=\; \frac{V_R^*(x) - V_{exact}(x)}{\sqrt{V_M^*(x)}} \tag{76}$$

and

$$D_m(x) \;=\; \frac{\mathrm{E}_D(\mathcal{M}(Y,x)) - \mu_{exact}(x)}{\sqrt{\mathrm{Var}_D(\mathcal{M}(Y,x))}} \;=\; \frac{\mu^*(x) - \mu_{exact}(x)}{\sqrt{\gamma^*(x)}} \tag{77}$$

which represent the distance the emulator mean is from the exact solution, standardised by the emulator variance. Contour plots of these diagnostic measures $D_v(x)$ and $D_m(x)$ are shown in figure 4 for all four time points considered. As can be seen, in general, the diagnostics proved to be comfortably acceptable, with the vast majority of input space satisfying $|D_v(x)| < 2$ and $|D_m(x)| < 2$. For time point $T = 18$ there is one area of input space where $D_v(x)$ does get larger than expected (the red area in figure 4). Further investigation showed that this is due to a combination of the exact variance surface exponentially increasing at the corner of the input space and distorting the emulator polynomial, and there being no model runs close to the problem region to correct for this. As shown later, this effect is not large enough to pose a problem.

6.2 Implausibility

We now use the Implausibility measures described in section 4 to learn about inputs to the model that may give rise to outputs that are in agreement with observed data. We first created some simulated data as follows. We chose values for the inputs $\lambda = 0.05$ and $\mu = 0.08$ and simulated 500 runs of the model using the Gillespie algorithm as

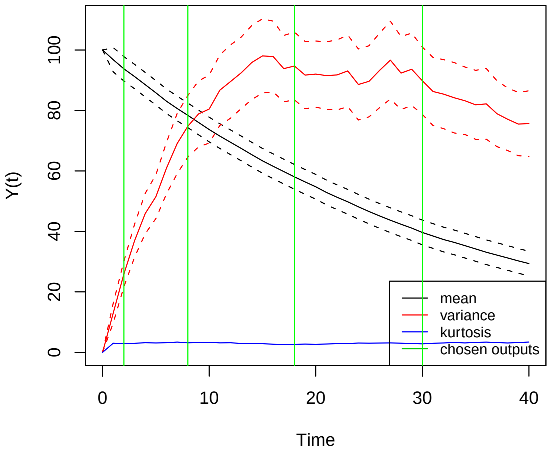

Figure 5: The (simulated) observed data of the Birth-Death process against time: the mean and variance are the black and red lines respectively, with their observational errors given by the dashed lines. Also shown are the 4 time points $T = (2, 8, 18, 30)$ considered as the vertical green lines, and the sample kurtosis in blue.

before. We took the sample mean and sample variance to be the observed mean and variance data (alternately represented by z), and the sample error on these quantities was used as the observational error e discussed in section 4. We also included a small model discrepancy d on each of the mean and variance quantities.

The observed data is shown in figure 5 with the mean and variance as the black and red lines respectively, with their observational errors given by the dashed lines. Also shown are the 4 time points considered as the vertical green lines, and the sample kurtosis is in blue.

We now use the implausibility measures given by equation 57, combined with the emulators for the mean and variance surface, to examine the input space of the Birth-Death model, and specifically to rule out regions of the input space that are deemed inconsistent with the observed data described above.

Figure 6 shows the implausibility measures $\mathcal{I}(x)$ obtained from equation 57 using either the variance (left column) or mean (right column) emulators, for each of the 4 time points considered (the four rows). The red areas represent high implausibility and these inputs would be considered highly unlikely to produce outputs consistent with the observed data of figure 5, and would be discarded from further analysis. The green areas may produce good fits to the observed data, or may warrant further investigation. The “true” input used to generate the observed data is shown as the blue dot.

As can be seen from the right column of figure 6, the mean process of the Birth-Death model can resolve uncertainty about the inputs x along the $\lambda - \mu$ direction only. In order to learn about the $\lambda + \mu$ direction we need information from the variance (left column), specifically at early times. This is of course, in agreement with known analytical properties of the Birth-Death process (which was chosen for analysis for precisely this reason).

We can combine such implausibility measures in many ways; possibly the simplest is to maximise $\mathcal{I}(x)$ over the different cases. In figure 7 we show the implausibility obtained from maximising $\mathcal{I}(x)$ over the 4 variance cases (left panel) corresponding

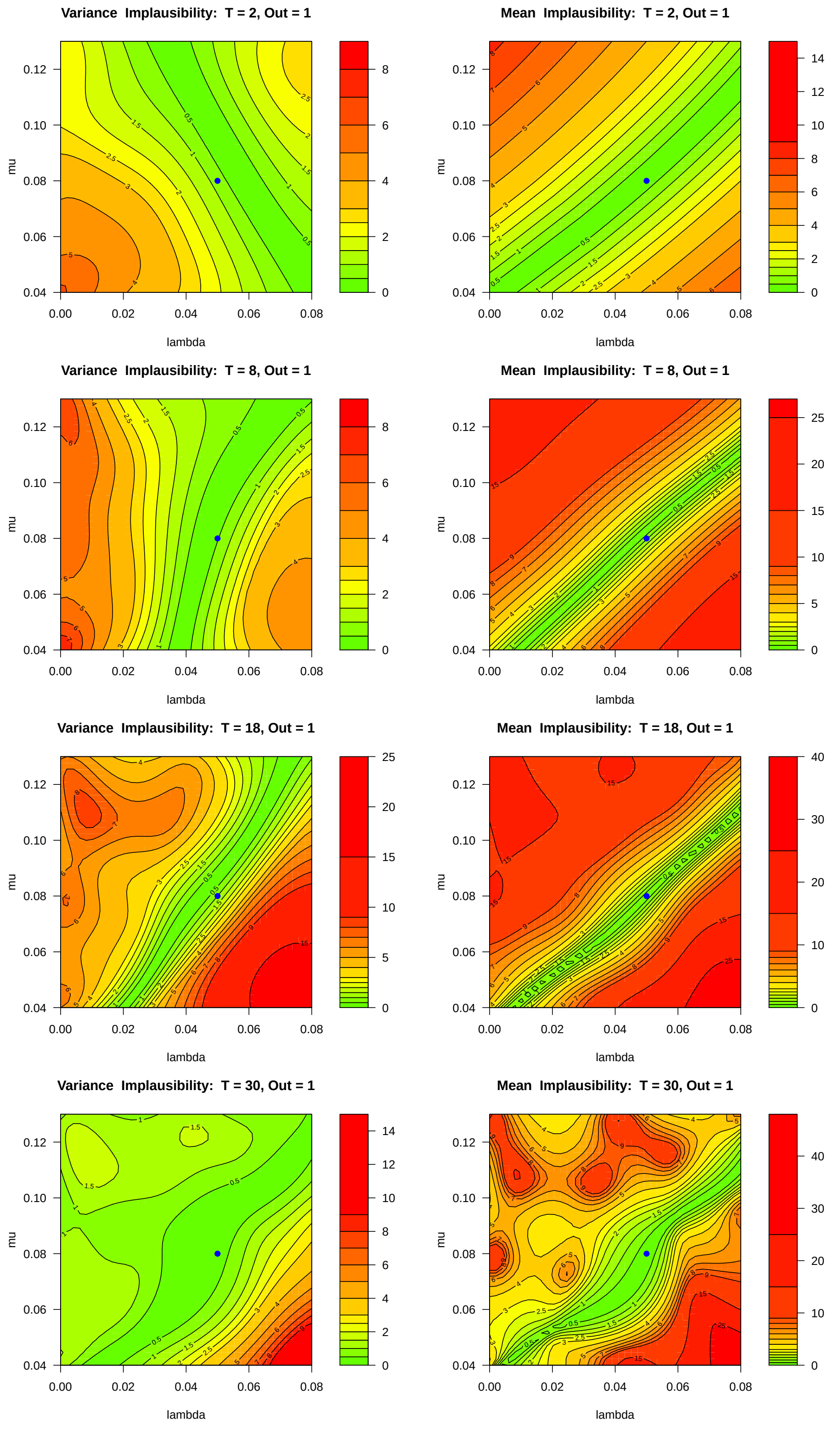

Figure 6: The implausibility measures $\mathcal{I}(x)$ obtained from equation 57 using either the variance (left column) or mean (right column) emulators, for each of the 4 time points considered (the four rows). The red areas represent high implausibility and these inputs would be considered highly unlikely to produce outputs consistent with the observed data of figure 5, and would be discarded from further analysis. The "true" input used to generate the observed data is shown as the blue dot.

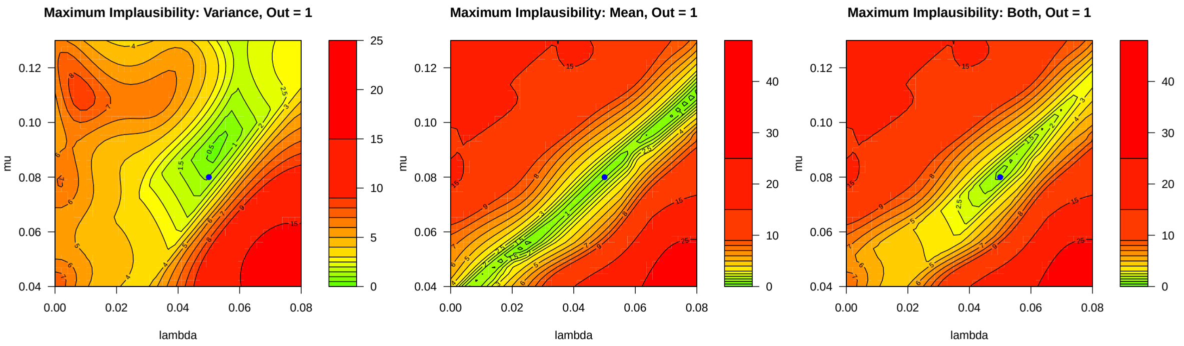

Figure 7: The implausibility obtained from maximising $\mathcal{I}(x)$ over the 4 variance cases (left panel), over the 4 mean cases (middle panel) and over all cases (right panel). The "true" input value is given by the blue dot. Red/orange areas of the input space would be ruled out as inconsistent with the data.

to the left column of figure 6, over the 4 mean cases (middle panel) corresponding to the right column of figure 6, and over all cases (right panel). Again the "true" input value is given by the blue dot. It can be seen that, using such Bayes Linear emulation analysis, we can rule out most of the input space, and that the green area of the right panel of figure 7 is consistent with the true input value. It is also clear that information from the variance of the Birth-Death model was vital in this process. We now go on to apply these techniques to the more complex Prokaryotic model.

7 Application to the Prokaryotic Auto-regulatory Gene Network

7.1 Variance and Mean Emulation

Similar to the above analysis of the simple Birth-Death model, we now apply our two stage Bayes Linear update strategy to the more complex Prokaryotic Auto-regulatory Gene Network described in section 5.4. This model is of far more interest, as it exhibits complex behaviour and is used to study features of gene regulation that occur within cells.

We restrict our attention to a 2-dimensional surface within the full 8-dimensional input space of the Prokaryotic model, defined by parameterising the inputs as: $x = (1, 10, 0.01, 10, 1, \lambda, \mu, 0.01)$, with $0 < \lambda < 7$ and $0.05 < \mu < 0.4$. Hence we explore the rate parameters corresponding to the reverse dimerisation reaction and the mRNA degradation discussed in section 5.4, and leave a full study of the input space to future work. As in section 6.1, we begin by using the Gillespie algorithm to perform a d = 15 point maximin latin hypercube design, with n = 40 repetitions at each input point. We use these runs to construct mean and variance emulators for the model output at 3 different time points $T = (2, 8, 18)$.

The model produces 5 outputs at each time point, corresponding to the 5 chemical species $g.P_2, g, r, P$ and P_2, that feature in the network. This means that we construct 5 variance and 5 mean emulators at each time point. Full analytic solutions to this network are not available, so we leave discussion of diagnostics of stochastic models such as this to a future work.

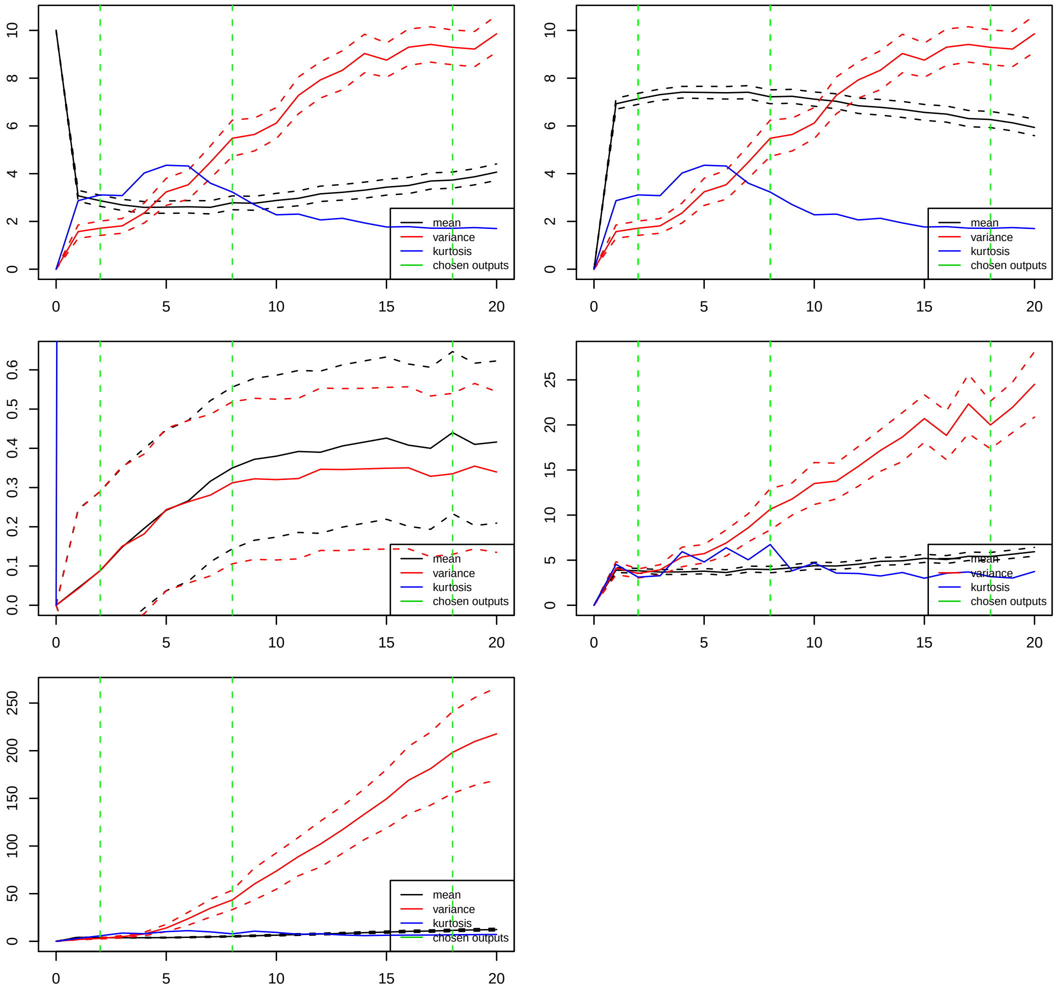

Figure 8: The (simulated) observed data for the $g.P_2, g, r, P$ and P_2 against time (panels from left to right, top to bottom). Means and variances are in black and red (solid lines) with the dotted lines representing observational errors plus model discrepancy. The green lines show the time points considered, and the sample kurtosis is in blue.

7.2 Implausibility

We simulate observed data, as in section 6.2, by evaluating the Prokaryotic network model for a chosen input with $\lambda = 2$ and $\mu = 0.15$, using the Gillespie algorithm with 500 repetitions. We took the sample mean and sample variance to be the observed mean and variance data z, and the sample error on these quantities was used as the observational error e, and we also included a small model discrepancy d on each of the mean and variance quantities, as in section 6.2. This observed data for the $g.P_2, g, r, P$ and P_2 against time, is shown in figure 8. Means and variances are in black and red (solid lines) with the dotted lines representing observational errors plus model discrepancy. The green lines show the time points considered. Note that the first two panels (top row) show the $g.P_2$ and g molecules which are related by a conservation law (which can be derived from the reaction matrix A). This will be seen in the implausibility plots.

As we now have 5 chemicals, 3 time points and a choice of variance or mean emulator, there are now a total of 30 possible implausibility plots that would be produced using equation 57. For each chemical, we take the variance implausibility and maximise it over all 3 time points. This produces the 5 plots show in the left column of figure 9. We then do the same for the mean implausibility, which produces the 5 plots in the middle column of figure 9. Maximising over both the mean and variance plots, produces the right column.

The 5 plots in the right column of figure 9 show what parts of the input space we learn about from measuring the 5 chemicals that feature in the network. The left and right columns show whether this information comes from the variance or the mean of the process. We can see that the first two chemicals $g.P_2$ and g give exactly the same information: this has to be true as they are linked via a conservation law. They are highly informative and we gain information from the variance and the mean. The mRNA molecule r is relatively uninformative about these inputs even though the second input μ controls its degradation. The forth and fifth chemicals, the proteins P and P_2, are both highly informative, with information coming from the mean for P_2, but both mean and variance for P. Considerations of this form regarding the possible information gaining from the measurement of each chemical species would help inform the design of some future biological experiment.

If we maximise the implausibility $\mathcal{I}(x)$ over all 30 possible plots, we obtain the plot shown in figure 10 (left panel), where again red/orange areas (with $\mathcal{I}(x) > 3$) would be discarded as implausible and the blue point represents the "true" inputs used to generate the observed data. This represents all the information we learn from all 5 chemicals over 3 time points, using both mean and variance information. It can be seen that the analysis work well: we can resolve much of the uncertainty as regards the inputs, and the non-implausible area (yellow/green) contains the true value. The right panel of figure 10 shows the same maximised implausibility plot, but for a case where we used a far greater number of model evaluations. Here, 20 input points were used in the design, with 200 repetitions at each point. This resulted in emulator uncertainties of comparable size to the model discrepancy and observational errors. This shows how much more of the input space we could resolve if allowed significantly more computer run time. A solution to this is to use a iterative strategy, and to perform more runs inside the current non-implausible region such as is employed in Craig et al. (1997) and Vernon et al. (2010).

8 Conclusions

We have developed a novel technique for analysing the input space of a stochastic computer model, based on a two stage Bayes Linear update emulation strategy that allows emulation of both variance and mean surfaces over the input space x. This is combined with the use of implausibility measures to discard regions of the input space inconsistent with observed data. This methodology was applied to a simple, analytically tractable Birth-Death model, and then to a more complex Prokaryotic Autoregulatory Gene Network of current interest to the modelling community (Wilkinson (2006)). We have restricted our analysis to the emulation of the means and variances of individual univariate quantities at fixed time points, as this relates to one type of experimental technique whereby the contents of hundreds of cells can be measured at a single time point, but the cells are simultaneously destroyed.

There are many future directions such analysis can take including: multivariate emulation including the emulation of the full covariance structure of the model itself; incorporation of the incomplete, infrequent but essentially dynamic data that is become possible using certain fluorescence experimental techniques; design of experiments including choosing the number of repetitions at each input point (see for example Boukouvalas et al. (2009)); and the iterative reduce of the input space of which the work presented here can be viewed as the first iteration or wave (Bower

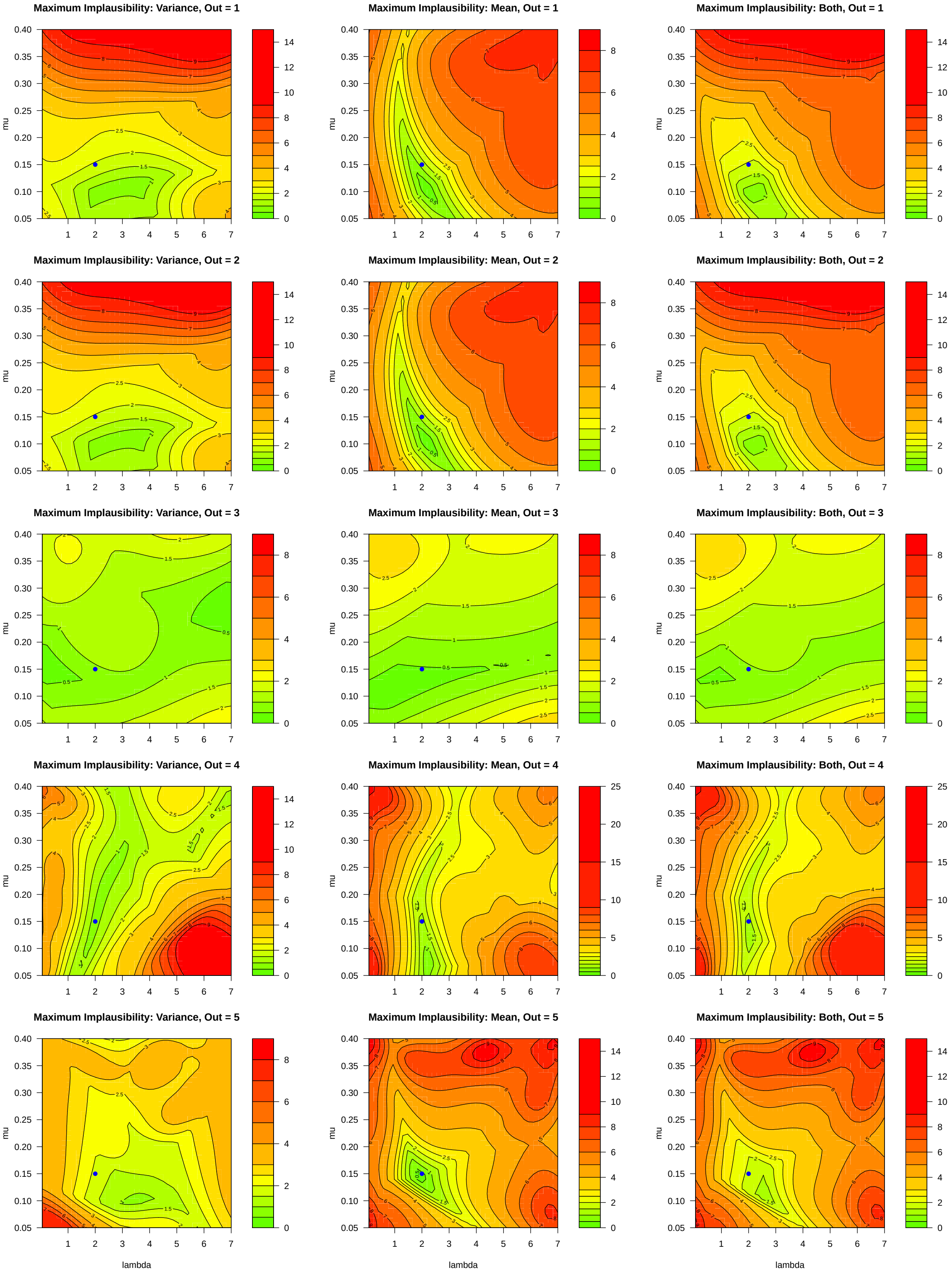

Figure 9: Implausibility plots maximised over the 3 time points for the variance (left column), mean (middle column) and maximised over both variance and mean (right column). The 5 rows represent the 5 chemical species $g.P_2$, g, r, P and P_2. Red indicates input space that we discard as implausible. The “real” input used to generate the observed data is given by the blue point.

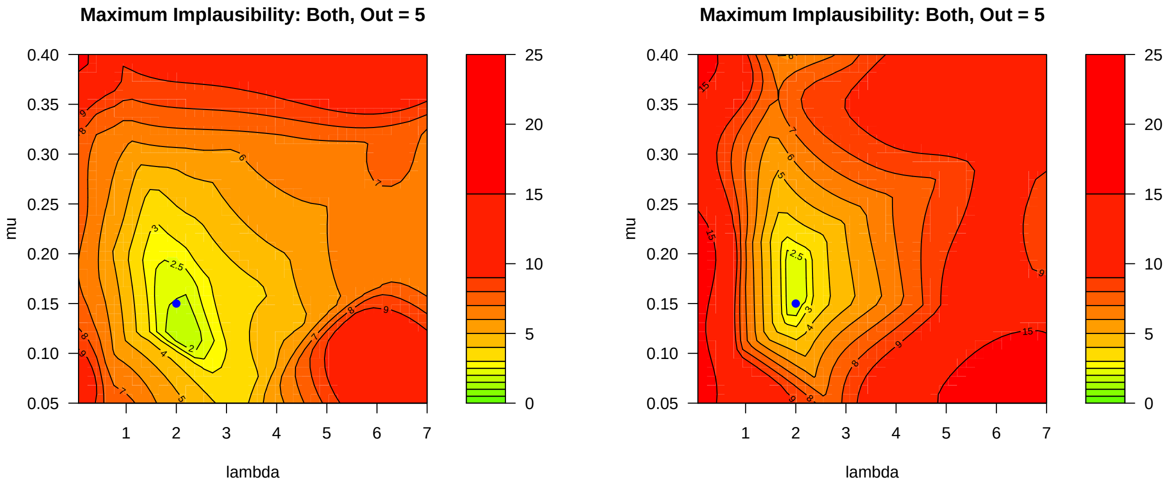

Figure 10: The implausibility maximised over all 3 time points, 5 chemical species and both mean and variance choices for the original design of 15 inputs with 40 repetitions (left panel). Again red/orange areas (with $\mathcal{I}(x) > 3$) would be discarded as implausible and the blue point represents the "true" inputs used to generate the observed data. The right panel shows the same plot, but generated using far more model runs: a design of 20 input points with 200 repetitions at each point.

et al. (2009); Vernon et al. (2010)).

Acknowledgements

IRV and MG acknowledge the support of the Basic Technology initiative as part of the Managing Uncertainty for Complex Models project. IRV acknowledges the support of an EPSRC mobility fellowship.

References

Boukouvalas, A., Cornford, D., and Stehlk, M. (2009), "Approximately Optimal Experimental Design for Heteroscedastic Gaussian Process Models," *MUCM Technical Report 09/06*.

Bower, R., Vernon, I., Goldstein, M., et al. (2009), "The Parameter Space of Galaxy Formation," *MUCM Technical Report 10/02, to appear in Mon.Not.Roy.Astron.Soc.*

Craig, P. S., Goldstein, M., Rougier, J. C., and Seheult, A. H. (2001), "Bayesian forecasting for complex systems using computer simulators," *Journal of the American Statistical Association*, 96, 717–729.

Craig, P. S., Goldstein, M., Seheult, A. H., and Smith, J. A. (1996), "Bayes linear strategies for history matching of hydrocarbon reservoirs," in *Bayesian Statistics 5*, eds. Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., Oxford, UK: Clarendon Press, pp. 69–95.

— (1997), "Pressure matching for hydrocarbon reservoirs: a case study in the use of Bayes linear strategies for large computer experiments," in *Case Studies in Bayesian*

Statistics, eds. Gatsonis, C., Hodges, J. S., Kass, R. E., McCulloch, R., Rossi, P., and Singpurwalla, N. D., New York: Springer-Verlag, vol. 3, pp. 36–93.

Cumming, J. A. and Goldstein, M. (2009), "Bayes linear uncertainty analysis for oil reservoirs based on multiscale computer experiments," in *Handbook of Bayesian Analysis*, eds. OHagan, A. and West, M., Oxford, UK: Oxford University Press.

Gillespie, D. T. (1977), "Exact stochastic simulation of coupled chemical reactions," *Journal of Physical Chemistry*, 81, 2340–2361.

Goldstein, M. and Rougier, J. C. (2009), "Reified Bayesian modelling and inference for physical systems (with Discussion)," *Journal of Statistical Planning and Inference*, 139, 1221–1239.

Golightly, A. and Wilkinson, D. J. (2005), "Bayesian Inference for Stochastic Kinetic Models Using a Diffusion Approximation," *Biometrics*, 61, 781–788.

Kennedy, M. C. and O'Hagan, A. (2001), "Bayesian calibration of computer models," *Journal of the Royal Statistical Society, Series B*, 63, 425–464.

O'Hagan, A. (2006), "Bayesian analysis of computer code outputs: A tutorial," *Reliability Engineering and System Safety*, 91, 1290–1300.

Rougier, J. (2009), "Formal Bayes methods for model calibration with uncertainty," in *Applied Uncertainty Analysis for Flood Risk Management*, eds. Beven, K. and Hall, J., Imperial College Press / World Scientific.

Vernon, I., Goldstein, M., and Bower, R. (2010), "Galaxy Formation: a Bayesian Uncertainty Analysis," *MUCM Technical Report 10/03, submitted to Bayesian Analysis.*

Wilkinson, D. J. (2006), *Stochastic Modelling for Systems Biology*, Taylor and Francis Group, LLC: Chapman and Hall.

A Mass Action Stochastic Kinetics.

We consider the general system which has k species $Y_1, Y_2, ..., Y_k$ and r reactions $R_1, R_2, ..., R_r$, and we assume the structure of the Reaction network can be described by a Petri Net (C, T, U, V, Y) where $C = (Y_1, .., Y_k)^T$ and $T = (R_1, ..., R_r)^T$. We assert that each reaction has a reaction rate or *Reaction Hazard* $h_i(y, x_i) = \lambda_i$, where x_i is the *Reaction Rate Constant* and $y = (y_1, ..., y_k)$ is the current state of the system, and that for reaction type R_i, *in the absence of any other reaction* the time to the next R_i reaction occurring is $Exp(h_i(y, x_i))$. When a reaction R_i occurs it changes the state vector y according to $y \rightarrow y + A_{(i)}$, where $A_{(i)}$ is the ith row of the reaction matrix A. Note that the notation of this appendix is distinct to and should not be confused with that of section 2.

The possible types of *Reaction Hazards* $h_i(y, x_i)$ are defined as follows, using mass-action kinetics, which states that the hazard of each reaction type will be proportional to the number of possible combinations of reactant molecules:

1. Zeroth Order Reaction (e.g. Immigration Process):

$$R_i : \qquad \emptyset \longrightarrow B \tag{78}$$

   has a constant reaction rate so:

$$h_i(y, x_i) = x_i. \tag{79}$$

2. First Order Reactions:
$$R_i: \qquad Y_j \;\longrightarrow\; B \tag{80}$$
Reaction proportional to the number of molecules of Y_j so reaction hazard is:
$$h_i(y, x_i) \;=\; x_i y_j. \tag{81}$$
3. Second Order Reactions (two types):
Type 1: different chemicals combining.
$$R_i: \qquad Y_j + Y_k \;\longrightarrow\; B \tag{82}$$
Reaction proportional to the number of molecules of Y_j and Y_k so reaction hazard is:
$$h_i(y, x_i) \;=\; x_i y_j y_k. \tag{83}$$
Type 2: Same chemical combining.
$$R_i: \qquad 2Y_j \;\longrightarrow\; B \tag{84}$$
Reaction proportional to the number of possible *pairs* of Y_j molecules which is $\binom{y_j}{2}$ so reaction hazard is:
$$h_i(y, x_i) \;=\; x_i \frac{y_j(y_j - 1)}{2}. \tag{85}$$

A very useful quantity is the *Combined Reaction Hazard* $h_0(y, x)$ defined as:
$$h_0(y, x) \;=\; \sum_{i=1}^{v} h_i(y, x_i), \tag{86}$$
where $x = (x_1, x_2, ..., x_r)$.

This is all we need to simulate a Stochastic Chemical Reaction Network using the Gillespie algorithm described in the next section.

B The Gillespie Algorithm.

We can now simulate the behaviour of a Stochastic Chemical Reaction Network described by any Petri Net (C, T, U, V, Y) with Reaction Rate Constants $x = (x_1, x_2, ..., x_r)$ using the Gillespie Algorithm as follows:

1. Start the system at time $t = 0$, with Reaction Rate constants $x = (x_1, ..., x_r)$ and initial numbers of molecules for each species $y = (y_1, ..., y_k)$.
2. For each $i = 1, 2, ..., r$ calculate $h_i(y, x_i)$ based on the current state y.
3. Calculate the Combined Reaction Hazard $h_0(y, x) = \sum_{i=1}^{v} h_i(y, x_i)$ for the current state y.
4. Simulate the time to the next event t' as a $Exp(h_0(y, x))$ random quantity.
5. Put $t = t + t'$.
6. Simulate the reaction index, j, of the *type* of reaction that has occurred as a discrete random quantity with probabilities $P(j = i) = h_i(y, x_i)/h_0(y, x)$, $i = 1, 2, ..., r$.
7. Update the state vector y according to reaction R_j. That is, put the new $y = y + S^{(j)}$, where $S^{(j)}$ denotes the jth column of the Stoichiometry matrix $S = A^T$.
8. Save values of x and t.
9. If $t < T_{max}$, return to step 2.

All runs of the model discussed in sections 5, 6 and 7 were performed using the above algorithm.