Matching Models Across Abstraction Levels with Gaussian Processes

Giulio Caravagna[1(✉)], Luca Bortolussi[2,3,4], and Guido Sanguinetti[1]

[1] School of Informatics, University of Edinburgh, Edinburgh, UK
giulio.caravagna@ed.ac.uk
[2] DMG, University of Trieste, Trieste, Italy
[3] ISTI-CNR, Pisa, Italy
[4] MOSI, Department of Informatics, Saarland University, Saarbücken, Germany

Abstract. Biological systems are often modelled at different levels of abstraction depending on the particular aims/resources of a study. Such different models often provide qualitatively concordant predictions over specific parametrisations, but it is generally unclear whether model predictions are quantitatively in agreement, and whether such agreement holds for different parametrisations. Here we present a generally applicable statistical machine learning methodology to automatically reconcile the predictions of different models across abstraction levels. Our approach is based on defining a correction map, a random function which modifies the output of a model in order to match the statistics of the output of a different model of the same system. We use two biological examples to give a proof-of-principle demonstration of the methodology, and discuss its advantages and potential further applications.

Keywords: Computational abstraction · Emulation · Gaussian Processes · Heteroschedasticity

1 Introduction

Computational modelling in the sciences is founded on the notion of abstraction, the process of identifying and representing mathematically the salient features and interactions of a real system. Abstraction is a human led and interdisciplinary activity: many factors influence the decision of which features/interactions are eventually represented in the abstracted model, including the specialist interests of the scientists formulating the model, as well as computational constraints on the level of detail which can feasibly be implemented on the available hardware. Such factors inevitably vary between different research groups and at different times, leading to a proliferation of different models representing the same underlying phenomena.

GC and GS gratefully acknowledge support from the European Research Council under grant MLCS306999. LB acknowledges partial support from the EU project QUANTICOL, 600708, and by FRA-UniTS. We thank Dimitris Milios for useful discussions and for providing us with the MATLAB for heteroscedastic regression.

E. Bartocci et al. (Eds.): CMSB 2016, LNBI 9859, pp. 49–66, 2016.
DOI: 10.1007/978-3-319-45177-0_4

Systems biology is a prominent field where models with different level of abstraction coexist. As an example, consider the process of gene expression, whereby genetic material stored in the DNA is transcribed into messenger RNA and eventually translated into protein. At the highest level of abstraction, which is frequently employed when studying high-throughput data, the process may be considered as a black box, and one may only be interested in characterising the statistical structures underlying observed levels of gene expression in different conditions [8]. Alternatively, one may want to mechanistically represent the dynamics of some of the steps involved in the process. The choice of which steps to include is again largely driven by extrinsic factors: examples in the literature range from highly detailed models where the synthesis of mRNA/protein is modelled through the binding and elongation of polymerase/ribosomes, to models where protein production is modelled as a single reaction with first order kinetics [1,10].

Representing the same physical process by multiple models at different levels of abstraction immediately engenders the question of how different model outputs can be reconciled. As the models all represent the same underlying physical system, it can be plausibly assumed that such models will agree at least qualitatively for suitable parametrisations. In general, however, models may not agree quantitatively, and their discrepancy may be a function of the parameters. Understanding and quantifying such discrepancies would often be very valuable: first of all, it can shed light on how simplifications within models affect predictions, and secondly it may open the opportunity to construct computationally smaller surrogates of complex models. Such surrogates can be precious when modelling requires intrinsically computationally intensive tasks like inference, as they have less parameters.

In this paper, we approach the problem of reconciling models from a statistical machine learning angle. We start by sampling a sparse subset of the parameter space over which we evaluate the models' outputs (generally by simulation). These evaluations are used as a *training set* to learn a *correction map* via a non-parametric regression approach based on Gaussian Processes. We show that our approach yields a consistent stochastic equivalence between models, which provably reconciles the predictions of the two models up to the second moment. We demonstrate the approach on two biological examples, showing that it can lead to non-trivial insights into the structure of the models, and provide an efficient way to simulate a complex model via a simpler model. Correction maps could be used to reduce the complexity of some common problems that we briefly discuss in the paper, e.g., model selection, synthesis and parameter estimation.

The rest of the paper is organised as follows: we start by giving a high level description of the problem and how we attack it. This is followed by a formal definition and a thorough illustration of the proposed solution, discussing its desirable theoretical properties. We then demonstrate the approach on two proof of principle examples, showing the potential for practical application of the method. We conclude the paper by discussing the relationship of our approach to existing ideas in model reduction and statistical emulation, as well as some possible extensions of our results.

2 Problem Definition

High level description of the approach. In this paper we consider the problem of performing analyses which require exhaustive sampling of a model's outputs, M, the dynamics of which are expensive to compute. We are not interested in the origin of such a complexity, and *we assume* to be given an *abstraction/surrogate* of this model, m, which is reasonably less challenging to analyze[1]. For this reason, *we want to investigate a general methodology to use* m *as a reliable proxy to get statistics over* M. Possible applications of this framework, which we discuss in Sect. 4, regard *model selection and synthesis*, *inference*, and *process equivalence/control.*

In general, we will assume both models to be stochastic processes, e.g. continuous time Markov Chains (CTMCs). Furthermore, we assume that the highly detailed model M and the reduced model m share some parameters $\boldsymbol{\theta}_{\mathsf{m}}$ and some observables/state variables X, but the large model will generally have many more state variables and parameters. In general we can compute some statistics of the shared state variables X (e.g. mean), and that such computation will be considerably more expensive using the detailed model M.

As both models are devised as abstractions of the same physical system, it is not unreasonable to assume that the expected values of the shared variables will be similar for some parametrisations of the models. However, it is in general not the case that the *distribution* of the shared variables implied by the two models will be the same, and, as we vary the shared parameters $\boldsymbol{\theta}_{\mathsf{m}}$, even the expected values may show non-negligible discrepancies. Our aim is to develop a generally applicable machine learning approach to correct the output of the reduced model, in order to match the distribution of the larger model. This has strong practical motivations: one of the primary uses of models in general is to test hypotheses statistically against observed data, and it is therefore essential to capture as accurately as possible the implied variability on observable variables.

The strategy we will follow is simple: we start by sampling values of the shared parameters $\boldsymbol{\theta}_{\mathsf{m}}$, and compute the first two statistics of the observed variables as implied by both the large and reduced models (by simulation). In general, one may expect the variance implied by the larger model to be larger, as a more detailed model will imply more stochastic steps. We can therefore correct the first two statistics (mean and variance) of the reduced model output by adding a random function of the shared parameters $\boldsymbol{\theta}_{\mathsf{m}}$, which can be learned by rephrasing it as a regression task. We will work with heteroschedastic Gaussian Processes [5].

[1] M could be complex to analyze either because of its structure, e.g., it might have many variables, or its numerical hurdles, e.g., the degree of non-linearity or parameters stiffness. For similar reasons, we do not care whether m is has been derived by means of independent domain-knowledge or automatic techniques.

2.1 Formal Problem Statement

We assume to be given two models of the same underlying physical system:

– a highly detailed model M, with state variables $\mathcal{Y}$ and parameters $\boldsymbol{\theta}_{\mathsf{M}}$.
– a reduced model m, with state variables $\mathcal{X}$ and parameters $\boldsymbol{\theta}_{\mathsf{m}}$.

We will have $|\mathcal{Y}| \gg |\mathcal{X}|$ and $|\boldsymbol{\theta}_{\mathsf{M}}| \gg |\boldsymbol{\theta}_{\mathsf{m}}|$.

Assumptions. In general, the problem we want to tackle draws immediate connection to that of using m as a *statistical emulation* of M. However, to exploit solutions from *regression analysis* and *machine learning*, we make further assumptions and discuss their limitations thorough the paper.

1. (*Observables*) we assume that it exists a non-empty set of state variables (or, more generally, observables) X, common to both models, that is sufficient to compute our statistics of interest.
2. (*Parameters*) we assume that model M is fully parametrized by $\boldsymbol{\theta}_{\mathsf{M}}$, a vector of real-valued parameters that breaks down as $\boldsymbol{\theta}_{\mathsf{M}} = [\boldsymbol{\theta}_{\mathsf{m}} \quad \boldsymbol{\theta}_{\mathsf{f}}]^{\top}$, with $\boldsymbol{\theta}_{\mathsf{m}}$ shared between models m and M. Here, we assume that m is fully parametrized[2] by $\boldsymbol{\theta}_{\mathsf{m}}$, which ranges in a domain set Θ. We term $\boldsymbol{\theta}_{\mathsf{f}}$ free parameters in M, given $\boldsymbol{\theta}_{\mathsf{m}}$. We further assume to have a probability density $p(\boldsymbol{\theta}_{\mathsf{f}})$ for the free parameters, and a probability density $p(\boldsymbol{\theta}_{\mathsf{m}})$ for the shared parameters, encoding our knowledge about them.
3. (*Sampling*) we assume that, for every parametrization, each model's dynamics is computable, i.e. it can be simulated.

In this work, we will consider correction maps conditioned to a reference statistic of interest, in the following sense.

Definition 1 (Statistic). *A statistic* η *is* any observable that we can compute from one, or from an ensemble of simulations of a model. *We denote its* estimator *computed from model* q *with parameters* x *as* $\mathsf{q}_{\hat{\eta}}(\mathsf{x})$*, and its true value* $\mathsf{q}_{\eta}(\mathsf{x})$*.*

Valid examples of such statistics are, e.g., a single value or the expectation of a variable in X, the satisfiability of a temporal logical formula depending on variables X that could be model-checked, etc. The richer the estimator, the higher number of samples are required for the estimator to converge to the true statistics. We will make use of estimators that are *consistent* in the following sense: $\mathsf{q}_{\hat{\eta}}(\mathsf{x}) \rightarrow \mathsf{q}_{\eta}(\mathsf{x})$ as the number of samples goes to infinity.

Definition 2 (Correction map). *We define* an ϵ-correction map $\mathcal{M}_{\eta} : \Theta \rightarrow \mathbb{R}^{w}$ *for a* statistics η *to be a function that for any* $\boldsymbol{\theta}_{\mathsf{m}} \in \Theta$*, satisfies*

$$\hat{\mathsf{M}}_{\eta}(\boldsymbol{\theta}_{\mathsf{m}}) \triangleq \mathsf{m}_{\eta}(\boldsymbol{\theta}_{\mathsf{m}}) + \mathcal{M}_{\eta}(\boldsymbol{\theta}_{\mathsf{m}}) \textit{ and } \int_{\Theta} \| \hat{\mathsf{M}}_{\eta}(\boldsymbol{\theta}_{\mathsf{m}}) - \mathbb{E}_{\boldsymbol{\theta}_{\mathsf{f}}}[\mathsf{M}_{\eta}(\boldsymbol{\theta}_{\mathsf{m}})] \|_{2} \, p(\boldsymbol{\theta}_{\mathsf{m}}) d\boldsymbol{\theta}_{\mathsf{m}} < \epsilon \quad (1)$$

[2] In principle, even m might have a set of free variables, with respect to M. However, as we have full control over that model, we could assume a parametrization of such variables and all what follows would be equivalent.

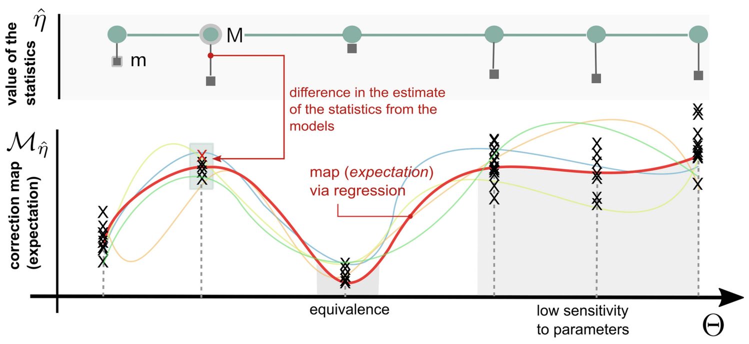
value of the statistics $\hat{\eta}$
m
M
difference in the estimate of the statistics from the models
correction map (expectation) $\mathcal{M}_{\hat{\eta}}$
map (expectation) via regression
equivalence
low sensitivity to parameters
Θ

Fig. 1. Correction maps as regression problems. From sampled estimates of a statistics $\hat{\eta}$ computed over domain Θ by models M and m, we can define a correction $\hat{\mathsf{M}}_{\hat{\eta}}(x) \triangleq \mathsf{m}_{\hat{\eta}}(x) + \mathcal{M}_{\hat{\eta}}(x)$ from their difference. According to the variables/parameters involved, we collect multiple values of such correction; see also Fig. 2. Then, a regression over such values leads to a correction model for M's prediction, from m's ones; this differs from standard emulation as we retain the mechanistic description of the system in m. A map is modeled as a random function via Gaussian Processes (GPs) with heteroschedastic variance; GPs induce a distribution over functions (colored lines), and the map will be the expectation (red line). Maps allow to detect regions of Θ where the predictions of the models are in agreement, $\mathcal{M}_{\hat{\eta}} \to 0$, or roughly constant (i.e., at low sensitivity). (Color figure online)

where $\epsilon > 0$ *is the precision, and* $\mathbb{E}_{\boldsymbol{\theta}_\mathsf{f}}[\mathsf{M}_\eta(\boldsymbol{\theta}_\mathsf{m})] = \int \mathsf{M}_\eta(\boldsymbol{\theta}_\mathsf{m}; \boldsymbol{\theta}_\mathsf{f}) p(\boldsymbol{\theta}_\mathsf{f}) d\boldsymbol{\theta}_\mathsf{f}$ *is the* expectation of the statistics *computed from* M, *with respect to its free parameters* $\boldsymbol{\theta}_\mathsf{f}$. $\hat{\mathsf{M}}$ *is our* corrected prediction *of* M.

Thus, $\mathcal{M}_{\hat{\eta}}$ can correct the outcomes of η assessed over m, $\mathsf{m}_\eta(\boldsymbol{\theta}_\mathsf{m})$, to match (with a given precision) those that we would have by computing the statistics over M. Notice that the corrected outcome has no more dependence from the free parameters, as this is a correction in expectation with respect to $\boldsymbol{\theta}_\mathsf{f}$.

Notice that the correction is a w-dimensional vector, and hence $\|\cdot\|_2$ is used as distance metric, and that the term ϵ allows for tolerance in the correction's precision. It is easy to define the optimal, zero-error, correction map:

$$\mathcal{M}_\eta{}^\star(\boldsymbol{\theta}_\mathsf{m}) \triangleq \mathbb{E}_{\boldsymbol{\theta}_\mathsf{f}}[\mathsf{M}_\eta(\boldsymbol{\theta}_\mathsf{m})] - \mathsf{m}_\eta(\boldsymbol{\theta}_\mathsf{m}) . \qquad (2)$$

However, the correction function $\mathcal{M}_\eta{}^\star(\boldsymbol{\theta}_\mathsf{m})$ is impossible to compute exactly, as we cannot compute neither M_η nor its marginalisation over $\boldsymbol{\theta}_\mathsf{f}$. Hence, we will learn an approximation of $\mathcal{M}_\eta{}^\star(\boldsymbol{\theta}_\mathsf{m})$ trying to keep its error low so to satisfy Definition 2. We turn this problem into a *regression task*, as graphically explained in Figs. 1 and 2, and exploit Gaussian Processes.

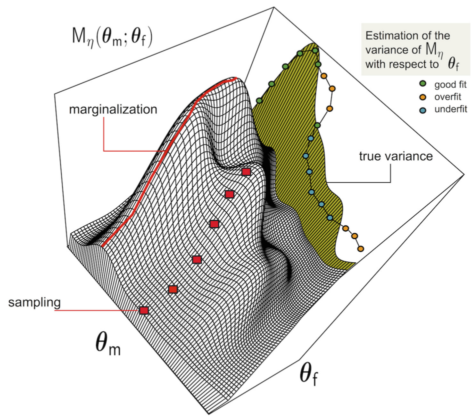
$M_\eta(\theta_m;\theta_f)$
Estimation of the variance of M_η with respect to θ_f
● good fit
● overfit
● underfit
marginalization
true variance
sampling
θ_m
θ_f

Fig. 2. For any pair of $\boldsymbol{\theta}_\mathsf{m}$ and $\boldsymbol{\theta}_\mathsf{f}$ we can compute the statistics for M, $\mathsf{M}_\eta(\boldsymbol{\theta}_\mathsf{m};\boldsymbol{\theta}_\mathsf{f})$. Since we do regression over $\boldsymbol{\theta}_\mathsf{m}$, we model the relation between such values and $\boldsymbol{\theta}_\mathsf{f}$ as the variance of a random variable, indexed by $\boldsymbol{\theta}_\mathsf{m}$, whose samples are the values, as a function of $\boldsymbol{\theta}_\mathsf{f}$. Marginalization is the exponential strategy that estimates such variance correctly; all downsampling strategies possibly over or under fitting. Accounting for the relation between this variance and $\boldsymbol{\theta}_\mathsf{m}$ can be achieved by heteroschedastic regression.

3 Learning the Correction Map

In this section we will present our machine learning strategy in more detail. We consider the case of a scalar statistics, as w-dimensional ones can be treated by solving w independent learning problems.

3.1 Marginalising $\boldsymbol{\theta}_\mathsf{f}$

In order to evaluate (2), we need to be able to compute or approximate the value $\mathbb{E}_{\boldsymbol{\theta}_\mathsf{f}}[\mathsf{M}_\eta(\boldsymbol{\theta}_\mathsf{m})]$ with respect to the free parameters of M, for a any given $\boldsymbol{\theta}_\mathsf{m}$. As this integral cannot be treated analytically or numerically, due to the large dimensionality involved (the cost is exponential in $|\boldsymbol{\theta}_\mathsf{f}|$), we will resort to statistical approximations. Before discussing them, let us comment on the distribution $p(\boldsymbol{\theta}_\mathsf{f})$, which is an input for our method. In particular, this distribution encodes our knowledge on the more plausible values of the free parameters. In case we have no information, we can choose an uniform distribution. On the other side of the spectrum, we may know the true value $\boldsymbol{\theta}_\mathsf{f}^*$ of $\boldsymbol{\theta}_\mathsf{f}$, and choose a delta Dirac distribution, which will dramatically simplify the evaluation of the integral. In this case, we can evaluate (2) as

$$\mathcal{M}_\eta(\boldsymbol{\theta}_\mathsf{m}) \triangleq \mathsf{M}_\eta(\boldsymbol{\theta}_\mathsf{m};\boldsymbol{\theta}_\mathsf{f}^*) - \mathsf{m}_\eta(\boldsymbol{\theta}_\mathsf{m})\,, \qquad (3)$$

Moreover, the approximation, $\int \mathsf{M}_\eta(\boldsymbol{\theta}_\mathsf{m};\boldsymbol{\theta}_\mathsf{f})p(\boldsymbol{\theta}_\mathsf{f})d\boldsymbol{\theta}_\mathsf{f} \approx \mathsf{M}_\eta(\boldsymbol{\theta}_\mathsf{m};\boldsymbol{\theta}_\mathsf{f}^*)$ is appropriate when the distribution $p(\boldsymbol{\theta}_\mathsf{f})$ is tightly concentrated around its mode $\boldsymbol{\theta}_\mathsf{f}^*$.

In general, however, when $p(\boldsymbol{\theta}_\mathsf{f})$ does not have this special form, we can resort to downsampling $\mathbb{E}_{\boldsymbol{\theta}_\mathsf{f}}[\mathsf{M}_\eta(\boldsymbol{\theta}_\mathsf{m})]$, by generating k samples ${\boldsymbol{\theta}_\mathsf{f}}^{(1)}, \ldots, {\boldsymbol{\theta}_\mathsf{f}}^{(k)}$ and approximating $\mathbb{E}_{\boldsymbol{\theta}_\mathsf{f}}[\mathsf{M}_\eta(\boldsymbol{\theta}_\mathsf{m})] \approx \frac{1}{k}\sum_j \mathsf{M}_\eta(\boldsymbol{\theta}_\mathsf{m}; {\boldsymbol{\theta}_\mathsf{f}}^{(j)})$. In the following, however, we will not necessarily aggregate the values $\mathsf{M}_\eta(\boldsymbol{\theta}_\mathsf{m}; {\boldsymbol{\theta}_\mathsf{f}}^{(j)})$, and treat them individually to better account for the variance in the observed predictions.

3.2 Gaussian Processes

We will solve the learning problem resorting to Gaussian Process (GP) regression [12]. GPs are random functions, i.e. probability distributions over a function space, in our case functions $f : \Theta \to \mathbb{R}$, with the property that any finite dimensional projection $f(\boldsymbol{\theta}_1), \ldots, f(\boldsymbol{\theta}_k)$ is a multidimensional Gaussian random variable. It follows that GP are defined by a mean function $\mu(\boldsymbol{\theta})$, returning the mean at any point in Θ, and by a covariance or kernel function $k(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$, for giving the covariance between any pair of points in Θ. GP can be used to solve regression tasks in a Bayesian setting. The idea is as follows: we put a GP prior on the space of functions $\{f \mid f : \Theta \to \mathbb{R}\}$, typically by assuming constant zero mean and fixing a kernel function[3], and then consider the posterior distribution given a set of observations $Y = \{y^{(i)}\}_I$, $i \in I$, at input points $X = \{\boldsymbol{\theta}_\mathsf{m}^{(i)}\}$. If we assume that $y^{(i)} = f(\boldsymbol{\theta}_\mathsf{m}^{(i)}) + \epsilon_i$, with ϵ_i a zero mean Gaussian noise term with variance σ^2, then we obtain that the posterior distribution given the observed data is still a GP, with mean and kernel functions that can be obtained analytically, see [12] for further details. GP regression is essentially the same if the observation noise σ^2 is different at different input points, i.e. $\sigma^2 = \sigma(\boldsymbol{\theta}_\mathsf{m}^{(i)})^2$, in which case we talk about heteroschedastic regression.

3.3 The Regression Task

Let $\boldsymbol{\theta}_\mathsf{m}^{(i)}$ for some index set $i \in I$ be the *input space*, and $\{\langle \boldsymbol{\theta}_\mathsf{m}^{(i)}, y^{(i)} \rangle\}_I$ our *training points*. In case we use Eq. (3) to evaluate the correction map, each $y^{(i)}$ is a scalar value, and a standard regression schema based on Gaussian Processes can be used. In that case we assume samples of the correction map y to be observations from a random variable centered at a value given by the *latent function*

$$y^{(i)} \sim \mathcal{N}(\mathcal{M}_\eta(\boldsymbol{\theta}_\mathsf{m}^{(i)}), \sigma^2)\,. \tag{4}$$

In this standard Gaussian Processes regression the noise model in the observations is assumed to be a constant σ^2 for all sampled points.

In the more general case we work with downsampling solutions that exploit k samples for the free variable, ${\boldsymbol{\theta}_\mathsf{f}}^{(1)}, \ldots, {\boldsymbol{\theta}_\mathsf{f}}^{(k)}$. In that case, we have k correction values per training point, $\left\{\langle \boldsymbol{\theta}_\mathsf{m}^{(i)}, [y^{(i,1)} \cdots y^{(i,k)}]^\top \rangle\right\}_I$, that we can use in a straightforward way to reduce to the above schema, or to estimate the variance of M conditioned to its free variables. In these cases, the training set is

[3] In this work, we use the classic Gaussian kernel fixing hyperparameters by maximising the type-II likelihood; see [12].

$\{\langle \boldsymbol{\theta}_{\mathsf{m}}^{(i)}, \overline{y}^{(i)}\rangle\}_I$, namely we do regression on the point-wise expectation of the correction (i.e., $\overline{y}^{(i)} = \frac{1}{k}\sum_{j=1}^{k} y^{(i,j)}$).

Estimator 1 (Empirical $\overline{\sigma}$-estimator). *Set $\overline{\sigma}$ to the empirical estimate of the variance* across all correction values *to exploit the schema in Eq. (4) with $\sigma^2 \triangleq \overline{\sigma}$.*

Besides the simple first case, it is more interesting to account for a *model of the variance in the observations of the predictions from a model*; we discuss how this could be done in two ways.

Estimator 2 (Point-wise σ-estimator). *Let $\sigma(\cdot)$ the* empirical estimator *of the variance of the correction values, per training-point*

$$\sigma(\boldsymbol{\theta}_{\mathsf{m}}^{(i)}) \triangleq \mathsf{Var}[y^{(i,1)}, \ldots, y^{(k,1)}]\,, \tag{5}$$

then define a model of the variance as a point-wise function of the regression parameter, that is

$$y^{(i)} \sim \mathcal{N}\Big(\mathcal{M}_\eta(\boldsymbol{\theta}_{\mathsf{m}}^{(i)}), \sigma(\boldsymbol{\theta}_{\mathsf{m}}^{(i)})^2\Big)\,. \tag{6}$$

In this case, the variance that we expect in each prediction of the latent function is estimated from the data, thus leading to a form of *heteroscedastic* regression.

We can estimate with higher precision a model of the variation in the variance across the input space; to do that we perform *regression of the variance change*, and then inform the outer regression task of that prediction.

Estimator 3 (Nested σ-estimator). *Consider the same estimator of the variance as above, and collect the variance estimates as $\{\langle \boldsymbol{\theta}_{\mathsf{m}}^{(i)}, w^{(i)}\rangle\}_I$. Learn a latent function model of the true variance $\sigma_\star(\cdot)$, which we assume to have a fixed variance $\sigma_\star^2$*

$$w^{(i)} \sim \mathcal{N}\Big(\sigma_\star(\boldsymbol{\theta}_{\mathsf{m}}^{(i)}), \sigma_\star^2\Big) \qquad\qquad y^{(i)} \sim \mathcal{N}\Big(\mathcal{M}_\eta(\boldsymbol{\theta}_{\mathsf{m}}^{(i)}), \sigma_\star(\boldsymbol{\theta}_{\mathsf{m}}^{(i)})^2\Big)\,. \tag{7}$$

This is also a form of heteroschedastic regression, but nesting two GP regressions to account in a finer way for the variance's changes.

3.4 Properties of the Correction Map

The correction map that we learn, for all variance schemes, is a statistically sound estimator of $\mathbb{E}_{\boldsymbol{\theta}_{\mathsf{f}}}[\mathsf{M}_\eta(\boldsymbol{\theta}_{\mathsf{m}})]$, in the sense of being consistent.

Theorem 1 (Correctness). *Let $\mathsf{m}_{\hat{\eta}}(\boldsymbol{\theta}_{\mathsf{m}})$ and $\mathbb{E}_{\boldsymbol{\theta}_{\mathsf{f}}}[\mathsf{M}_{\hat{\eta}}(\boldsymbol{\theta}_{\mathsf{m}})]$ be consistent estimators of $\mathsf{m}_\eta(\boldsymbol{\theta}_{\mathsf{m}})$ and $\mathbb{E}_{\boldsymbol{\theta}_{\mathsf{f}}}[\mathsf{M}_\eta(\boldsymbol{\theta}_{\mathsf{m}})]$, then $\hat{\mathsf{M}}_{\hat{\eta}}(\boldsymbol{\theta}_{\mathsf{m}}) \triangleq \mathsf{m}_{\hat{\eta}}(\boldsymbol{\theta}_{\mathsf{m}}) + \mathcal{M}_{\hat{\eta}}(\boldsymbol{\theta}_{\mathsf{m}})$ is a consistent estimator of $\mathbb{E}_{\boldsymbol{\theta}_{\mathsf{f}}}[\mathsf{M}_\eta(\boldsymbol{\theta}_{\mathsf{m}})]$, for any estimation strategy of $\mathcal{M}_{\hat{\eta}}$.*

The result follows because $\mathcal{M}_{\hat{\eta}}$ converges to $\mathcal{M}_\eta$ due to properties of GPs [12], and because of the consistency of $\mathsf{m}_{\hat{\eta}}$ and $\mathbb{E}_{\boldsymbol{\theta}_{\mathsf{f}}}[\mathsf{M}_{\hat{\eta}}(\boldsymbol{\theta}_{\mathsf{m}})]$. The proof is sketched in Appendix A.2.

The correction map $\mathcal{M}_{\hat{\eta}}(\boldsymbol{\theta}_{\mathsf{m}})$ is estimated from samples of the system, hence it is an approximation of the exact map defined by Eq. (2). Thus, it is a correction map in the sense of Definition 2. However, being the result of a GP regression, $\mathcal{M}_{\hat{\eta}}(\boldsymbol{\theta}_{\mathsf{m}})$ is in fact a random function. Therefore, in measuring the error according to Eq. (1), we need to consider the average value of the random function $\mathbb{E}[\mathcal{M}_{\hat{\eta}}(\boldsymbol{\theta}_{\mathsf{m}})]$. The variance $\mathsf{Var}[\mathcal{M}_{\hat{\eta}}(\boldsymbol{\theta}_{\mathsf{m}})]$, instead, provides a way of computing the error ϵ, but in a statistical sense with confidence α: ϵ can be estimated by averaging over Θ (with respect to $p(\boldsymbol{\theta}_{\mathsf{m}})$) the half-width of the region containing $\alpha\%$ of the probability mass of $\mathcal{M}_{\hat{\eta}}(\boldsymbol{\theta}_{\mathsf{m}})$.

The cost of all our approaches inherently depends on how many samples we pick from Θ, the way parameters in $\boldsymbol{\theta}_{\mathsf{f}}$ are accounted for and the number of parameters in $\boldsymbol{\theta}_{\mathsf{m}}$. The sampling cost in general grows exponentially with $|\boldsymbol{\theta}_{\mathsf{m}}|$, and each Gaussian regression is cubic in the number of sampled values. Notice that, asymptotically, the cost of the empirical and nested σ-estimators is the same, as the two regressions are executed in series.

4 Applications

We discuss now several potential applications of our framework. The advantages of using our approach are mostly computational: the reduced model is simpler to analyze, yet it retains a mechanistic interpretation, compared to statistical emulation.

Model Building. Many common problems in the area of dynamical systems can be tackled by resorting to correction maps.

Problem 1 (Model selection). *Let* M *be a model, and* $\mathsf{m}_1, \ldots, \mathsf{m}_k$ *a set of candidate models for correction, each one having a correction map* $\mathcal{M}_{\eta,1}, \ldots, \mathcal{M}_{\eta,k}$. *The smallest-correction model* m^* *for a statistic* η *is* $\mathsf{m}^* \triangleq \arg\min_{\mathsf{m}_i} \int \mathcal{M}_{\eta,i}(\boldsymbol{\theta})d\boldsymbol{\theta}$.

This criterion is certainly alternative to structural Bayesian approaches [3], which can be used to select the structurally smaller model *within* an equivalence class (see below). Also, allows to frame a model synthesis problem.

Problem 2 (Model synthesis). *For a model* M *with parameters* $\boldsymbol{\theta}_{\mathsf{M}}$ *and for a statistic* η: (*i*) *partition* $\boldsymbol{\theta}_{\mathsf{M}}$ *into sets* $\boldsymbol{\theta}_{\mathsf{m}}$ *and* $\boldsymbol{\theta}_{\mathsf{f}}$, (*ii*) *generate a finite set of plausible reduced models with parameters* $\boldsymbol{\theta}_{\mathsf{m}}$ *and* (*iii*) *select the best one, according to the above model selection criterion.*

In this case, the partition might aim at identifying in $\boldsymbol{\theta}_{\mathsf{M}}$ the model's parameters which contribute the most to the variance for the statistics η. Opportunities for *control* are also plausible if one can reduce the problem of controlling M to "controlling and correcting" a reduced model m. This should be easier as m is structurally smaller than M, and so is lower the complexity of solving a *controller synthesis problem.*

Parameter Estimation. Another application of our framework is in Bayesian parameter estimation of the parameters $\boldsymbol{\theta}_{\mathsf{m}}$ of the big model M, given observations D of variables X, using the small model and the corrected statistics to build approximations of the likelihood function $p(D \mid \boldsymbol{\theta}_{\mathsf{m}})$. For instance, this can be done by correcting the mean of variables X, and using the variance of the correction map as a proxy of the variance X in M after marginalisation of $\boldsymbol{\theta}_{\mathsf{f}}$. We can then plug the distribution of X in a Bayesian inference scheme, and compute an approximate posterior over $\boldsymbol{\theta}_{\mathsf{m}}$.

Model Equivalence. Correction maps can also be used to compare processes, via weak forms of *bisimilarity* with respect to the observations and a statistics.

Definition 3 (Model equivalence). *Models* M_1 *and* M_2*, for a statistic* η *and parameter sets* $\boldsymbol{\theta}_{\mathsf{m}}$ *and* $\boldsymbol{\theta}_{\mathsf{f}}$*, are* η-bisimilar conditioned to m, $\mathsf{M}_1 \equiv^{\eta}_{\mathsf{m}} \mathsf{M}_2$*, if and only if for all* $\boldsymbol{\theta}_{\mathsf{m}} \in \Theta$*, it holds* $\mathcal{M}_{\eta,1}(\boldsymbol{\theta}_{\mathsf{m}}) = \mathcal{M}_{\eta,2}(\boldsymbol{\theta}_{\mathsf{m}})$*. A class of equivalence of models with respect to* m *and* η *is the set of all such bisimilar models.*

This notion of bisimilarity resembles conditional dependence, as we are saying that two models are equivalent if an observer corrects both the same way. In this case, m is a *universal corrector* of $\equiv^{\eta}_{\mathsf{m}}$, as it can correct for all the models in the class. The class of models that are equivalent to a model M is $\{\mathsf{M}^* \mid \mathsf{M}^* \equiv^{\eta}_{\mathsf{M}} \mathsf{M}\}$ – i.e., the class of models with correction zero; notice that in this case $\boldsymbol{\theta}_{\mathsf{f}} = \emptyset$. The previous definition can be relaxed by asking that $|\mathcal{M}_{\eta,1}(\boldsymbol{\theta}_{\mathsf{m}}) - \mathcal{M}_{\eta,2}(\boldsymbol{\theta}_{\mathsf{m}})| \leq \epsilon$, for all $\boldsymbol{\theta}_{\mathsf{m}} \in \Theta$.

Remark 1. Criterion $\equiv^{\eta}_{\mathsf{m}}$ is a weaker form of *probabilistic bisimilarity*, namely if $\mathsf{M}_1 \equiv \mathsf{M}_2$ are bisimilar, then $\mathsf{M}_1 \equiv^{\eta}_{\mathsf{m}} \mathsf{M}_2$ for some m and any statistics of interest. For instance, for CTMCs, this follows because $\equiv$ implies that M_1 and M_2 have the same infinitesimal generators for any parameter $\boldsymbol{\theta}_{\mathsf{m}}$ and $\boldsymbol{\theta}_{\mathsf{f}}$, hence the outcomes of M_1 and M_2 are indistinguishable, and so are their statistics.

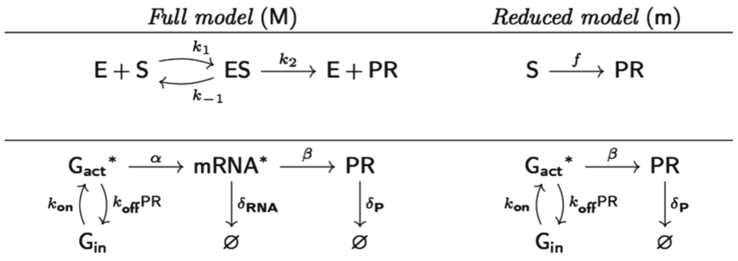
Full model (M)
Reduced model (m)
$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} ES \xrightarrow{k_2} E + PR$
$S \xrightarrow{f} PR$
$G_{act}^* \xrightarrow{\alpha} mRNA^* \xrightarrow{\beta} PR$
k_{on}, $k_{off}PR$, G_{in}, δ_{RNA}, δ_P, $\varnothing$
$G_{act}^* \xrightarrow{\beta} PR$

Fig. 3. Example models tested in this paper. Top panel: the Henri-Michaelis-Menten model, where m is derived when $\mathbf{C} : [\mathsf{E}]_0 + [\mathsf{ES}]_0 \ll [\mathsf{S}]_0 + K_{\mathsf{MM}}$. Bottom panel, a protein translation network where m when $\mathbf{C} : \beta \gg \alpha$.

5 Examples

We investigate two examples to better illustrate our method.

5.1 Model Reduction via QSSA

Consider the irreversible canonical enzyme reaction with its exact representation (here, model M), for enzyme E, complex ES , substrate S and product PR (Fig. 3, top left panel). When the concentration of the intermediate complex does not change on the time-scale of product formation, product is produced at rate $f \triangleq V_{\mathsf{M}}\mathsf{S}/(K_{\mathsf{M}}+\mathsf{S})$ where $V_{\mathsf{M}} = k_2([\mathsf{E}]_0 + [\mathsf{ES}]_0)$ and $K_{\mathsf{MM}} = (k_{-1} + k_2)/k_1$. This is the Henri-Michaelis-Menten kinetics and is technically derived by a quasi-steady-state assumption (QSSA), i.e., $\dot{\mathsf{ES}} = \dot{\mathsf{E}} = 0$, that is in place when $\mathbf{C}$: $[\mathsf{E}]_0 + [\mathsf{ES}]_0 \ll [\mathsf{S}]_0 + K_{\mathsf{MM}}$, where $[x]_0$ is the initial amount of species x. m is thus the QSSA reduced model (Fig. 3, top right panel).

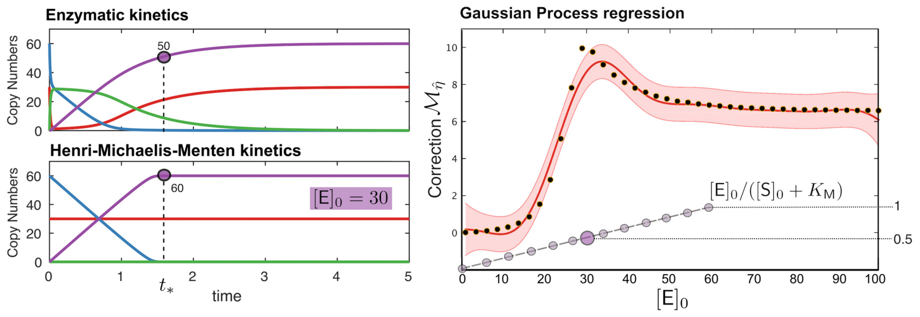

Fig. 4. Correction of the product formation at the transient time $t_* = 1.5$, for a mean field model of irreversible canonical enzyme reaction and its simplified Henri-Michaelis-Menten form. Here $k_1 = 2$, $k_{-1} = 1$ and $k_2 = 1.5$, $[S]_0 = 60$ and $[P]_0 = 0$. Regression over $[\mathsf{E}]_0$ is done with 40 training points from $(0, 100]$, and the correction in Eq. (3) as M's free variables are part of the Michaelis-Menten constant.

We interpret these two models as two systems of *ordinary differential equations*, see Appendix A.1, and learn a correction for the following statistics

$$\eta : \mathbb{E}[\mathsf{PR}(t_*)], \text{ with } \mathsf{PR}(t_*) \text{ the number of products at time } t_* \tag{8}$$

For non-degenerate parameter values both models predict the same equilibrium state, where a total transformation of substrate into product has happened, $\mathbb{E}[\mathsf{PR}(t)] \to [\mathsf{S}]_0$ for large t. Thus, we are not interested in correcting the dynamics of m for long-run times, but rather in the transient (i.e., for small t_*).

Also, as the QSSA hypothesis does not hold for certain initial conditions, we set $\boldsymbol{\theta}_{\mathsf{m}} = \{[\mathsf{E}]_0\}$ as the regression variable, and set $[S]_0 = 60$ and $[P]_0 = 0$. The other parameters are $k_1 = 2$, $k_{-1} = 1$ and $k_2 = 1.5$ with unit *(mole* *per second)*$^{-1}$. In terms of regression, we pick 40 samples of the initial enzyme amount from $(0, 100]$, and set $t_* = 1.5$ as it is a time in the transient (manual tuning). This particular example is rather simple, as the free parameters of M are part of the Michaelis-Menten constant and fixed, so we use the simpler correction of Eq. (3). Also, knowing when the QSSA holds gives us an interval where we expect the correction to shrink towards zero. The map is shown in Fig. 4, which depicts the expected concordance among the correction map and validity of the QSSA.

5.2 Model Reduction via Time-Scale Separation

Consider a gene switching among active and inactive states of mRNA transcription to be ruled by a *telegraph process* with rates $k_{\mathsf{off}}/k_{\mathsf{on}}$. A reaction model of such gene G, protein PR, messenger mRNA with transcription/translation rates α/β as in Fig. 3, bottom left panel.

Here the gene switches among active and inactive states, with rates k_{on} and k_{off}, and PR feedbacks positively on inactivation. Proteins and mRNAs are degraded with rates δ_{P} and δ_{RNA}. In the uppermost part of the diagram species

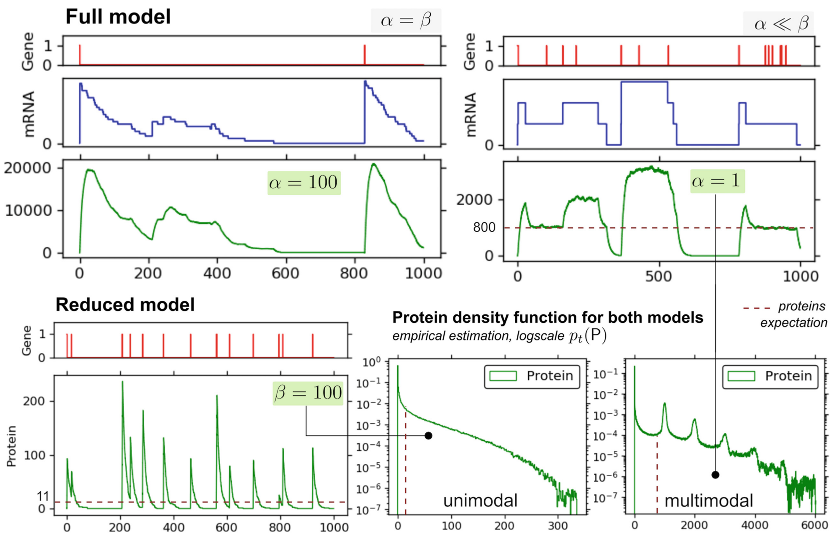

Fig. 5. Comparison between the dynamics of the full and the reduced models from Sect. 5.2, with $k_{\mathsf{off}} = k_{\mathsf{on}} = \delta_{\mathsf{P}} = 10^{-2}$, $\delta_{\mathsf{RNA}} = 10$, $[\mathsf{G}_{\mathsf{act}}]_0 = 1$. Values for transcription (α) and translation (β) are reported in the figure. The reduced model predicts spiked dynamics, leading to a unimodal distribution of proteins. The larger model, instead, can either predict protein buffers, when there is no time scale separation ($\alpha = \beta$), or multiple equilibria, leading to a multimodal distribution of protein counts. Observe that the expectation on the number of proteins in the long run spans over different order of magnitudes, according to the relation between α and β.

marked with a $*$ symbol are not consumed by a reaction, i.e., mRNA transcription is $\mathsf{G_{act}} \rightarrow \mathsf{G_{act}} + \mathsf{mRNA}$. This model can be easily encoded in a Markov process, as discussed in A.1.

We can derive an approximation to M where the transcription step is omitted. This is valid when $\mathbf{C} : \beta \gg \alpha$ (*time-scales separation*), namely for every copy of mRNA a protein is immediately translated.

Correction of protein dynamics. We build a correction map with $\boldsymbol{\theta}_{\mathsf{m}} = \{\beta\}$. In this case the telegraph process is common to both models, but α and δ_{RNA} are free variables of M; here we assume to have a prior delta distribution over the values of mRNA's degradation, so we set $\boldsymbol{\theta}_{\mathsf{f}} = \{\alpha\}$. For some values of the transcription rate condition $\mathbf{C}$ does not hold; in this case it is appropriate to account for α's contribution to the variance in the statistics that we want to correct, when we do a regression over β. Note that also β is part of $\mathbf{C}$.

Model M leads to stochastic bursts in PR's expression when the baseline gene switching is slower than the characteristic times of translation/transcription. Here we set $k_{\mathsf{off}} = k_{\mathsf{on}} = 10^{-2}$, and assume mRNA's lifespan to be longer than protein's ones (also in light of condition $\mathbf{C}$), so $\delta_{\mathsf{RNA}} = 10\delta_{\mathsf{P}} = 10^{-2}$. We simulate both models with one active gene, $[\mathsf{G_{act}}]_0 = 1$; example dynamics are shown in Fig. 5, for $\beta = 100$ and $\alpha \in \{1, 100\}$. We observe that, when $\mathbf{C}$ does not hold ($\alpha = \beta$) the protein bursts increases of one order of magnitude, and the long-run probability density function for the proteins, $p_t(\mathsf{PR})$, becomes multimodal.

We define two statistics. One measures the *first moment* of the random variable that models the number of proteins in the long run; the other is a metric interval temporal logic formula [2], expressing the probability of a protein burst within the first 100 time units of simulation.

$$\eta_1 : \mathbb{E}[\mathsf{PR}(t_*)], \text{ with } \mathsf{PR}(t_*) \text{ the number of proteins at time } t_* \gg 0 \tag{9}$$

$$\eta_2 : \mathbb{E}[p(\varphi)], \text{ with } \varphi \triangleq \mathbf{F}_{[0,100]}\mathsf{PR}(t) > 200 \tag{10}$$

The former is evaluated by a unique long-run simulation of the model, as its dynamics are ergodic. For the latter we estimate the *satisfaction probability of the formula* via multiple ensembles, as in a parametric *statistical model checking problem* [4].

For the regression task we sample 50 values of β, in the range $[0.1, 100]$. For α, instead, we sample 50 random values in the same interval, for each value of β; notice that with high probability we pick values where $\mathbf{C}$ does not hold, so we might expect high correction factors. Data generated and the regression results are shown in Fig. 6, for both fixed-variance regression, empirical σ-estimator in Eq. (4) and with the σ-estimator, Eq. (6). Because variance spans over many orders of magnitude, regression is performed in the logarithmic space. Results highlight a general difference between the posterior variance between the two estimators.

For the second statistics, data is generated from an initial condition where one gene is inactive, $[\mathsf{G_{in}}]_0 = 1$. Notice that the expected time for the gene to trigger its activation is $1/k_{\mathsf{on}} = 100$ (the time upper-bound of the formula),

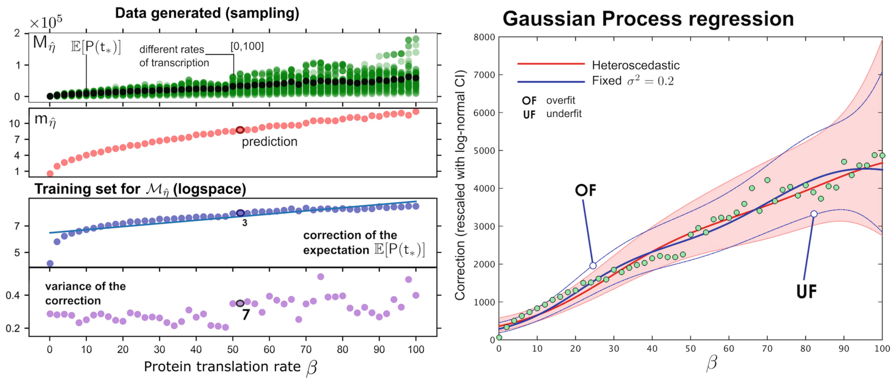

Fig. 6. For the regression task we sample 50 values of β from $[0.1, 100]$. For each value of β, we sample 50 random values of α in the same interval. For each pair (α, β) we estimate the statistics for both models (green, M; red, m), and obtain 50 correction values indexed by each β (green). Correction values (blue and pink) are the expectation and variance of the difference between M's and m's predictions. We transform them logarithmically before doing regression; observe that, on the linear scale, the correction is of the order of 10^3 with variance 10^7 (midpoint value $\beta \approx 50$). Gaussian Process regression (right panel) is performed with a constant $\sigma^2 = 0.2$, Eq. (4) and with the σ-estimator, Eq. (6). Values are re-scaled linearly, and 95 % log-normal confidence intervals are shown; regression highlights that the posterior variances are similar, but the fixed-variance schema tends to underfit or overfit the heteroscedastic variance (*assumed it to be closer to the true one*). (Color figure online)

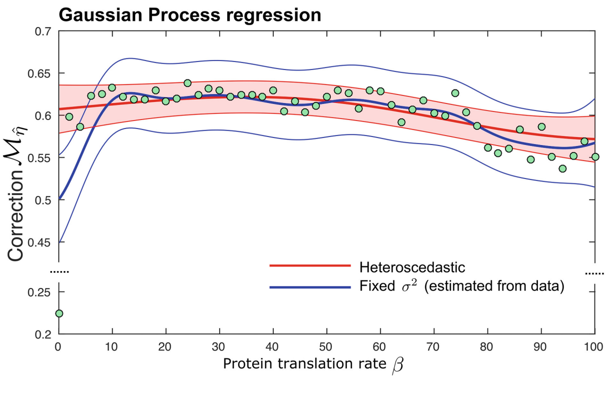

Fig. 7. Correction map for the expected satisfaction probability of the linear logic formula η_2 in Eq. (9). Comparison between the point-wise σ-estimator and the empirical $\overline{\sigma}$-estimator. Sampled data are shown in Appendix Fig. 8.

so for some parametrization there will be non-negligible probability of having no protein spike above threshold 200. The formula satisfaction probability is evaluated with 100 independent model runs, and data generated are shown in Appendix Fig. 8. Regression results are shown in Fig. 7, where the point-wise

σ-estimator and the empirical $\overline{\sigma}$-estimator are compared, highlighting the more robustness of the former with respect to the sampled bottom-left outlier.

6 Conclusions

Abstraction represents a fundamental tool in the armoury of the computational modeller. It is ubiquitously used in biology as an effective mean to reduce complexity, however systematic analysis tools to quantify discrepancies introduced by abstraction are lacking. Prominent examples, beyond the ones already mentioned, include models with delays, generally introduce to approximately lump multiple biochemical steps [6], or physiological models of organ function which adopt multi-scale abstractions to model phenomena at different organisational levels. These include some of the most famous success stories of systems biology, such as the heart model of [11], which also constitutes perhaps the most notable example of a physical systems which has been modelled multiple times at different levels of abstraction. Employing our techniques to clarify the quantitative relationship between models of cardiac electrophysiology would be a natural and very interesting next step.

Our approach has deep roots in the statistical literature. In this paper we have focussed on the scenario where we try to reconcile the predictions of two separate models, however the complex model was simply used as a sample generator black box. As such, nothing would change if instead of a complex model we used a real system which can be interrogated as we vary some control parameters. Our approach would then reduce to fitting the simple model with a structured, parameter dependent error term. This is closely related with the use of Gaussian Processes in the geostatistics literature [7], where simple (generally linear) models are used to explain spatially distributed data, with the residual variance being accounted for by a spatially varying random function. Another important connection with the classical statistics literature is with the notion of *emulation* [9]. Emulation constructs a statistical model of the output of a complex computational model by interpolating sparse model results with a Gaussian Process. Our approach can be viewed as a *partial emulation*, whereby we are interested in retaining mechanistic detail for some aspects of the process, and emulate statistically the residual variability. In this light, our work represents a novel approach to bridge the divide between the model-reduction techniques of formal computational modelling and the statistical approximations to complex models.

A Appendix

All the code that replicate these analysis is available at the corresponding author's webpage, and hosted on Github (repository GP-correction-maps).

A.1 Further Details on the Examples

The two models from Sect. 5.1 correspond to these systems of differential equations

$$\mathsf{M}{:=}\begin{cases}\dot{\mathsf{E}} = -k_1\mathsf{E}\cdot\mathsf{S} + k_{-1}\mathsf{ES} + k_2\mathsf{ES}\\ \dot{\mathsf{S}} = -k_1\mathsf{E}\cdot\mathsf{S} + k_{-1}\mathsf{ES}\\ \dot{\mathsf{ES}} = k_1\mathsf{E}\cdot\mathsf{S} - k_{-1}\mathsf{ES} - k_2\mathsf{ES}\\ \dot{\mathsf{PR}} = +k_2\mathsf{ES}\end{cases} \qquad \mathsf{m}{:=}\begin{cases}\dot{\mathsf{E}} = 0\\ \dot{\mathsf{S}} = -V_M\mathsf{S}/(K_M+\mathsf{S})\\ \dot{\mathsf{ES}} = 0\\ \dot{\mathsf{PR}} = +V_M\mathsf{S}/(K_M+\mathsf{S})\end{cases}$$

which we solved in MATLAB with the ode45 routine with all parameters (InitialStep, MaxStep, RelTol and AbsTol) set to 0.01.

Concerning the Protein Translation Network (PTN) in Sect. 5.2, the set of reactions and their propensity functions that we can use to derive a Continuous Time Markov Chain model of the network are the following. Here $\boldsymbol{x}$ denotes a generic state of the system and, for instance, $\boldsymbol{x}_{\mathsf{mRNA}}$ the number of mRNA copies in $\boldsymbol{x}$.

event	*reaction*	*propensity*
activation	$\mathsf{G_{in}} \rightarrow \mathsf{G_{act}}$	$a_1(\boldsymbol{x}) = k_{\mathsf{on}} \cdot \boldsymbol{x}_{\mathsf{G_{in}}}$
inactivation	$\mathsf{G_{act}} \rightarrow \mathsf{G_{in}}$	$a_2(\boldsymbol{x}) = k_{\mathsf{off}} \cdot \boldsymbol{x}_{\mathsf{PR}}$
transcription	$\mathsf{G_{act}} \rightarrow \mathsf{G_{act}} + \mathsf{mRNA}$	$a_3(\boldsymbol{x}) = \alpha \cdot \boldsymbol{x}_{\mathsf{G_{act}}}$
mRNA decay	$\mathsf{mRNA} \rightarrow \varnothing$	$a_4(\boldsymbol{x}) = \delta_{\mathsf{RNA}} \cdot \boldsymbol{x}_{\mathsf{mRNA}}$
translation	$\mathsf{mRNA} \rightarrow \mathsf{mRNA} + \mathsf{PR}$	$a_5(\boldsymbol{x}) = \beta \cdot \boldsymbol{x}_{\mathsf{mRNA}}$
PR decay	$\mathsf{PR} \rightarrow \varnothing$	$a_4(\boldsymbol{x}) = \delta_{\mathsf{P}} \cdot \boldsymbol{x}_{\mathsf{PR}}$

The reduced PTN model is a special of this reactions set where transcription and mRNA decay are omitted. In this case we used StochPy to simulate the models and generate the input data per regression – see http://stochpy.sourceforge.net/; data sampling exploits python parallelism to reduce execution times.

For regression, we used the Gaussian Processes for Machine Learning toolbox for fixed-variance regression, see http://www.gaussianprocess.org/gpml/code/matlab/doc/ and a custom implementation of the other forms of regression.

A.2 Proofs

Proof of Theorem 1

Proof. Both the empiricals and nested estimator rely on an *unbiased* estimator of the mean/variance, which means that if $k \to \infty$, i.e., we sample all possible values for the free variables, we would have a true model of $\overline{y}$ σ. This means that, for each sampled value from Θ, even the simplest $\overline{\sigma}$-estimator would be equivalent, in expectation, to the marginalization of the free variables. This is enough, combined with properties of Gaussian Processes regression (i.e., the convergence to the true model with infinite training points), to state that the overall approach leads to an unbiased estimator of the correction map.

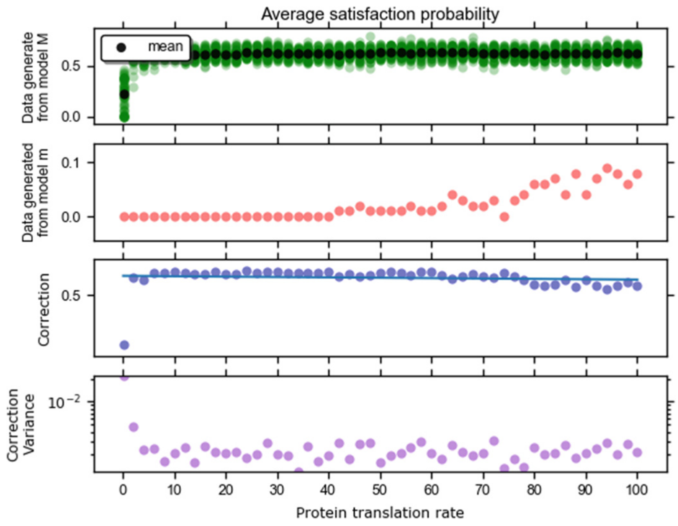

Fig. 8. Data generated to compute the satisfaction probability of the linear logic formula η_2 in Eq. (9). For each model 100 independent simulations are used to estimate the expectation of the probability. The regression input space is the same used to compute η_1, but the models are simulated with just one inactive gene in the initial state. The heteroscedastic variance in the regression is computed as the variance of the correction of the expected satisfaction probability (point-wise σ-estimator); the fixed-variance regression is computed by estimating the variance from the data (empirical $\overline{\sigma}$-estimator).

References

1. Aitken, S., Alexander, R.D., Beggs, J.D.: A rule-based kinetic model of rna polymerase ii c-terminal domain phosphorylation. J Roy. Soc. Interface **10**(86), 20130438 (2013)
2. Alur, R., Feder, T., Henzinger, T.A.: The benefits of relaxing punctuality. J. ACM **43**(1), 116–146 (1996)
3. Barber, D.: Bayesian Reasoning and Machine Learning. Cambridge University Press, Cambridge (2012)
4. Bortolussi, L., Milios, D., Sanguinetti, G.: Smoothed model checking for uncertain continuous-time markov chains. Inf. Comput. **247**, 235–253 (2016)
5. Bortolussi, L., Sanguinetti, G.: Learning and designing stochastic processes from logical constraints. In: Joshi, K., Siegle, M., Stoelinga, M., D'Argenio, P.R. (eds.) QEST 2013. LNCS, vol. 8054, pp. 89–105. Springer, Heidelberg (2013)
6. Caravagna, G.: Formal modeling and simulation of biological systems with delays. Ph.D. thesis, University of Pisa (2011)
7. Cressie, N., Wikle, C.K.: Statistics for Spatio-Temporal Data. Wiley, New York (2015)
8. Hoyle, D.C., Rattray, M., Jupp, R., Brass, A.: Making sense of microarray data distributions. Bioinformatics **18**(4), 576–584 (2002)
9. Kennedy, M.C., O'Hagan, A.: Bayesian calibration of computer models. J. Roy. Stat. Soc.: Ser. B (Stat. Methodol.) **63**(3), 425–464 (2001)

10. Lawrence, N.D., Sanguinetti, G., Rattray, M.: Modelling transcriptional regulation using gaussian processes. In: Advances in Neural Information Processing Systems, pp. 785–792 (2006)
11. Noble, D.: Modeling the heart-from genes to cells to the whole organ. Science **295**(5560), 1678–1682 (2002)
12. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)