Journals
The Institution of Engineering and Technology

Identifying latent dynamic components in biological systems

ISSN 1751-8849
Received on 31st July 2014
Accepted on 19th November 2014
doi: 10.1049/iet-syb.2014.0013
www.ietdl.org

Ivan Kondofersky[1,2], *Christiane Fuchs*[1,2], *Fabian J. Theis*[1,2] ✉

[1] *Institute of Computational Biology, German Research Center for Environmental Health, Helmholtz Zentrum München, Ingolstädter Landstr. 1, 85764 Neuherberg, Germany*
[2] *Center for Mathematics, Chair of Mathematical Modeling of Biological Systems, Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany*
✉ *E-mail: fabian.theis@helmholtz-muenchen.de*

Abstract: In computational systems biology, the general aim is to derive regulatory models from multivariate readouts, thereby generating predictions for novel experiments. In the past, many such models have been formulated for different biological applications. The authors consider the scenario where a given model fails to predict a set of observations with acceptable accuracy and ask the question whether this is because of the model lacking important external regulations. Real-world examples for such entities range from microRNAs to metabolic fluxes. To improve the prediction, they propose an algorithm to systematically extend the network by an additional latent dynamic variable which has an exogenous effect on the considered network. This variable's time course and influence on the other species is estimated in a two-step procedure involving spline approximation, maximum-likelihood estimation and model selection. Simulation studies show that such a hidden influence can successfully be inferred. The method is also applied to a signalling pathway model where they analyse real data and obtain promising results. Furthermore, the technique can be employed to detect incomplete network structures.

1 Introduction

A central objective in computational systems biology is to identify components of biological system networks and their relation to one another [1, 2]. For the prediction of time-resolved, dynamical network behaviour, mathematical models are employed that typically involve several unknown parameters in addition to the network components. A popular modelling approach for time-resolved measurements is given by ordinary differential equations (ODEs) that represent the dynamics of and dependencies between the components of the network. The parameters describing the dynamics in an ODE must be inferred statistically, and in the case of several competing network models, the most appropriate model can be chosen by model selection methods. Hence, one deals with a mathematical modelling problem and a statistical estimation problem, simultaneously [3].

In such an analysis, ODEs directly arise from the network topology, that is, the modeller specifies the components of the network and possible interactions. In many applications, the key elements of the dynamics of interest have been previously determined in various studies and are well-known from the literature. It is possible, however, that some interaction partners or connections remain unspecified. For example, in addition to transcription factors modulating gene regulation, strong evidence indicates that microRNAs play an important role in transcription and translation processes [4]. Translation can also be influenced by external stimuli like drugs [5–7]. Consequently, a mathematical model may be insufficient to explain the dynamics of interest, that is, discrepancies with the measured data which are not simply because of the measurement error may be evident even with the best model fit.

A promising way of addressing such discrepancies is given by employing additional network components to extend the proposed model. Our main focus in this paper is a systematic model extension. A substantial amount of work has been conducted in the past years in this field.

In [8], additional links in undirected graphs are identified using Gaussian graphical models. These links represent model extensions and are systematically identified using an l_1-penalised likelihood. However, the proposed algorithm is not applicable to dynamical data.

Gao *et al.* [9] and Honkela *et al.* [10] also consider a model extension, this time for dynamical data. Similar to the approach to be presented here, they describe their models in terms of ODEs with a latent variable. Using Gaussian processes, they infer the time course of this variable and predict its behaviour. However, they do not model entire networks, which may possibly involve numerous links between components, but rather focus only on transcription and translation of single genes and on analytical solutions of the specific ODE models.

Furthermore, model extension by latent variables is utilised in the context of latent confounder modelling [11, 12]. Here, the most frequently used method is structural equation modelling (SEM) [13, 14]. SEM allows the identification of multiple latent variables and their relationship with observed variables by exploiting the data covariance structure. SEM is mainly formulated for single time points, and an extension to dynamical data is quite limited and often not possible.

In the present study, we address the problem of poor model quality in dynamical models by considering the effect of hidden influences on the network. We do not assume a functional form for the putative time courses of such hidden processes, but flexibly estimate their dynamics and interaction strengths. Wherever a hidden influence is observed that substantially improves the model's ability to represent the data, we attempt to provide a biological meaning with the help of experimental collaborators. Thus, we can guide the design of additional experiments in a detailed manner by providing a quantification of the hidden time courses as well as relative interaction rates between the hidden components and the existing network.

This paper is organised as follows. Section 2 presents the ODE models considered and the means by which a hidden influence is included therein as well as a schematic representation of the developed method. The hidden component and the model parameters are statistically estimated in a two-step procedure, as described in Section 3. Furthermore, this section also discusses parameter uncertainty and explains how the most appropriate model among several competing ones can be chosen with model selection methods. The proposed method is applicable to Lipschitz continuous ODE models, for example, gene regulation models or signal transduction models. Section 4 applies the developed technique to different scenarios and to a real data example – the JAK2-STAT5 signalling pathway. Section 5 concludes this paper and discusses the strengths and limitations of the proposed method.

2 Approach

In this section, we highlight the main ideas of the present study. Systematic network extension is illustrated by considering a small motif example. Next, we generalise this extension to networks of size *N*.

Consider a simple motif like the one presented in Fig. 1.

We use the schematic representation of small network motifs shown in Fig. 1 to illustrate our method as follows. Fig. 1*a* shows a simple network motif comprising two components x_1 and x_2, which influence each other, as indicated by the corresponding arrows. With the proposed method, we estimate a hidden component *h*, shown in Fig. 1*b*, which may substantially contribute to the network dynamics, but was not previously considered. Thus, we call *h* a 'hidden influence.' Fig. 1*c* stresses that not all components must be observed.

ODEs are the most prevalent choice for describing such network dynamics [15]. We assume these to be Lipschitz continuous; thus, the existence and uniqueness of an ODE solution are guaranteed. For motif A in Fig. 1, the corresponding equation is

$$\dot{x}_i(t) = \psi_i(\boldsymbol{k}, \boldsymbol{x}(t)) \tag{1}$$

with parameter vectors $\boldsymbol{k} = (k_1, \ldots, k_L)^{\mathrm{T}}$, $k_l \in R_{\geq 0}$, non-negative state vector $\boldsymbol{x}(t) = (x_1(t), \ldots, x_N(t))^{\mathrm{T}}$, $x_i(t) \in R_{\geq 0}$, derivatives with respect to time $\dot{x}_i(t)$, possibly non-linear functions $\psi_i{:}R^p_{\geq 0} \times R^N_{\geq 0} \to R$ and suitable initial values $x_i(t_0)$, where $t \geq t_0$ represents the time. The functions ψ_i generate the network structure and may include several combinations of the state variables $\boldsymbol{x}(t)$ such as linear combinations, Michaelis–Menten kinetics, complex formation and others. The connection between the state variables is described by the parameters $\boldsymbol{k}$.

The components $x_i(t)$ may be observed or unobserved. In addition to the motif in Fig. 1*a*, we now assume a time-varying hidden component $h(t)$ that acts linearly on $\dot{x}_i(t)$, as shown in Fig. 1*b*. The system of differential equations then changes to

$$\dot{x}_i(t) = \psi_i(\boldsymbol{k}, \boldsymbol{x}(t)) + a_i h(t) \tag{2}$$

with weights $\boldsymbol{a} = (a_1, \ldots, a_N)^{\mathrm{T}}$, $a_i \in R$. Positive weights a_i in this context represent activation of the *i*-th component, whereas a negative value of a_i implies inhibition. A similar model was considered, e.g. in [16]. Other than for $x_i(t)$, we do not assume any parametric structure for the hidden component. The time course of $h(t)$ cannot be observed directly.

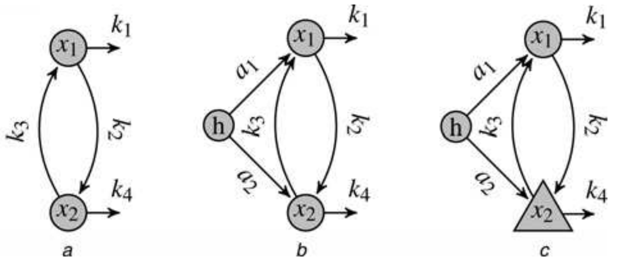
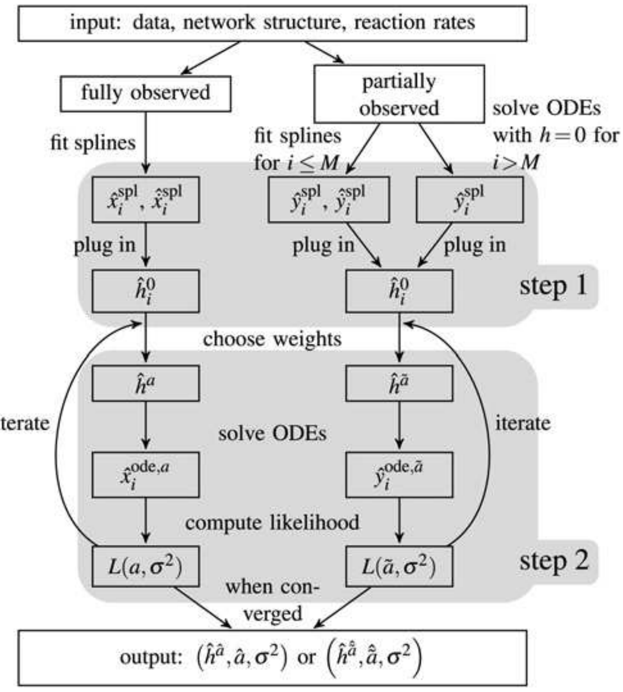


Fig. 1 *Example network motifs (circles: observed and hidden components; triangle: unobserved or indirectly observed component)*

a Motif without hidden components, that is, all components are observed
b Motif with a single hidden component (*h*), where all other components are observed
c Motif with a single hidden component (*h*) and partially observed components
Partially observed networks are discussed in Section 3.5



input: data, network structure, reaction rates
fully observed
partially observed
fit splines
fit splines for $i \leq M$
solve ODEs with $h = 0$ for $i > M$
$\hat{x}_i^{\mathrm{spl}}, \hat{\dot{x}}_i^{\mathrm{spl}}$
$\hat{y}_i^{\mathrm{spl}}, \hat{\dot{y}}_i^{\mathrm{spl}}$
$\hat{y}_i^{\mathrm{spl}}$
plug in
plug in
plug in
$\hat{h}_i^0$
$\hat{h}_i^0$
step 1
choose weights
$\hat{h}^a$
$\hat{h}^{\tilde{a}}$
iterate
solve ODEs
iterate
$\hat{x}_i^{\mathrm{ode},a}$
$\hat{y}_i^{\mathrm{ode},\tilde{a}}$
compute likelihood
$L(a, \sigma^2)$
$L(\tilde{a}, \sigma^2)$
step 2
when converged
output: $(\hat{h}^{\hat{a}}, \hat{a}, \sigma^2)$ or $(\hat{h}^{\hat{\tilde{a}}}, \hat{\tilde{a}}, \sigma^2)$

Fig. 2 *Flowchart illustrating the details of the developed method*

Method involves estimating hidden components according to two major steps (indicated by grey boxes). We distinguish between fully and partially observed networks. If at least one network component is unobserved, the procedure requires a preliminary step where the system of ODEs is solved without considering a hidden component. As a next step in both scenarios, penalisation splines are fitted to the measurements using cross-validation for the estimation of the smoothing parameter. Finally, a maximum-likelihood loop is performed until convergence to estimate the time course of the hidden component and its interaction weights as well as the noise parameter σ^2

Six elements determine the model: the components x_i and their time derivatives $\dot{x}_i$, the parameter vectors $\boldsymbol{k}$ and $\boldsymbol{a}$, the dependency describing functions ψ_i and the hidden influence *h*. We will extend established models from the literature by adding hidden components and applying our estimation method described in Section 3. For reasons of simplicity, we assume that the reaction rates $\boldsymbol{k}$ and dependency functions ψ_i are known. Both assumptions can also be relaxed, as is demonstrated in Section 4.2 where we additionally estimate $\boldsymbol{k}$ and recover a missing feedback, thus altering the network structure.

The objective of our study is to estimate a_i and $h(t)$, and examine if they improve the ability of the model to represent the data; this also requires the estimation of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$. In our analysis, we exploit the following connection between *h* and all other components

$$h(t) = \frac{\dot{x}_i(t) - \psi_i(\boldsymbol{k}, \boldsymbol{x}(t))}{a_i} \tag{3}$$

for all *t* and *i* with $a_i \neq 0$. The hidden influence can then be estimated according to two major steps as follows. First, we fit penalisation splines to the measurements of $\boldsymbol{x}$. This allows a direct computation of the time derivatives $\dot{\boldsymbol{x}}$ such that the right-hand side of (3) is known up to a scaling factor a_i. These factors are then estimated via likelihood maximisation, utilising the differential equation structure. A flowchart that illustrates the details of the developed method is shown in Fig. 2.

3 Methods

This section describes the above-mentioned two-step procedure for the estimation of the hidden influence *h*. As a basis, we assume observations $x_i^{\mathrm{obs}}(t_j)$ of all components $x_1, \ldots, x_N$ at discrete time points $t_0, \ldots, t_n$. The first step is presented in Section 3.1, where we use spline functions to approximate the time courses of $x_1, \ldots, x_N$ and their derivatives. In a second step, we define a noise model for the data and estimate the weights a_i using likelihood maximisation in Section 3.2. In Section 3.3, we discuss uncertainty and the fit quality of the parameters of interest. In Section 3.4, we perform model selection on the considered networks. Finally, in Section 3.5, we extend the estimation methods to the case of partially observed networks.

3.1 Spline estimation for observed time courses and their derivatives

Splines are a convenient way to approximate the time course of a series of measurements in a functional form and are successfully used to model time-resolved, biological data, for example, [17]. They typically arise as a linear combination $\sum_k \beta_k \phi_k(t)$ of known basis functions $\phi_1, \ldots, \phi_K$ with corresponding coefficients $\beta_1, \ldots, \beta_K \in R$, as described, for example, in [18]. These coefficients are chosen such that the differences between the measured data and the corresponding function values are minimised. The wiggliness of the spline depends on, in addition to other settings, the dimension K of the basis spanning the functional space. A large K can result in overfitting in the sense of exaggerated data faithfulness. In this case, the spline will perfectly resemble the presently observed data but poorly predict future measurements. A low value for K, on the other hand, can lead to an oversimplified approximation of the true dynamics. To achieve a good trade-off between these two extremes, we use penalisation splines where the wiggliness of the spline is controlled by an additional smoothing parameter $\lambda > 0$ [19]. Applied to the model given in (2), for a given λ_i, the basis coefficients $\beta_{1i}, \ldots, \beta_{Ki}$ are chosen such that the following term is minimised [20]

$$\left(\boldsymbol{x}_i^{\mathrm{obs}} - \sum_{k=1}^{K} \beta_{ki}\phi_k\right)^{\mathrm{T}} \left(\boldsymbol{x}_i^{\mathrm{obs}} - \sum_{k=1}^{K} \beta_{ki}\phi_k\right) + \lambda_i \int_{t_0}^{t_n} \left(\sum_{k=1}^{K} \beta_{ki}\ddot{\phi}_k(s)\right)^2 \mathrm{d}s \quad (4)$$

In this notation, $\boldsymbol{x}_i^{\mathrm{obs}} = (x_i^{\mathrm{obs}}(t_0), \ldots, x_i^{\mathrm{obs}}(t_n))^{\mathrm{T}}$ is the vector of the measured data, and the basis functions evaluated at the observation times are denoted by $\phi_k = (\phi_k(t_0), \ldots, \phi_k(t_n))^{\mathrm{T}}$. As the index i in λ_i suggests, a penalisation parameter is chosen for each component separately.

In this study, we choose a sufficiently large K and estimate λ_i using leave-one-out cross-validation. Hereby, splines are estimated with n observations and one observation is left out. The excluded observation is subsequently compared to the spline approximation at the same time point. This comparison is repeated for all $n+1$ observations and different values of λ_i. Finally, this procedure leads to an optimal λ_i for which the spline approximation is close to the data without overfitting it. Alternatively, one could use bootstrap techniques or more generalised cross-validation criteria for this purpose, as discussed in [20].

Minimisation of (4) yields optimal coefficients $\hat{\beta}_{ki}$, and consequently an approximation for the time course of the observed components

$$\hat{x}_i^{\mathrm{spl}}(t) := \sum_{k=1}^{K} \hat{\beta}_{ki}\phi_k(t) \quad (5)$$

and their time derivatives

$$\hat{\dot{x}}_i^{\mathrm{spl}}(t) := \frac{\partial \hat{x}_i^{\mathrm{spl}}(t)}{\partial t} = \sum_{k=1}^{K} \hat{\beta}_{ki}\dot{\phi}_k(t) \quad (6)$$

The estimation of the time derivatives given by (6) plays an equally important role as the estimation of the splines given by (5) for the final estimation of the time course of the hidden component given by (3). Thus, particularly for the analysis of very noisy data, additional smoothing techniques, such as a higher penalisation order in (4), may be considered. Furthermore, the shape of the estimated spline is not only determined by the parameters K and λ_i but also by the type of basis functions used. Common types of spline functions are B-splines, Fourier splines (applicable to, e.g. for periodic data), polynomial splines, among others [20]. We implemented our method to include all these possibilities. The results presented in Section 4 are based on cubic B-splines defined on an equally spaced time grid. These functions are twice continuously differentiable such that the penalisation term in (4) is well-defined [20]. In practice, however, the integral is approximated through finite differences [20]. For that reason, we only require the basis functions to be at least once differentiable so that (6) is well-defined.

With the approximations given by (5) and (6), we can now estimate the numerator in (3)

$$\hat{h}_i^0(t_j) = \hat{\dot{x}}_i^{\mathrm{spl}}(t_j) - \psi_i(\boldsymbol{k}, \hat{\boldsymbol{x}}^{\mathrm{spl}}(t_j)) \quad (7)$$

For the estimation of the denominator in (3), we apply a likelihood approach, as described in the following section.

3.2 Maximum-likelihood estimation

Given a weight vector $\boldsymbol{a}$, the hidden influence can be approximated as

$$\hat{h}^{\boldsymbol{a}}(t_j) = \frac{1}{N^{\boldsymbol{a}}} \sum_{\{i: a_i \neq 0\}} \frac{\hat{h}_i^0(t_j)}{a_i} \quad (8)$$

where $N^{\boldsymbol{a}}$ is the number of non-zero weights a_i. The case $\boldsymbol{a} = \boldsymbol{0}$ can be excluded without loss of generality because it indicates the absence of a hidden influence extending the network. This approximation of h will later be plugged into (2) where it is multiplied with a_i. If $\hat{h}_i^0$ is the true numerator of (3), the time courses $\hat{h}_i^0$ on the right-hand side will all be identical up to a scaling factor. However, because it is an approximation there will be differences between them in practice. For this reason we consider the pointwise weighted average in (8), which presents a natural choice of a summary statistic. If it holds that the single estimates $\hat{h}_i^0(t_j)$ strongly differ from each other, then the weighted average $\hat{h}^{\boldsymbol{a}}(t_j)$ will be inaccurate, and this in turn will be reflected in the likelihood function and the corresponding information criterion that we later formulate in (13) and (16), respectively, thus leading to the rejection of the proposed model.

Plugging in h into (2) and multiplying it by a_i introduces a non-identifiability. Because of

$$a_i \hat{h}^{\boldsymbol{a}} = (\xi a_i)\left(\frac{\hat{h}^{\boldsymbol{a}}}{\xi}\right) \quad (9)$$

for any $\xi \neq 0$, the weights a_i are non-identifiable. For this reason, we restrict $\boldsymbol{a}$ to $\sum_i |a_i| = 1$. In the special case of a network consisting of only one component x_1, we only estimate the interaction direction of the hidden influence, that is, $a \in \{-1, 1\}$.

In most biological applications, the data contains noise of different origins, such as measurement noise or technical noise [21, 22]. The most common assumption is that measurement errors are independent and normally distributed with mean zero and constant standard deviation $\sigma > 0$

$$x_i^{\mathrm{obs}}(t_j) = x_i(t_j) + \varepsilon_{ij}, \quad \varepsilon_{ij} \overset{\mathrm{iid}}{\sim} \mathcal{N}(0, \sigma^2) \quad (10)$$

In applications, $x_i(t_j)$ often has a positive domain, and in this case, (10) might be ill-defined. Note that we do not restrict our methods to only this type of noise. In Appendix 1, we also specifically derive all the equations given in this section for log-normally distributed multiplicative noise. The distribution of ε_{ij} immediately propagates to the measurements

$$x_i^{\mathrm{obs}}(t_j)|x_i(t_j) \overset{\mathrm{iid}}{\sim} \mathcal{N}\left(x_i(t_j),\, \sigma^2\right) \tag{11}$$

While the true time course $x_i(t)$ is unknown, it has already been approximated in (5). This approximation, however, does not contain any information about $\boldsymbol{a}$, which we seek to estimate in the following. Hence, we introduce another approximation for $x_i(t)$, this time exploiting the ODE structure given in (2): for a given $\boldsymbol{a}$, we plug in $\hat{h}^{\boldsymbol{a}}$ from (8) into the ODE given in (2) and solve the differential equations either analytically or numerically, as described, for example, in [23]. This yields $\hat{x}_i^{\mathrm{ode},\boldsymbol{a}}(t_j)$ and leads to the approximate distribution

$$x_i^{\mathrm{obs}}(t_j)|\hat{x}_i^{\mathrm{ode},\boldsymbol{a}}(t_j) \overset{\mathrm{iid}}{\sim} \mathcal{N}(\hat{x}_i^{\mathrm{ode},\boldsymbol{a}}(t_j),\, \sigma^2) \tag{12}$$

Overall, we arrive at the conditional likelihood function

$$L(\boldsymbol{a}, \sigma^2) = \prod_{i=1}^{N}\prod_{j=0}^{n} f_{\boldsymbol{a},\sigma^2}(x_i^{\mathrm{obs}}(t_j)) \tag{13}$$

where $f_{\boldsymbol{a},\sigma^2}$ is the probability density function corresponding to the chosen error specification.

In addition, a conditional estimate for σ^2 can be derived analytically

$$\hat{\sigma}^2_{\mathrm{ML}} = \frac{1}{N(n+1)}\sum_{i=1}^{N}\sum_{j=0}^{n}\left(x_i^{\mathrm{obs}}(t_j) - \hat{x}_i^{\mathrm{ode},\boldsymbol{a}}(t_j)\right)^2 \tag{14}$$

The parameters $\boldsymbol{a}$ and σ^2 are jointly estimated using (14) and a numerical optimisation of (13). In addition, unknown initial conditions $\boldsymbol{x}(t_0)$ are treated as unknown parameters and are equivalently estimated.

3.3 Parameter uncertainty

We further explore our likelihood approach with respect to parameter uncertainty. The overall estimation performance of the unknown parameters σ^2 and $\boldsymbol{a}$ can be analysed by computing the Cramer-Rao lower bound (CRLB) [24, 25] which is defined as the inverse expected Fisher information matrix. This theoretical value describes a lower bound for the mean squared error (MSE) of a given parameter. To that end, we look at the diagonal elements of the expected Fisher information matrix, which, in the case of a normally distributed error, have the following form

$$I_k(\boldsymbol{a}, \sigma^2) = \begin{cases} \dfrac{1}{\sigma^2}\displaystyle\sum_i\sum_j\left(\frac{\partial}{\partial a_k}\hat{x}_i^{\mathrm{ode},\boldsymbol{a}}(t_j)\right)^2 & k \le N \\ N(n+1)/\left(2\sigma^4\right) & k = N+1 \end{cases} \tag{15}$$

In practice, we solve the ODEs numerically. Here, a sophisticated ODE solver, such as the Runge-Kutta fourth order method [26] can be employed to produce accurate estimates. However, an analytical derivation of the CRLB for such a method becomes very complex because of the complicated recursive formulation of the ODE solution. For exemplary purposes, we outline a derivation for a specific small example using the Euler method in Appendix 2.

Large values on the diagonal of the expected Fisher information matrix represent parameters with a small CRLB. These parameters can be estimated accurately with an (asymptotically) efficient estimator. For the parameters a_l, the respective lth diagonal element increases if

- σ^2 is small, that is, the data are subject to a small amount of noise,
- $\left((\partial/\partial a_k)\hat{x}_i^{\mathrm{ode},\boldsymbol{a}}(t_j)\right)^2$ is large, that is, the ODE solution is sensitive to changes in the parameter a_k and
- n and/or N are large, that is, the data arise from a large number of time points and different (observed) species.

For the parameter σ^2, we look at the $(N+1)$th diagonal element of (15), which increases if

- σ^2 is small, that is, the data are subject to a small amount of noise and
- n and/or N are large, that is, the data arise from a large number of time points and different (observed) species.

We can conclude that, as expected, the estimation accuracy will suffer if we apply our method to small networks, few observations, conditions indicative of a weak influence of the hidden component and large noise. As indicated in Section 3.2, we estimate the parameters with a maximum-likelihood approach. The estimation is asymptotically efficient [27]; thus, the CRLB is asymptotically achieved. However, the approximation of the time courses using splines as described in Section 3.1, introduces additional uncertainty. In Appendix 2, we examine this loss of accuracy for a given showcase network and various parameter combinations, thereby concluding that our method produces estimates that are very close to the CRLB.

3.4 Model selection

The vector $\boldsymbol{a}$ controls the interaction strength between the hidden influence h and the network components x_i. If a weight a_i is estimated to be close to zero, it will have a negligible effect on the network and will probably improve the model fit only slightly. In such a case, one may ask whether the inclusion of this parameter a_i is worth the involved estimation effort or whether one should simply set this component equal to zero, thus reducing the complexity of the model.

To quantify the trade-off between improved model fitting and increased model complexity, we consider the Akaike information criterion (AIC) and the Bayesian information criterion (BIC), which are established model choice rules [28, 29]

$$\begin{aligned} \mathrm{AIC}(\hat{\boldsymbol{\theta}}) &= -2\log(L(\hat{\boldsymbol{\theta}})) + 2\dim(\hat{\boldsymbol{\theta}}) \\ \mathrm{BIC}(\hat{\boldsymbol{\theta}}) &= -2\log(L(\hat{\boldsymbol{\theta}})) + \log\left((n+1)N\right)\dim(\hat{\boldsymbol{\theta}}) \end{aligned} \tag{16}$$

In these equations, $\hat{\boldsymbol{\theta}}$ denotes a vector containing all parameter estimates, $L(\hat{\boldsymbol{\theta}})$ is the likelihood function (13) evaluated at $\hat{\boldsymbol{\theta}}$ and $\dim(\hat{\boldsymbol{\theta}})$ is the number of estimated parameters. The AIC and BIC weigh the accuracy of the fit (measured by the first summand) against the complexity of the model (measured by the second summand). One then chooses the model with the smallest AIC or BIC.

To consider the complexity of the overall estimation procedure, the vector $\boldsymbol{\theta}$ can be chosen to include all unknowns determined in our two-step approach, that is, all λ_i, β_{ik} and σ^2. In our considerations, however, the number of variables is constant apart from the number of non-zeros a_i. Hence, we can replace $\dim(\hat{\boldsymbol{\theta}})$ by $N^{\boldsymbol{a}}$ as defined in (8), to compare different models.

The models that we are considering with our method are all of a nested type. The special case of $\boldsymbol{a} = \boldsymbol{0}$ is the null model and is nested within all other models with arbitrary $\boldsymbol{a}$. Regarding the decision of which values a_i to set equal to zero, we follow three conventional variable selection methods: best subset selection, forward stepwise selection and backward stepwise selection. These and additional model selection methods are discussed in [30]. In the best subset selection, the AIC or the BIC is computed for all possible models, and the model with the best score is chosen. This approach is computationally expensive but guarantees to find the best model. The latter two methods are greedy algorithms that compute the AIC/BIC only for a fraction of possible models, which exploit the nested structure, thus saving computational time while yielding satisfactory results in practice.

In the forward stepwise selection, we begin with the model given in (2), which contains no interactions between the hidden influence and the network components, that is, all a_i equal zero. In the second step, N models are estimated, where, for each of the models a different element of $\boldsymbol{a}$ is non-zero while the others are held equal to zero. If the best model outperforms the selected model from the previous step, this model is accepted, and in the subsequent step, another component of $\boldsymbol{a}$ is set to a non-zero value. This step is repeated until no increase in model performance is achieved with a more complicated model. Once a component a_i is chosen to be non-zero, it will remain non-zero in all subsequent steps.

Backward stepwise selection is an analogy of the forward stepwise selection wherein the initial model selected is the most complicated model for which all interactions between the hidden and the other components are estimated. In each subsequent step, a single entry of $\boldsymbol{a}$ is fixed to zero until no lower value of AIC/BIC is achieved.

In Section 4, we employ the BIC for model choice on synthetic and real data because this criterion penalises the model complexity more than the AIC.

3.5 Partially observed network components

In the estimation procedure discussed in Sections 3.1 and 3.2, we assumed that the components x_i were 'directly' observed and that 'all' of them were observed. In this section, we now consider the case where the observed time courses are affine linear transformations $y_1, \ldots, y_M$ of $x_1, \ldots, x_N$ and the number M of observed time courses is smaller than the total number of network components N. The flowchart shown in Fig. 2 illustrates the single steps of the estimation procedure.

As an example for non-direct observations, consider the motif depicted in Fig. 1*c*, which is assumed to follow exactly the same dynamics as that in Fig. 1*b* and can therefore be described in terms of the ODEs given in (2). Suppose that one can now only measure time courses of the observation functions $y_1(t) = x_1(t)$ and $y_2(t) = bx_2(t) + c$ for scalars $b \neq 0$ and c. The ODEs given in (2) can then be translated to

$$\dot{y}_m(t) = \eta_m(\boldsymbol{\kappa}, \boldsymbol{y}(t)) + \tilde{a}_m h(t) \tag{17}$$

with appropriate η_m, $\tilde{a}_m$ depending on $\boldsymbol{a}$, b and c and κ being the collection of the interaction rates $\boldsymbol{k}$ and transformation parameters b and c. As the observation functions $\boldsymbol{y}$ are affine linear transformations of the network components $\boldsymbol{x}$, we can extract the hidden influence following (7) and (8)

$$\hat{h}^{\tilde{\boldsymbol{a}}}(t_j) = \frac{1}{N^{\tilde{\boldsymbol{a}}}} \sum_{\{i:\tilde{a}_i \neq 0\}} \frac{\dot{\hat{y}}_i^{\mathrm{spl}}(t_j) - \eta_i(\boldsymbol{\kappa}, \hat{\boldsymbol{y}}^{\mathrm{spl}}(t_j))}{\tilde{a}_i} \tag{18}$$

Note that, for non-linear observation functions, we cannot directly apply our method possibly because of, for example, quadratic or higher order terms of $h(t)$ in (17); however, in the above case, one can proceed in a manner analogous to that given in Sections 3.1 and 3.2 for the estimation of h and $\tilde{\boldsymbol{a}}$ if both y_1 and y_2 are observed.

As an example of partial observation, we can assume that only y_1 is observed. As the two-dimensional (2D) ODE system given in (1) contains no redundant equation, the dynamics of interest are fully described by a network of only two components. Hence, in addition to the observed variable y_1, we include one latent component (LC) y_2 in our analysis, for example, $y_2 = x_2$ or $y_2 = bx_2 + c$, as discussed above. More generally, we consider a network with observed components $y_1, \ldots, y_M$ and unobserved components $y_{M+1}, \ldots, y_N$. The estimation of a hidden influence and its weights changes slightly as opposed to the fully-observed case because there is no spline approximation possible for the time courses of $y_{M+1}, \ldots, y_N$.

In this case, we approximate $y_1, \ldots, y_M$ and their derivatives as before [see (5) and (6)]. Furthermore, we approximate $y_{M+1}, \ldots, y_N$ by their solutions of the N-dimensional ODE system given by (17) with $h \equiv 0$. For simplicity, we denote these approximations by $\hat{y}_i^{\mathrm{spl}}$ for all i, although there are no splines involved for $i > M$. The starting values $y_i(t_0)$ are treated as additional unknown parameters. Owing to the ODE-based derivation of $\hat{y}_i^{\mathrm{spl}}$ for the latent variables, estimation of the corresponding $\tilde{a}_i$ is not feasible in the first step of the estimation procedure. Hence, we restrict the components of $\tilde{\boldsymbol{a}}$ to be zero for $i > M$. For a given weight vector, the hidden influence is estimated through (18). In the second step, the likelihood function results as in (13) as a product over all observed components ($i \in \{1, \ldots, M\}$) and observation times ($j \in \{0, \ldots, n\}$). Maximisation of the likelihood function yields estimates for h and $\tilde{\boldsymbol{a}}$ for all $i \in \{1, \ldots, N\}$.

4 Results

In this section, we demonstrate several different applications of our method. In Section 4.1, the prediction of the time course of a hidden component is evaluated. In Section 4.2, we present our method as a tool that guides the reconstruction of a previously misspecified network. Finally, in Section 4.3, we identify a LC in the JAK2-STAT5 signalling pathway using real-world data.

4.1 Method performance evaluated with synthetic data

To evaluate the performance of our method, we conduct several simulation studies. All test runs are performed with the statistical software R [31]. We examine the robustness of our method by varying the noise intensity of the simulated data. In addition, we evaluate networks of different sizes and study the dependence of the results on the number of unobserved components.

The parameters $\boldsymbol{k}$ and $\boldsymbol{a}$ are chosen at random for each simulation run, and conditioned on these, we generate artificial data at 30 equally spaced time points. We use log-normal noise (see Appendix 1), and the three noise levels that we consider are low ($\sigma = 0.01$), medium ($\sigma = 0.1$) and high ($\sigma = 0.3$). In the simulation, we allow only linear interactions between the network components which, indicates that the structure of the ODEs can be summarised as

$$\dot{x}_i(t) = \sum_{u=1}^{N} \big(k_{iu} x_u(t) - k_{ui} x_i(t)\big) + a_i h(t) \tag{19}$$

with uniformly distributed k_{iu} in [0, 1] for describing the reaction strength between the ith and uth components and uniformly distributed a_i in $[-1, 1]$.

We use the same hidden influence for each simulation run, thus producing comparable results. After application of our estimation procedure, the resulting fit quality is measured by

$$s = \frac{1}{n+1} \sum_{j=0}^{n} |\hat{h}(t_j) - h(t_j)| \tag{20}$$

We estimate rates $\boldsymbol{a}$ with the forward selection technique. As illustrated in Fig. 3, results of 100 simulations indicate that a smaller network size and a smaller fraction of observed components lead to increasingly poor model fitting performance.

Only small differences are observed between low and high noise intensities, indicating that our method can accommodate a high degree of noise while extracting the relevant information from the data. In addition, it appears that the network size plays only a minor role with regard to the estimation quality of our method because the scores for larger networks decrease only slightly.

Our approach also yields estimates of the time course of the hidden component, which we can compare with the true hidden component used to generate the data. Fig. 4 shows the mean and 5 and 95% pointwise quantile time courses of three exemplary simulation scenarios. The shape of the hidden influence is reproduced satisfactorily, albeit differently. For a network comprising three components and a high degree of noise, the estimates produce additional fluctuations that are not present in the true time course and the confidence intervals are very broad. For a

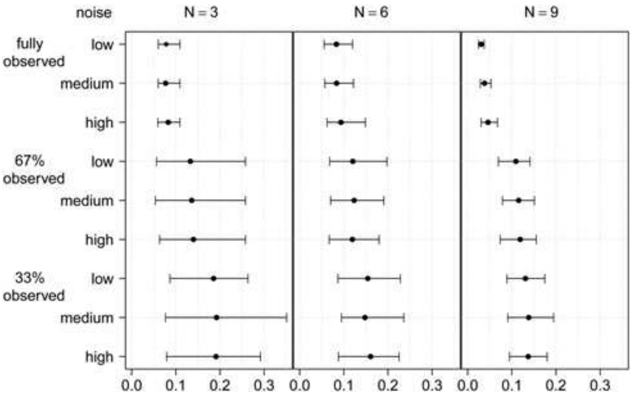


Fig. 3 *Description of the simulation studies used to evaluate the performance of our method*

For 27 different combinations of network size (3, 6 and 9 components), noise intensity (‘low’ ≡ $\sigma = 0.01$, ‘medium’ ≡ $\sigma = 0.1$ and ‘high’ ≡ $\sigma = 0.3$) and ratio of observed to unobserved components (100% observed, 67% observed and 33% observed), 100 different simulated networks are created and the mean as well as the 5 and 95% quantiles of the error measurement in (20) are displayed for each combination. All interaction rates between components are chosen randomly. It holds that, the smaller the value of s, the better the estimated time course

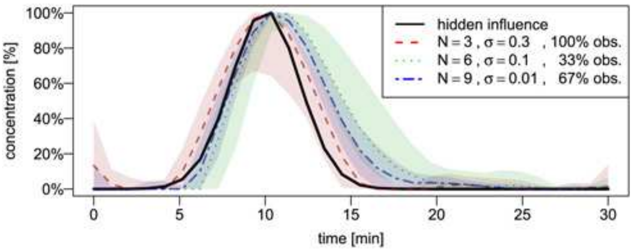


Fig. 4 *Time course of the hidden influence*

Hidden influence used in the simulations (solid line) and the mean and 5 and 95% pointwise quantile courses of three exemplary simulation scenarios with different parameters defined as follows: a fully observed network of size 3 with high noise intensity (dashed); a partially observed network (33%) of size 6 with medium noise intensity (dotted); a partially observed network (67%) of size 9 with low noise intensity (dashed-dotted). Mean and confidence intervals are based on 100 estimates $\hat{h}(t)$

larger network size (6 or 9 components), the estimates become more stable and recover the peak of the true time course; however, the second part of the peak is slightly overestimated because of the network being partially observed.

4.2 Recovering misspecified networks with a latent variable

Our method can be used for a guided repair of a wrongly specified network. We demonstrate this using artificial data in a further example. In this example, the network from which we simulate time-dependent observations consist of four players that are connected with each other in a forward cascade ending with a feedback loop, as shown in Fig. 5*a*. However, we assume that the initial hypothesis suggests a network structure with a missing feedback loop. Furthermore, we do not assume known reaction rates $\boldsymbol{k}$; thus, we incorporate the fitting of $\boldsymbol{k}$ into the application of our method.

Fig. 5 shows that we can simultaneously reconstruct the misspecified network structure and estimate $\boldsymbol{k}$ very well. The model without a feedback loop is best estimated with parameters $\hat{\boldsymbol{k}} = (0.05, 0.06, 0.01)^{\mathrm{T}}$ and has a BIC value of 1376.78 [Fig. 5*b* (1)]. The identified LC has a positive interaction with the first species $x_1(t)$ and a negative interaction with the last species $x_4(t)$. This suggests that a feedback loop might be missing in the network specification. The corresponding BIC value is 857.48. The estimate $\hat{\boldsymbol{k}} = (0.15, 0.29, 0.20)^{\mathrm{T}}$ is close to the true $\boldsymbol{k}$ [Fig. 5*b* (2,3)]. The constellation of interactions between the hidden component and $\boldsymbol{x}$ suggests a feedback loop. Inclusion of this loop further improves the BIC value to 854.15 and slightly alters $\hat{\boldsymbol{k}} = (0.14, 0.30, 0.20)^{\mathrm{T}}$ [Fig. 5*b*(4)]. Subsequent application of our method does not identify a LC which significantly improves the model fit [Fig. 5*b*(5, 6)].

This example demonstrates the ability of our method to recover misspecified network structures. We repeated the presented example with random data 100 times and concluded the same missing feedback in 97% of the repetitions (results not shown). However, in general networks, misspecifications may occur in very a complex manner; thus, overall it will be difficult to always apply our method under all conditions. Nevertheless, even if the network structure cannot be recovered completely, a hidden component may indicate which network components are candidates for refining the network structure and whether inhibition or activation of certain network components is more likely to improve a given model.

4.3 JAK2-STAT5 signalling pathway

The simulation study in Section 4.1 has shown that our estimation procedure can reliably detect and quantify a hidden influence on a given network. We now focus on models and real-world data from the literature. A prominent and well-studied example is the erythropoietin (Epo) signalling pathway which tranduces Epo stimulation via JAK2-STAT5 [32]. Epo signalling plays an important role in proliferation, differentiation and survival of erythroid progenitor cells [33]. After binding of the Epo hormone to its receptor, STAT5 can also bind. Subsequently, dimerisation of STAT5 results in a translocation to the nucleus where the STAT5 dimer acts as a transcription factor.

Several models exist which explain the molecular dynamics in various ways [34–37]. We analyse immunoblotting data which have already been analysed by a basic model [34] using the following system of ODEs

$$\begin{aligned}\dot{x}_1 &= -k_1 x_1 \mathrm{EpoR_A}\\ \dot{x}_2 &= -k_2 x_2^2 + k_1 x_1 \mathrm{EpoR_A}\\ \dot{x}_3 &= -k_3 x_3 + 0.5 k_2 x_2^2\\ \dot{x}_4 &= +k_3 x_3\end{aligned} \tag{21}$$

Here, the different states of STAT5 are cytoplasmic unphosphorylated STAT5 (denoted by x_1), cytoplasmic phosphorylated monomeric STAT5 (x_2), cytoplasmic phosphorylated dimeric STAT5 (x_3) and STAT5 in the nucleus (x_4). $\mathrm{EpoR_A}$ describes the Epo-induced tyrosine phosphorylation which can be measured up to a scaling factor. The initial values are $x_1(0) > 0$ (to be estimated) and $x_2(0) = x_3(0) = x_4(0) = 0$.

In the above mentioned literature, the model given in (21) is further refined by, for example, introducing an additional transition from nuclear STAT5 to the cytoplasmic unphosphorylated state, thus completing the loop from x_1 to x_4, or introducing time delays. These model refinements typically lead to an improved representation of the measured data, confirmed by, for example, likelihood ratio tests, information criteria (AIC/BIC) or Bayes factors. To start from the best-known model, we extend the refined model by incorporating a hidden influence. As a first step, we consider

$$\begin{aligned}\dot{x}_1 &= -k_1 x_1 \mathrm{EpoR_A} + a_1 h\\ \dot{x}_2 &= -k_2 x_2^2 + k_1 x_1 \mathrm{EpoR_A} + a_2 h\\ \dot{x}_3 &= -k_3 x_3 + 0.5 k_2 x_2^2 + a_3 h\end{aligned} \tag{22}$$

Here, we do not consider the fourth row of (22) because we have no information about x_4 as we use the measurements of experiment

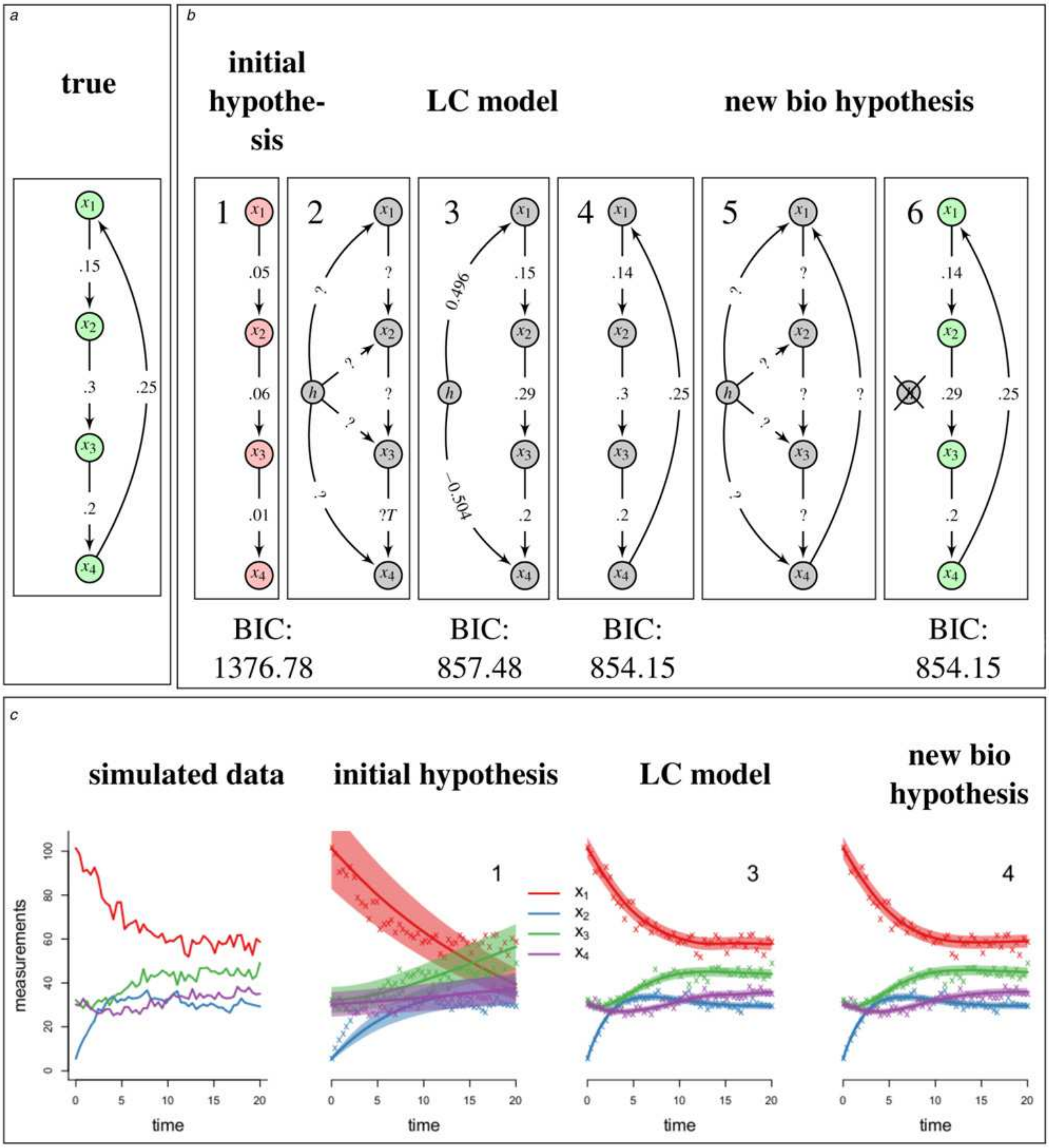


Fig. 5 *True course of hidden influence*

a True network from which the data is simulated
b Schematic representation of the workflow
(1) Misspecified network
(2), (3) LC model suggests a feedback loop
(4)–(6) Model with feedback loop cannot be further improved
c Corresponding data and model fits

number 1 provided as supporting material in [34]. These measurements describe the total amount of cytoplasmic tyrosine phosphorylated STAT5, that is, $y_1 = k_5(x_2 + 2x_3)$, the total amount of cytoplasmic STAT5, $y_2 = k_6(x_1 + x_2 + 2x_3)$, and the Epo-induced tyrosine phosphorylation, $y_3 = k_7\mathrm{EpoR_A}$. All three measured time-varying variables were experimentally quantified up to scaling factors denoted by k_5, k_6 and k_7. Evidently, only transformations of the ODE components x_1, …, x_3 are observed. Furthermore, a system comprising only y_1, y_2 and y_3 cannot be described in closed form. For that reason, we also include the auxiliary variable x_3. The differential equations for the observed and LCs are as follows

$$
\begin{aligned}
\dot{y}_1 &= \frac{k_1 k_5 y_2 y_3}{k_6 k_7} - \frac{k_1 y_1 y_3}{k_7} - 2k_3 k_5 x_3 + k_5(a_2 + 2a_3)h \\
\dot{y}_2 &= -2k_3 k_6 x_3 + k_6(a_1 + a_2 + 2a_3)h \\
\dot{x}_3 &= -k_3 x_3 + \frac{k_2 y_1^2}{2k_5^2} - \frac{2k_2 y_1 x_3}{k_5} + 2k_2 x_3^2 + a_3 h
\end{aligned}
\tag{23}
$$

We further refine the model by completing the loop from x_4 to x_1 and including a time delay, as has been done previously [38]. The authors suggested the use of a linear chain trick [39] and introduced a delayed loop. Thus, two (or possibly more) additional variables in the system of differential equations are introduced

$$\begin{aligned}
\dot{x}_1 &= -k_1 x_1 \mathrm{EpoR_A} + 2k_4 z_2 + a_1 h \\
\dot{x}_2 &= -k_2 x_2^2 + k_1 x_1 \mathrm{EpoR_A} + a_2 h \\
\dot{x}_3 &= -k_3 x_3 + 0.5 k_2 x_2^2 + a_3 h \\
\dot{x}_4 &= +k_3 x_3 - k_4 z_2 \\
\dot{z}_1 &= \frac{1}{\tau}(x_3 - z_1) \\
\dot{z}_2 &= \frac{2}{\tau}(z_1 - z_2)
\end{aligned} \tag{24}$$

Analogously, we can transform these equations to counterparts depending on y_1, y_2 and y_3

$$\begin{aligned}
\dot{y}_1 &= \frac{k_1 k_5 y_2 y_3}{k_6 k_7} - \frac{k_1 y_1 y_3}{k_7} - 2k_3 k_5 x_3 + k_5(a_2 + 2a_3)h \\
\dot{y}_2 &= -2k_3 k_6 x_3 + 2k_4 k_6 z_2 + k_6(a_1 + a_2 + 2a_3)h \\
\dot{x}_3 &= -k_3 x_3 + \frac{k_2 y_1^2}{2k_5^2} - \frac{2k_2 y_1 x_3}{k_5} + 2k_2 x_3^2 + a_3 h \\
\dot{z}_1 &= \frac{1}{\tau}(x_3 - z_1) \\
\dot{z}_2 &= \frac{2}{\tau}(z_1 - z_2)
\end{aligned} \tag{25}$$

This representation captures the dynamics of the observed variables. The right-hand side of (25) depends on the observed components y_1 to y_3, the hidden component h, the unobserved component x_3 and the two artificially introduced delay variables z_1 and z_2. For this reason, we must estimate x_3, z_1 and z_2 prior to h. This is achieved by numerically computing the solution of the model given in (24) without considering the hidden component (i.e. $\boldsymbol{a} = \mathbf{0}$) and using the approximations for x_3, z_1 and z_2 arising from this model. Once these quantities are determined, we estimate the three transformed weighting coefficients $\tilde{\boldsymbol{a}} = (k_5(a_2 + 2a_3), k_6(a_1 + a_2 + a_3), a_3)$ and use the estimates for x_3, z_1 and z_2 as input in the new iteration. This procedure is repeated until convergence. Once $\tilde{\boldsymbol{a}}$ is successfully obtained, we simply calculate $\boldsymbol{a}$ from $\tilde{\boldsymbol{a}}$ up to the scaling factors k_5 and k_6.

Fig. 6 shows a schematic description of the estimated model given in (25). According to our estimation performed by best subset selection, the hidden component interacts only with the first and third states of STAT5. Interestingly, the interaction direction (activating x_1 and inhibiting x_3) hints at a translocation of STAT5 from its nuclear state to the cytoplasm as also hypothesised by other authors [34].

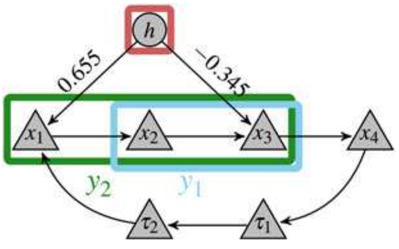


Fig. 6 *Schematic representation of the JAK2-STAT5 signalling pathway*

Four different states of STAT5 are regulated by a LC h with different weights, as estimated by our method. Observed variables y_1 and y_2 are linear combinations of the single states x_1 to x_3 τ_1 and τ_2 represent artificial delay variables

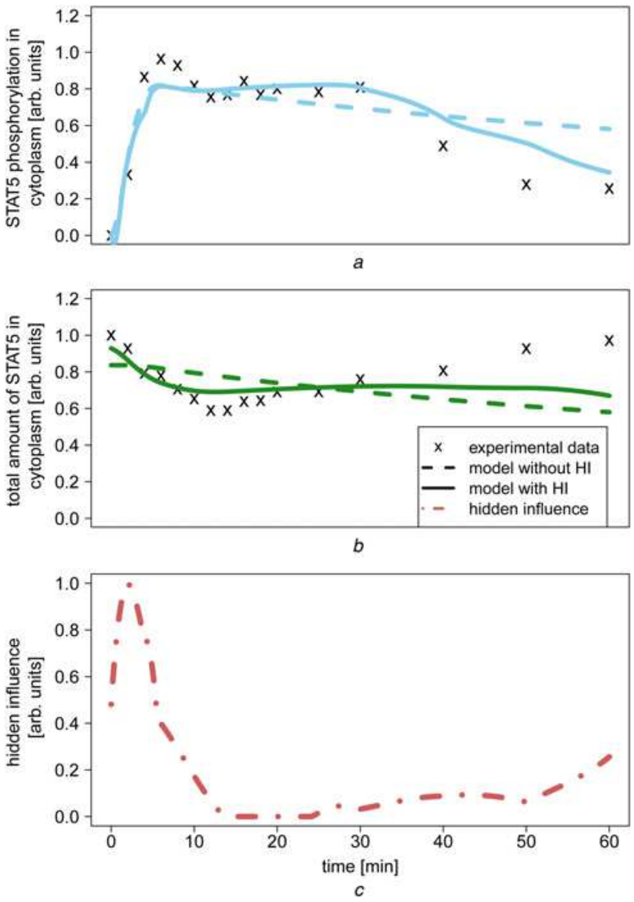


Fig. 7 *Experimental data and the estimated time courses of y_1 and y_2*

a Experimental data and model fitting for STAT5 phosphorylation in cytoplasm (y_1)
b Experimental data and model fitting for the total amount of STAT5 in cytoplasm (y_2)
Model that includes no hidden component is indicated by the dashed lines, whereas that which includes a hidden component is indicated by solid lines
Model with a hidden influence produces a time course that better fits the experimental data
c Estimated time course of the hidden component, which exhibits a strong peak at the beginning of the experiment, quickly drops to 0 and begins to increase again after 30 min

Figs. 7*a* and *b* show the experimental data and the estimated time courses of y_1 and y_2. The model with a hidden component h outperforms the model without h because it best represents the experimental data. Most importantly, the time course produced with a hidden component is considerably more flexible but does not overfit the data. The time course of the estimated hidden component (third panel of Fig. 7*c*) exhibits large values at the beginning of the experiment, decreases and then begins increasing after 30 min. Our interpretation of this behaviour is that an external quantity should be present at the beginning of the experiment (or shortly after); thus, the entire signalling pathway is kick-started. This external stimulus depletes completely and its influence slowly begins increasing after 30 min, bringing the entire system into equilibrium with the inhibition of the dimerised STAT5, and simultaneously the activation of the monomeric STAT5.

5 Conclusion

The main objective of this study is to provide a new method for model extension by introducing a hidden component to known networks. With the proposed method, we cannot only derive the relative time course of the hidden component but also predict the influence of the hidden component on all other network components.

We first fit splines to the observed components or to observation functions that are affine linear transformations of the former. The advantage of splines is that we can immediately compute the corresponding time derivative. On the basis of the observation error distribution, we apply maximum-likelihood estimation and model selection. By so doing, we can estimate a combination of the time course of a hidden component and the weights that lead to the best model in terms of data faithfulness without overfitting.

The method was applied to artificial data to test robustness and applicability. The results suggest a robust and good performance for the identification of the time course of the hidden component even in scenarios with high noise intensity.

One application of our method is the detection of misspecified networks. As a demonstration, we chose a network which included a feedback loop that was missing in the model specification. The loop was successfully recovered, thus providing a promising application variant of our technique. Our method, however, is not a tool for general network inference in its current form. An automation of the process by combining theory from network topology estimation with the proposed latent variable model presents a possible extension in the future work.

We applied the method to the well-studied JAK2-STAT5 signalling pathway. Model extension with a LC was performed on a system of ODEs with introduced delay. Our method could improve the model quality in terms of BIC and produced results which are in conformity with other methods suggested in the literature.

A key element in general model building is the estimation of parameters and possibly topology from data. Here we propose to interpret model estimation as a latent variable problem in a dynamical system. We target applications in which latent variables are influencing observations but not vice versa. A coupling in $h(t)$ in the sense of feedback of observed network components to the LC is possible; however, we mainly see two limitations of this approach. First, additional assumptions about $h(t)$ must be made, and second, including $h(t)$ into the system of ODEs limits its shape and does not allow for additional flexibility.

We currently only allow linear model extensions. In ongoing work, we extend our method to more general settings. More specifically, we consider multiple independent latent variables that influence a system of differential equations. In this scenario, in addition to the estimation of the latent time courses, we study possible ways of separation of the single variables. Modelling endogenous dependencies between latent and system variables without losing the flexibility of the hidden quantity is another extension that may be addressed in the future. Furthermore, although the proposed method was extensively tested in designed simulation scenarios, many additional challenges such as dependent errors, low sample sizes (in the sense of a very short time-series or a large amount of missing data) and non-linear ODEs remain, and at the same time, present additional method extensions. Finally, one might investigate additional model selection methods that are applicable to our method and produce stable and efficient results. Examples are established methods such as likelihood ratio tests, lasso [40] and elastic net [41]. Extending the method to Bayesian theory would further allow the application of Bayes factors [42] and thermodynamic integration [43, 44].

For the method presented here, we intentionally chose to separate the two major estimations into two steps, and both steps can be associated with two major modelling perspectives [3]. While fitting the spline parameters can be associated with a statistical perspective, exploiting the network structure for inference of the latent time-course and its interaction weights is closely connected to the mathematical modelling perspective. Model selection and thus network prediction, as discussed in Section 3.4, bring the method back to the statistical perspective. Formulating the problem as a joint optimisation of all parameters involved (reaction rates, spline parameters and noise parameters) is possible. This, however, leads to a considerably more complex and computationally intensive method.

As we demonstrate in Appendix 2, the performance of our method depends on the quality of the spline approximation. This quality will typically suffer if the modelled data are sparse, contain extreme outliers, are corrupted by a high amount of noise or the chosen spline representation cannot resemble fluctuations of the observed time-series appropriately.

The results of the proposed method can be employed as a promising aid for guiding future experiments, thus helping to complete the systems biology loop [45, 46] between experimental data and model analysis.

6 Acknowledgments

This work was supported by the European Union within the ERC grant LatentCauses. The authors are grateful to Jan Hasenauer, Sabine Hug, Nikola Müller and Daniel Schmidl for their helpful discussions and critical comments.

7 References

1 Aloy, P., Russell, R.B.: ‘Structural systems biology: modelling protein interactions’, *Nat. Rev. Mol. Cell Biol.*, 2006, **7**, (3), pp. 188–197

2 Kitano, H.: ‘Computational systems biology’, *Nature*, 2002, **420**, (6912), pp. 206–210

3 Emmert-Streib, F., Dehmer, M., Haibe-Kains, B.: ‘Untangling statistical and biological models to understand network inference: the need for a genomics network ontology’, *Front. Genet.*, 2014, **5**, 299 doi: 10.3389/fgene.2014.00299

4 Hornstein, E., Shomron, N.: ‘Canalization of development by microRNAs’, *Nat. Genet.*, 2006, **38**, pp. S20–S24

5 Chickarmane, V., Peterson, C.: ‘A computational model for understanding stem cell, trophectoderm and endoderm lineage determination’, *PLoS One*, 2008, **3**, (10), pp. e3478

6 Lin, T., Ambasudhan, R., Yuan, X., *et al.*: ‘A chemical platform for improved induction of human iPSCs’, *Nat. Methods*, 2009, **6**, (11), pp. 805–808

7 Borowiak, M., Maehr, R., Chen, S., *et al.*: ‘Small molecules efficiently direct endodermal differentiation of mouse and human embryonic stem cells’, *Cell Stem Cell*, 2009, **4**, (4), pp. 348–358

8 Ambroise, C., Chiquet, J., Matias, C.: ‘Inferring sparse Gaussian graphical models with latent structure’, *Electron. J. Stat.*, 2009, **3**, pp. 205–238

9 Gao, P., Honkela, A., Rattray, M., Lawrence, N.D.: ‘Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities’, *Bioinformatics*, 2008, **24**, (16), pp. i70–i75

10 Honkela, A., Girardot, C., Gustafson, E.H., *et al.*: ‘Model-based method for transcription factor target identification with limited data’, *Proc. Nat. Acad. Sci.*, 2010, **107**, (17), pp. 7793–7798

11 Ramb, R., Eichler, M., Ing, A., *et al.*: ‘The impact of latent confounders in directed network analysis in neuroscience’, *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, 2013, **371**, (1997), p. 20110612

12 Hoyer, P.O., Shimizu, S., Kerminen, A.J.: ‘Estimation of linear, non-Gaussian causal models in the presence of confounding latent variables’, *arXiv preprint cs/0603038*, 2006

13 Monecke, A., Leisch, F.: ‘semPLS: structural equation modeling using partial least squares’, *J. Stat. Softw.*, 2012, **48**, (3), pp. 1–32

14 Bollen, K.A.: ‘Structural equation models’ (Wiley Online Library, 1998)

15 Gilbert, D., Fuß, H., Gu, X., *et al.*: ‘Computational methodologies for modelling, analysis and simulation of signalling networks’, *Brief. Bioinform.*, 2006, **7**, (4), pp. 339–353

16 Blöchl, F., Theis, F.J.: ‘Estimating hidden influences in metabolic and gene regulatory networks’, in Adali, T., Jutten, C., Marcos, J., Romano, T., Baros, A. K. (Eds.): ‘Independent component analysis and signal separation’ (Springer, 2009), pp. 387–394

17 Bar-Joseph, Z., Gerber, G.K., Gifford, D.K., Jaakkola, T.S., Simon, I.: ‘Continuous representations of time-series gene expression data’, *J. Computat. Biol.*, 2003, **10**, (3–4), pp. 341–356

18 De Boor, C.: ‘A practical guide to splines’ (Springer Verlag, 2001), vol. 27

19 Eilers, P.H., Marx, B.D.: ‘Flexible smoothing with B-splines and penalties’, *Stat. Sci.*, 1996, **11**, (2), pp. 89–102

20 Ramsay, J.O., Silverman, B.W.: ‘Functional data analysis’ (Springer, New York, 2005)

21 Paulsson, J.: ‘Summing up the noise in gene networks’, *Nature*, 2004, **427**, (6973), pp. 415–418

22 Raser, J.M., O’Shea, E.K.: ‘Noise in gene expression: origins, consequences, and control’, *Science*, 2005, **309**, (5743), pp. 2010–2013

23 Ross, S.: ‘Differential equations’ (JohnWiley & Sons, New York, 1984), vol. 3

24 Cramér, H.: ‘Mathematical methods of statistics’ (Princeton University Press, 1945), vol. 9

25 Rao, C.R.: ‘Information and accuracy attainable in the estimation of statistical parameters’, *Bull. Calcutta Math. Soc.*, 1945, **37**, (3), pp. 81–91

26 Butcher, J.C.: ‘The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods’ (Wiley-Interscience, 1987)

27 Zacks, S.: ‘The theory of statistical inference’ (Wiley, New York, 1971), vol. 34

28 Akaike, H.: ‘A new look at the statistical model identification’, *IEEE Trans. Autom. Control*, 1974, **19**, (6), pp. 716–723

29 Schwarz, G.: ‘Estimating the dimension of a model’, *Ann. Stat.*, 1978, **6**, (2), pp. 461–464

30 Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., Tibshirani, R.: ‘The elements of statistical learning’ (Springer, 2009), vol. 2.1

31 R Development Core Team: ‘R: a language and environment for statistical computing’ (R Foundation for Statistical Computing, Vienna, Austria, 2011)

32 Darnell, J.: ‘STATs and gene regulation’, *Science*, 1997, **277**, (5332), pp. 1630–1635

33 Klingmüller, U., Bergelson, S., Hsiao, J., Lodish, H.: ‘Multiple tyrosine residues in the cytosolic domain of the erythropoietin receptor promote activation of STAT5’, *Proc. Nat. Acad. Sci.*, 1996, **93**, pp. 8324–8328

34 Swameye, I., Müller, T., Timmer, J., Sandra, O., Klingmüller, U.: ‘Identification of nucleocytoplasmic cycling as a remote sensor in cellular signaling by databased modeling’, *Proc. Nat. Acad. Sci. USA*, 2003, **100**, (3), p. 1028

35 Toni, T., Stumpf, M.: ‘Simulation-based model selection for dynamical systems in systems and population biology’, *Bioinformatics*, 2010, **26**, (1), pp. 104–110

36 Timmer, J., Müller, T., Swameye, I., Sandra, O., Klingmüller, U.: ‘Modeling the nonlinear dynamics of cellular signal transduction’, *Int. J. Bifurcation Chaos Appl. Sci. Eng.*, 2004, **14**, pp. 2069–2080

37 Müller, T., Faller, D., Timmer, J., Swameye, I., Sandra, O., Klingmüller, U.: ‘Tests for cycling in a signalling pathway’, *J. R. Stat. Soc. C, Appl. Stat.*, 2004, **53**, (4), pp. 557–568

38 Nikolov, S., Georgiev, G., Kotev, V., Wolkenhauer, O.: ‘Stability analysis of a time delay model for the JAK-STAT signalling pathway’, *Series Biomech.*, 2007, **23**, (1), pp. 52–65

39 Fall, C.: ‘Computational cell biology’ (Springer Verlag, 2002), vol. 20

40 Tibshirani, R.: ‘Regression shrinkage and selection via the lasso’, *J. R. Stat. Soc. B, Methodol.*, 1996, **58**, (1), pp. 267–288

41 Zou, H., Hastie, T.: ‘Regularization and variable selection via the elastic net’, *J. R. Stat. Soc. B, Stat. Methodol.*, 2005, **67**, (2), pp. 301–320

42 Kass, R.E., Raftery, A.E.: ‘Bayes factors’, *J. Am. Stat. Assoc.*, 1995, **90**, (430), pp. 773–795

43 Kirk, P., Thorne, T., Stumpf, M.P.: ‘Model selection in systems and synthetic biology’, *Curr. Opinion Biotechnol.*, 2013, **24**, (4), pp. 767–774

44 Schmidl, D., Hug, S., Li, W.B., Greiter, M.B., Theis, F.J.: ‘Bayesian model selection validates a biokinetic model for zirconium processing in humans’, *BMC Syst. Biol.*, 2012, **6**, (1), p. 95

45 de Ridder, D., de Ridder, J., Reinders, M.J.: ‘Pattern recognition in bioinformatics’, *Brief. Bioinform.*, 2013, **14**, (3), pp. 633–647

46 Endler, L., Rodriguez, N., Juty, N., *et al.*: ‘Designing and encoding models for synthetic biology’, *J. R. Soc. Interface*, 2009, **6**, pp. S405–S417

8 Appendix

8.1 Appendix 1: Log-normally distributed multiplicative noise

Normally distributed error terms, as in (10), imply that the noise level is independent of the magnitude of the measurements. Although this assumption is commonly made, it is not appropriate for all biological applications, for example, when concentrations are measured over time. This assumption is particularly problematic for concentrations close to zero because the model description of $x_i^{\text{obs}}(t_j)$ can then easily become negative. This difficulty is avoided by assuming multiplicative log-normally distributed noise

$$x_i^{\text{obs}}(t_j) = x_i(t_j) \cdot \varepsilon_{ij}, \quad \varepsilon_{ij} \overset{\text{iid}}{\sim} \mathcal{LN}\left(-\frac{\sigma^2}{2}, \sigma^2\right)$$

The parameter choice of the log-normal distribution is motivated because of the implication $\mathbb{E}\left(x_i^{\text{obs}}(t_j)|x_i(t_j)\right) = x_i(t_j)$.

The distribution of ε_{ij} immediately propagates to the measurements

$$x_i^{\text{obs}}(t_j)|x_i(t_j) \overset{\text{iid}}{\sim} \mathcal{LN}\left(\log(x_i(t_j)) - \frac{\sigma^2}{2}, \quad \sigma^2\right)$$

As discussed in Section 3.2, we exploit the ODE structure of our system which leads to the approximate distribution

$$x_i^{\text{obs}}(t_j)|\hat{x}_i^{\text{ode},\boldsymbol{a}}(t_j) \overset{\text{iid}}{\sim} \mathcal{LN}\left(\log(\hat{x}_i^{\text{ode},\boldsymbol{a}}(t_j)) - \frac{\sigma^2}{2}, \quad \sigma^2\right)$$

The parameter σ^2 is estimated as

$$\hat{\sigma}^2_{\text{ML}} = -2 + 2\sqrt{1 + \frac{\sum_{i=1}^{N}\sum_{j=0}^{n}\left[\log\left(\left(x_i^{\text{obs}}(t_j)\right)/\left(\hat{x}_i^{\text{ode},\boldsymbol{a}}(t_j)\right)\right)\right]^2}{N(n+1)}}$$

The diagonal elements of the expected Fisher information matrix are derived as

$$I_k(\boldsymbol{a}, \sigma^2) = \begin{cases} \dfrac{1}{\sigma^2}\sum\limits_i\sum\limits_j\left(\dfrac{(\partial/\partial a_k)\hat{x}_i^{\text{ode},\boldsymbol{a}}(t_j)}{\hat{x}_i^{\text{ode},\boldsymbol{a}}(t_j)}\right)^2 & k \le N \\ \dfrac{N(n+1)}{2}\left(\dfrac{1}{\sigma^4} + \dfrac{1}{2\sigma^2}\right) & k = N+1 \end{cases}$$

8.2 Appendix 2: Example for parameter uncertainty calculation

In this section, we evaluate the goodness of fit for parameters associated with a specific example. For simplicity, we choose a small network comprising two species and normally distributed measurement noise. The two parameters of interest are a_1 and σ^2. We analytically compute the CRLB for both parameters and examine whether it is achieved using a simulation with known parameters.

The specific network is described by (2) and Fig. 1*b* and we consider a network size $N = 2$ and linear combinations of the components. Without loss of generality, we assume $a_2 = 1 - a_1$ and $a_1 \in (0, 1)$.

The CRLB for parameter a_1 involves $(\partial/\partial a_1)\hat{x}_i^{\text{ode},\boldsymbol{a}}(t_j)$. The value of this term depends on the numerical method chosen for solving the ODE. If the Euler method is used, the recursive solution has the form

$$\begin{aligned} \hat{x}_1^{\boldsymbol{a}}(t_{j+1}) &= \hat{x}_1^{\boldsymbol{a}}(t_j) + \Delta(c_1\hat{x}_1^{\boldsymbol{a}}(t_j) + c_2\hat{x}_2^{\boldsymbol{a}}(t_j) + a_1\hat{h}^{\boldsymbol{a}}(t_j)) \\ \hat{x}_2^{\boldsymbol{a}}(t_{j+1}) &= \hat{x}_2^{\boldsymbol{a}}(t_j) + \Delta(c_3\hat{x}_2^{\boldsymbol{a}}(t_j) + c_4\hat{x}_1^{\boldsymbol{a}}(t_j) + (1 - a_1)\hat{h}^{\boldsymbol{a}}(t_j)) \end{aligned}$$

with $c_1 = -k_2 - k_1$, $c_2 = k_3$, $c_3 = -k_4 - k_3$, $c_4 = k_2$ and Δ denoting the time step, and $\hat{x}_i^{\boldsymbol{a}}$ being a shortened version of $\hat{x}_i^{\boldsymbol{a},\text{ode}}$. We assume that the initial values $x_i(t_0)$ do not depend on a_1. One can then show by full induction that, for $j = 1, \ldots, n$

$$\begin{aligned} \frac{\partial x_1(t_j)}{\partial a_1} &= \frac{F_1(t_j)}{(1 - a_1)^2} - \frac{F_2(t_j)}{a_1^2} \\ \frac{\partial x_2(t_j)}{\partial a_1} &= \frac{F_3(t_j)}{a_1^2} + \frac{F_4(t_j)}{(1 - a_1)^2} \end{aligned} \qquad (26)$$

with

$$\boldsymbol{F}(t_{j+1}) = \boldsymbol{F}(t_j) + \Delta\begin{pmatrix} c_1F_1(t_j) + c_2F_4(t_j) + \frac{1}{2}\hat{h}_2^0(t_j) \\ c_1F_2(t_j) - c_2F_3(t_j) \\ c_3F_3(t_j) - c_4F_2(t_j) - \frac{1}{2}\hat{h}_1^0(t_j) \\ c_3F_4(t_j) + c_4F_1(t_j) \end{pmatrix}$$

and $\boldsymbol{F}(t_0) = (0, 0, 0, 0)^{\text{T}}$. $\hat{h}_i^0(t)$ are the unweighted estimates of the time course of the hidden component, as described in (7). The entries of $\boldsymbol{F}(t_j)$ are independent of a_1. For given σ^2, $\boldsymbol{k}$, a_1, spline approximations of $\boldsymbol{x}(t)$ and data dimensions, (26) allows the analytical computation of the expected Fisher information $I(a_1)$, and hence that of the CRLB $I^{-1}(a_1)$. For this specific example, the expected Fisher information matrix has a diagonal form and $I^{-1}(\sigma^2)$ equals $\frac{2\sigma^4}{N(n+1)}$.

The just derived CRLBs are lower bounds for the MSE of the maximum-likelihood estimates. Our estimation procedure, however, consists of two steps: spline approximation and maximum-likelihood estimation, each entailing uncertainty in the parameter estimates. We hence computed Monte Carlo estimates for the MSE of a_1 and σ^2 using two different approaches. First, we used the true hidden time course during the estimation procedure. The resulting empirical MSE is the one resulting from the maximum-likelihood step and is bounded below by Γ^{-1}. Second, we estimated the hidden time course as well. The resulting MSE is slightly larger and accounts for the uncertainty of the overall estimation procedure.

The results of the simulation are shown in Table 1. We examine different combinations of σ^2 and $\boldsymbol{a}$. For each combination, we simulate 500 time courses at 100 time points and estimate the parameters. We numerically compute the MSE of these 500 estimates. In the table, we show this MSE and the corresponding CRLB for a given parameter combination.

The results of Table 1 show that the CRLB is achieved for σ^2 if we consider only maximum-likelihood approximation. If we additionally consider the uncertainty introduced by the spline approximation, the ratio between MSE and CRLB increases slightly. For a_1, we observe a similar result in that the ratio increases if we consider spline approximation, while, nevertheless, providing MSE values that are very close to the corresponding CRLB. Another interesting result is the increase in fit quality for smaller σ^2 values, as we already discussed in Section 3.3.

Table 1 Results of the parameter uncertainty simulation

		MSE (CRLB)	
		$\boldsymbol{a} = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$	$\boldsymbol{a} = \begin{pmatrix} 0.9 \\ 0.1 \end{pmatrix}$
maximum-likelihood approximation			
$\sigma = 1$	a_1	9.90×10^{-6} (1.63×10^{-6})	3.73×10^{-9} (9.27×10^{-10})
	σ^2	1.07×10^{-2} (1.00×10^{-2})	1.02×10^{-2} (1.00×10^{-2})
$\sigma = 0.5$	a_1	2.33×10^{-6} (4.08×10^{-7})	9.04×10^{-10} (2.32×10^{-10})
	σ^2	6.19×10^{-4} (6.25×10^{-4})	6.38×10^{-4} (6.25×10^{-4})
$\sigma = 0.1$	a_1	9.06×10^{-8} (1.62×10^{-8})	3.54×10^{-11} (9.27×10^{-12})
	σ^2	1.04×10^{-6} (1.00×10^{-6})	9.83×10^{-7} (1.00×10^{-6})
maximum-likelihood and spline approximations			
$\sigma = 1$	a_1	1.55×10^{-4} (1.24×10^{-5})	5.23×10^{-6} (1.13×10^{-7})
	σ^2	7.59×10^{-2} (1.00×10^{-2})	1.33×10^{-1} (1.00×10^{-2})
$\sigma = 0.5$	a_1	1.62×10^{-5} (3.13×10^{-6})	9.57×10^{-7} (3.44×10^{-8})
	σ^2	4.29×10^{-3} (6.25×10^{-4})	5.44×10^{-3} (6.25×10^{-4})
$\sigma = 0.1$	a_1	6.22×10^{-7} (1.25×10^{-7})	2.69×10^{-8} (1.08×10^{-9})
	σ^2	6.24×10^{-6} (1.00×10^{-6})	4.31×10^{-6} (1.00×10^{-6})

MSE and corresponding CRLB (in parentheses) for the two parameters of interest.