IST AUSTRIA
Institute of Science and Technology

Modular Parameter Identification of Biomolecular Networks

Moritz Lang and Jörg Stelling

Technical Report No. IST-2017-811-v1+1
Deposited at 20 Apr 2017 07:24
https://repository.ist.ac.at/811/1/modular$_{p}arameter_{i}dentification.pdf$

IST Austria (Institute of Science and Technology Austria)
Am Campus 1
A-3400 Klosterneuburg, Austria

SIAM J. Sci. Comput.
Vol. xx, No. x, pp. x–x

MODULAR PARAMETER IDENTIFICATION OF BIOMOLECULAR NETWORKS

MORITZ LANG[†][‡][§] AND JÖRG STELLING[†][¶]

Abstract. The increasing complexity of dynamic models in systems and synthetic biology poses computational challenges especially for the identification of model parameters. While modularization of the corresponding optimization problems could help reduce the ”curse of dimensionality”, abundant feedback and crosstalk mechanisms prohibit a simple decomposition of most biomolecular networks into sub-networks, or modules. Drawing on ideas from network modularization and multiple-shooting optimization, we here present a modular parameter identification approach that explicitly allows for such interdependencies. Interfaces between our modules are given by the experimentally measured molecular species. This definition allows deriving good (initial) estimates for the inter-module communication directly from the experimental data. Given these estimates, the states and parameter sensitivities of different modules can be integrated independently. To achieve consistency between modules, we iteratively adjust the estimates for inter-module communication while optimizing the parameters. After convergence to an optimal parameter set–but not during earlier iterations–the inter-module communication as well as the individual modules' state dynamics agree with the dynamics of the non-modularized network. Our modular parameter identification approach allows for easy parallelization; it can reduce the computational complexity for larger networks and decrease the probability to converge to sub-optimal local minima. We demonstrate the algorithm's performance in parameter estimation for two biomolecular networks, a synthetic genetic oscillator and a mammalian signaling pathway.

Key words. Modularization, Parameter Identification, Systems Biology

AMS subject classifications. 90C30,34A55,92C42

DOI.

1. Introduction.

The values for kinetic parameters in mathematical models for biomolecular networks are often not experimentally accessible, except indirectly through their influences on the measurable dynamics of biomolecular species. Solving the inverse problem of inferring kinetic parameters from such indirect experimental measurements is referred to as *parameter identification* [2, 17].

Parameter identification for large biomolecular networks poses particular challenges because high-dimensional parameter spaces need to be explored, and the “curse of dimensionality” [5] entails computation times that may increase exponentially with the dimension of the parameter space. In addition, parameter values in biomolecular networks are often poorly constrained by prior knowledge. For example, enzyme kinetic parameter values distribute over several orders of magnitude [4], which makes it often difficult to ascertain even rough estimates when parameter values cannot be determined experimentally.

Many different algorithmic solutions for parameter identification of biomolecular networks exist (see e.g. [28, 29, 34] for extensive reviews). Most approaches repeatedly simulate the complete model over the full time-domain. They mainly differ in how parameter sets are selected for simulation, given the outcomes of previous simulations. Recent developments of modern hybrid approaches are considered promising to increase efficiency and to guarantee robustness in global parameter optimization [34].

[†] Department of Biosystems Science and Engineering, ETH Zürich, and Swiss Institute of Bioinformatics, Mattenstrasse 26, 4058 Basel, Switzerland

[‡] Present address: IST Austria, Am Campus 1, 3400 Klosterneuburg, Austria

[§] moritz.lang@ist.ac.at

[¶] joerg.stelling@bsse.ethz.ch

Such hybrid approaches switch from a global (typically stochastic) to a local (typically deterministic) algorithm in promising parameter regions [3,15]. Hybrid methods can be considered as a further development of two-phase methods [29] such as multistart [8], clustering [9], or random linkage [27]. For the performance of such hybrid or two-phase approaches, the speed, the stability, and the domain of convergence to good optima of the local deterministic method are of central importance [3].

In contrast to the vivid research on the selection of parameter sampling points in (global) optimization, complementary approaches to deal with the ”curse of dimensionality” by decomposing the parameter identification problem into smaller subproblems have received less attention. For biomolecular networks, divide-and-conquer strategies are based on splitting the original network into several *modules*, each depending only on a subset of the parameters, and by performing a modular parameter identification [16]. Ideally, the parameter subsets of the modules are non-overlapping and of equal size, and each module can be optimized independently–or at least consecutively. Existing approaches to modular parameter estimation rely on decomposing a network into independent pathways [20], or on graph decomposition with sequential block ordering without [19] or with [22] iterative adjustment of numerical integration methods. However, for most biomolecular networks, such “sequential” approaches are not possible because of omnipresent feedback loops and crosstalk mechanisms. Local (e.g., per enzyme in a biomolecular reaction network) identification methods impose strict requirements on measured data and on correctness of the model topology [21] that can rarely be fulfilled in practice.

For more general approaches to a modular identification of biomolecular networks, the requirement of independent or consecutive optimization of modules has to be relaxed by allowing the dynamics in one module to be affected by changes in other modules’ parameters. Modular response analysis (MRA; see [10, 18], and [31] for a related approach) mathematically captures such inter-module dependencies. In MRA, the effects of parameter variations of unknown magnitude on (steady-state) species concentrations of a modularized network are used to determine the quantitative interactions (sensitivities) between modules. The approach relies on (i) local (*intra-level*) sensitivities of the steady-state species’ concentrations to parametric perturbations inside the respective module, assuming the dynamics of all other modules to be fixed, and (ii) global (*inter-level*) sensitivities of the steady-states to perturbations of the modules’ interface species’ concentrations [10, 18]. MRA has not been designed for parameter identification, but a similar concept might reduce the amount of sensitivity equations to be numerically integrated in a local parameter identification approach.

Modular response analysis has conceptual similarities with multiple-shooting [7, 24, 30], a local deterministic parameter identification method. Multiple-shooting partitions the time-domain into several non-overlapping intervals, akin to partitioning the network into modules in MRA. With each of these time intervals, separate initial conditions are associated. Thus, the states of a biomolecular network model, as well as its parameter sensitivity equations, can be numerically integrated separately for each time interval. In each iteration of the multiple-shooting algorithm, not only the parameter vector is updated, but also the initial conditions of the time intervals. These updates are defined such that after converging to a (locally) optimal parameter set–but not necessarily during earlier iterations–the species’ trajectories are continuous over the full time domain. Importantly, this constraint on the joint species’ trajectories, respectively the update of the intervals’ initial conditions, can be interpreted as equivalent to the interface or the communication between modules.

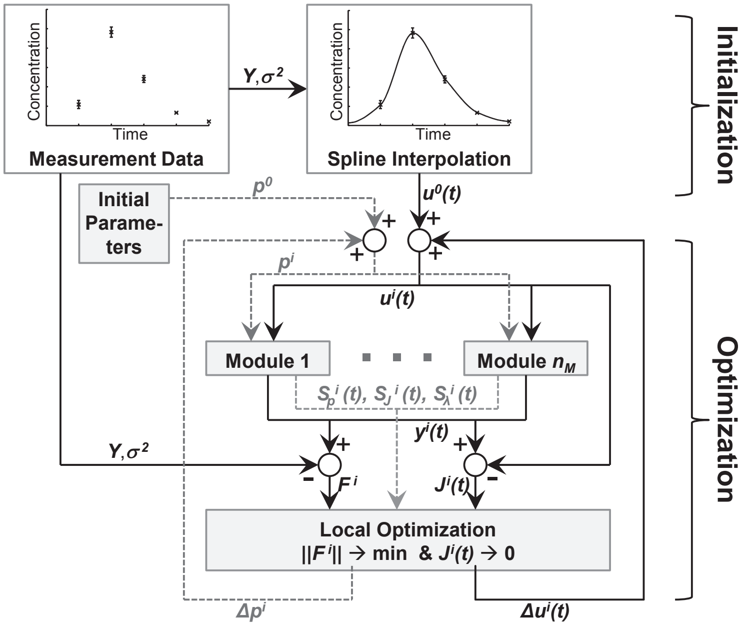

Fig. 1. *Simplified workflow of the iterative modular parameter identification approach. A biomolecular network is divided into $m = 1, \ldots, n_M$ possibly overlapping modules with states $\mathbf{x^i_m}(t)$ at time $t \geq 0$ in iteration $i \in \mathbb{N}_{\geq 0}$. Each module contains exactly one state $y^i_m(t) = \mathbf{c_m}^T \mathbf{x^i_m}$ which corresponds to an experimentally measured species, and to which we refer as the output of the module. The outputs $\mathbf{y^i}(t) = (y^i_1(t), \ldots, y^i_{n_M}(t))^T$ of all modules form the interface between the modules: for a given module m, the dynamics of all other modules $k \neq m$ might only (directly) depend on $y^i_m(t)$, but not on any other of the module's species. To simulate the modules independently, we substitute all dependencies on $y^i_m(t)$ in modules $k \neq m$ by the curves $u^i_m(t)$, to which we refer as the inputs of the modules. We initialize these inputs $\mathbf{u^i}(t) = (u^i_1(t), \ldots, u^i_{n_M}(t))^T$ by splines through experimental data consisting of mean concentrations $\mathbf{Y}$ and variances $\sigma^\mathbf{2}$ at different measurement times. Early in the optimization, the input curves $\mathbf{u^i}(t)$ and the output trajectories $\mathbf{y^i}(t)$ can differ substantially, and we denote this difference as the inconsistency $\mathbf{J^i}(t) = \mathbf{y^i}(t) - \mathbf{u^i}(t)$. Our algorithm iteratively adjusts the parameter values $\mathbf{p}^{i+1} = \mathbf{p}^i + \mathbf{\Delta p}^i$ as well as the input trajectories $\mathbf{u^{i+1}}(t) = \mathbf{u^i}(t) + \mathbf{\Delta u^i}(t)$ to simultaneously minimize the difference $\mathbf{F^i}$ between the model dynamics and the experimental data and the inconsistencies $\mathbf{J^i}(t)$ between inputs and outputs. For this purpose, we calculate the (local) sensitivities of the output trajectories with respect to perturbations of the parameter values ($\mathbf{S^i_p}(t)$), respectively of the input trajectories ($\mathbf{S^i_J}(t)$ and $\mathbf{S^i_\lambda}(t)$, see main text). After convergence, the inconsistencies are (approximately) zero, such that each module's states show (approximately) the same dynamics as the corresponding states of the non-modularized network.*

Thus, the sensitivities to the parameter values in each interval in multiple-shooting are equivalent to the local intra-level response coefficients, and the sensitivities with respect to the intervals' initial conditions to the global inter-level responses in MRA. Multiple-shooting has several advantageous features, including a comparatively high numerical robustness and the possibility to derive good initial estimates for the intervals' initial conditions directly from the experimental data, leading to improved convergence properties and the avoidance of sub-optimal local minima [7, 24].

Here, we present a modular parameter identification method for biomolecular networks (see Fig. 1) that uses ideas from MRA and multiple-shooting to combine the advantages of both approaches. As previously [23], we define modules such that all species in the modules' interfaces are experimentally measured. In this way good initial estimates of the interface species' dynamics can be derived directly from the experimental data–similar to multiple-shooting [24]. We use these estimates to (partly) insulate the modules from each other: each module can be separately simulated given these estimates. We iteratively optimize the parameters while at the same time consolidating the estimates for the interfaces' dynamics to eventually achieve consistency between the modules' dynamics. After convergence of our algorithm, the dynamics of the modules are consistent with the dynamics of the non-modularized network.

This article is structured as follows: the Methods section provides the mathematical problem statement, specifies our modularization approach, derives our modular parameter identification algorithm, and discusses its computational requirements as well as its (local) convergence guarantees. In the Results section, we exemplify the performance of our framework with respect to parameter identification for two biomolecular networks, a synthetic genetic oscillator [13] and a more complex mammalian signaling pathway [1].

2. Methods.

2.1. Problem Statement.

In this article, we consider dynamic models of biomolecular networks with n_X states (species) with concentrations $\mathbf{x}(t) \in \mathbb{R}^{n_X}_{\geq 0}$ at time $t \in [0, t_f]$. The species participate in n_R reactions with rates $\mathbf{v}(\mathbf{x}(t), \mathbf{p}) \in \mathbb{R}^{n_R}$ which depend on the states and n_P (unknown) parameters $\mathbf{p} \in \mathbb{R}^{n_P}$. The model of a network is given by the ordinary differential equations (ODEs)

$$\begin{aligned} \frac{d}{dt}\mathbf{x}(t) &= \mathbf{N}\mathbf{v}(\mathbf{x}(t), \mathbf{p}),\ \mathbf{x}(0) = \mathbf{x_0} \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t), \end{aligned} \tag{2.1}$$

with $\mathbf{N} \in \mathbb{R}^{n_X \times n_R}$ the stoichiometric matrix mapping the reaction rates to the species dynamics, and $\mathbf{x_0} \in \mathbb{R}^{n_X}_{\geq 0}$ the initial conditions. The matrix $\mathbf{C} \in \{0,1\}^{n_M \times n_X}$ maps the states of the model to $n_M > 1$ trajectories $\mathbf{y}(t) = (y_1(t), \ldots, y_{n_M}(t))^T$, referred to as the *outputs* of the network. Here, we require that $\mathbf{C}$ has exactly one non-zero entry in each row. Throughout this article, we assume the reaction rates $\mathbf{v}$ to be continuously differentiable with respect to $\mathbf{x}$ and $\mathbf{p}$, and that a unique solution $\mathbf{x}(t)$ to the initial value problem exists for $t \in [0, t_f]$ and all feasible parameter sets.

The inverse, or parameter identification problem is to find the optimal set of parameters that minimizes the difference between the model's dynamics and given experimental data. Here, we assume that this data represents measured mean concentrations $Y_{m,s}$ and variances $\sigma^2_{m,s}$ of the species corresponding to the model's outputs $m = 1, \ldots, n_M$ at times $0 \leq t_s \leq t_f$, $s = 1, \ldots, n_S$. Assuming the measures to be independently and normally distributed, the parameter identification problem is mathematically formulated as

$$\min_{\mathbf{p}} \sum_{m=1}^{n_M} \sum_{s=1}^{n_S} \left(\frac{y_m(t_s) - Y_{m,s}}{\sigma_{m,s}} \right)^2 \tag{2.2}$$

subject to:

$$\begin{aligned} \frac{d}{dt}\mathbf{x}(t) &= \mathbf{N}\mathbf{v}(\mathbf{x}(t), \mathbf{p}),\ \mathbf{x}(0) = \mathbf{x_0} \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \\ \mathbf{p_{min}} &\leq \mathbf{p} \leq \mathbf{p_{max}}, \end{aligned}$$

with $\mathbf{p_{min}}, \mathbf{p_{max}} \in \mathbb{R}^{n_P}$ lower and upper bounds on the parameter values.

To simplify presentation, we restricted ourselves to lower and upper parameter bounds in contrast to more general inequality and equality constraints (see e.g. [28]) that are rarely used–in our experience–for parameter optimization of biomolecular networks. The structural requirements on $\mathbf{C}$ imply that each experimentally measured species has to correspond to exactly one state of the model. In practice, proteins with different post-translational modifications, and similar, are often non-distinguishable experimentally [26]. Thus, some experimental data might represent the sum of several states of the model. In such cases, an adequate coordinate transformation of the state-space $\tilde{\mathbf{x}}(t) = \begin{pmatrix}\mathbf{C^T} & \mathbf{Q^T}\end{pmatrix}^T \mathbf{x}(t)$, with $\mathbf{Q} \in \mathbb{R}^{n_X - n_M \times n_X}$ chosen such that $(\mathbf{C}^T, \mathbf{Q}^T)$ has full rank, is typically sufficient to bring the model into the form imposed by the requirements on $\mathbf{C}$. Furthermore, we conveniently assumed the initial conditions $\mathbf{x_0}$ of the network to be given. If the initial conditions $\mathbf{x_0}$ are unknown, they can become decision variables of the optimization by appending them to the parameter vector $\bar{\mathbf{p}}^T = (\mathbf{p}^T, \mathbf{x_0}^T)$. Note that in this case, the initial conditions of the respective parameter sensitivities (Section 2.3) have to be adjusted (see e.g. [24]). Finally, time-dependent external inputs like dynamically changing environmental conditions can be incorporated by allowing for an additional dependence of the rates $\mathbf{v}$ on these external inputs.

2.2. Network Modularization. To identify the parameters of larger biomolecular networks, we modularize these networks (Eq. 2.1) into $m = 1, \ldots, n_M$ modules. Similar to our previous work [23], we require that each module contains exactly one output $y_m(t)$. Furthermore, each module's dynamics can depend on n_M trajectories $\mathbf{u}(t) = (u_1(t), \ldots, u_{n_M}(t))^T \in \mathbb{R}^{n_M}_{\geq 0}$, referred to as *inputs*. The modules are connected to one another by using the modules' outputs as each others inputs, i.e. by setting $\mathbf{u}(t) = \mathbf{y}(t)$. Upon interconnection, we require that the modules' dynamics resemble the dynamics of the whole (non-modularized) network (see below).

The dynamics of each module $m = 1, \ldots, n_M$ satisfy the ODEs

$$\begin{aligned} \frac{d}{dt}\mathbf{x_m}(t) &= \mathbf{N_m}\mathbf{v_m}(\mathbf{x_m}(t), \mathbf{p}, \mathbf{u}(t)),\ \mathbf{x_m}(0) = \mathbf{x_{m,0}} \\ y_m(t) &= \mathbf{c_m^T}\mathbf{x_m}(t), \end{aligned} \tag{2.3}$$

with $\mathbf{x_m}(t), \mathbf{x_{m,0}} \in \mathbb{R}^{n_{X,m}}_{\geq 0}$, $\mathbf{v_m}(\mathbf{x_m}(t), \mathbf{p}, \mathbf{u}(t)) \in \mathbb{R}^{n_{R,m}}$, $\mathbf{N_m} \in \mathbb{R}^{n_{X,m} \times n_{R,m}}$, and $\mathbf{c_m} \in \mathbb{R}^{n_{X,m}}$.

The relationship between module m and the whole network is established by identifying the module's species with the species of the whole network having indices $I_{X,m} \subseteq \{1, \ldots, n_X\}$, $|I_{X,m}| = n_{X,m}$. Similarly, the module's reactions are identified with the reactions of the whole network having indices $I_{R,m} \subseteq \{1, \ldots, n_R\}$, $|I_{R,m}| = n_{R,m}$. Given a valid choice for $I_{X,m}$ and $I_{R,m}$ (see below), this relationship is explicitly given by $\mathbf{x_m}(t) = \mathbf{P_m}\mathbf{x}(t)$, $\mathbf{x_{m,0}} = \mathbf{P_m}\mathbf{x_0}$, $\mathbf{N_m} = \mathbf{P_m}\mathbf{N}\mathbf{Q_m^T}$, and $\mathbf{C^T} = (\mathbf{P_1^T}\mathbf{c_1}, \ldots, \mathbf{P_{n_M}^T}\mathbf{c_{n_M}})$. The matrix $\mathbf{P_m} \in \{0,1\}^{n_{Xm} \times n_X}$ maps the states of the whole system (Eq. 2.1) to the states of the m^{th} module, with $(\mathbf{P_m})_{I_{X,m}}$ the identity matrix and $(\mathbf{P_m})_{I^C_{X,m}}$ the zero matrix. Here, $(.)^C$ denotes the set complement, i.e. $I^C_{X,m} = \{1, \ldots, n_X\} \setminus I_{X,m}$. Similarly, $\mathbf{Q_m} \in \{0,1\}^{n_{Rm} \times n_R}$ maps the reaction rates of the complete system to the rates of the module, with $(\mathbf{Q_m})_{I_{R,m}}$ the identity matrix and $(\mathbf{Q_m})_{I^C_{R,m}}$ the zero matrix. Finally, the reaction rates of module m are given by

$$\mathbf{v_m}\left(\mathbf{x_m}(t), \mathbf{p}, \mathbf{u}(t)\right) = \mathbf{Q_m}\mathbf{v}\left(\mathbf{P_m}^T\mathbf{x_m}(t) + \mathbf{C}^T(\mathbf{I} - \mathbf{C}\mathbf{P_m}^T\mathbf{P_m}\mathbf{C}^T)\mathbf{u}(t), \mathbf{p}\right),$$

with $\mathbf{I}$ the identity matrix. Intuitively, this corresponds to replacing all dependencies of the reaction rates in module m on species inside module m by the corresponding module's species ($\mathbf{P_m}^T\mathbf{x_m}(t)$), and all dependencies on interface species not being part of the module by the corresponding inputs ($\mathbf{C}^T(\mathbf{I}-\mathbf{C}\mathbf{P_m}^T\mathbf{P_m}\mathbf{C}^T)\mathbf{u}(t)$).

Upon interconnection of the modules, each module's reaction rates should equal the corresponding rates of the non-modularized network, i.e.

$$\mathbf{Q_m}\mathbf{v}(\mathbf{x}(t),\mathbf{p}) = \mathbf{v_m}(\mathbf{x_m}(t),\mathbf{p},\mathbf{u}(t)) \text{ for } \mathbf{u}(t)=\mathbf{y}(t).$$

Thus, $I_{X,m}$ and $I_{R,m}$ have to be chosen such that $\frac{\delta}{\delta x_i}\mathbf{Q_m}\mathbf{v}(\mathbf{x}(t),\mathbf{p}) = \mathbf{0}$ for all $i \notin I_{X,m}$ and $x_i(t)$ being not the output of any module, with $\frac{\delta}{\delta x_i}$ the partial derivative with respect to $x_i(t)$. Trivially, such a modularization is always possible by choosing $I_{X,m}=\{1,\ldots,n_X\}$ and $I_{R,m}=\{1,\ldots,n_R\}$. To find the minimal sets of species $\hat{I}_{X,m}$ and reactions $\hat{I}_{R,m}$ for each module, we apply our previous results (Lemma 10 in [23]) based on a so called species-reaction graph (see Figs. 2A and 5) that describes the directed information flow in a biomolecular network. In [23], we showed that

$$\text{(2.4a)} \qquad i \in \hat{I}_{X,m} \Rightarrow \left(\mathbf{C}\exp\left[\chi(\mathbf{N})\chi\left(\frac{\delta}{\delta\mathbf{x}}\mathbf{v}(\mathbf{x}(t),\mathbf{p})\right)\left(\mathbf{I}-\mathbf{C}^T\mathbf{C}\right)\right]\right)_{m,i} \neq 0$$

$$\text{(2.4b)} \qquad i \in \hat{I}_{R,m} \Rightarrow \left(\mathbf{C}\exp\left[\chi(\mathbf{N})\chi\left(\frac{\delta}{\delta\mathbf{x}}\mathbf{v}(\mathbf{x}(t),\mathbf{p})\right)\left(\mathbf{I}-\mathbf{C}^T\mathbf{C}\right)\right]\chi(\mathbf{N})\right)_{m,i} \neq 0$$

with exp the matrix exponent, and χ the element-wise indicator function for a matrix $\mathbf{F}$ defined by

$$(\chi(\mathbf{F}))_{kl} = \begin{cases} 0 & \text{if } \forall \mathbf{x}(t),\mathbf{u}(t)\in\mathbb{R}^{n_X}_{\geq 0}, \mathbf{p_{min}} \leq \mathbf{p} \leq \mathbf{p_{max}}:\ F_{kl}=0 \\ 1 & \text{otherwise.} \end{cases}$$

Eqs. 2.4a and 2.4b correspond to necessary conditions for observability of x_i, respectively v_i, by the output y_m, when removing all dependencies of the reaction rates on directly experimentally measured species. These conditions are in general not sufficient to decide if a certain species or reaction has to belong to a given module, for example, because of potential symmetries in the network. However, given our experience, such cases are rather rare for models of biomolecular networks, and for most models Eqs. 2.4a and 2.4b are necessary and sufficient. Thus, we apply these conditions to define the indices $I_{X,m}$ and $I_{R,m}$ of the species and reactions belonging to a given module m. Nevertheless, we recommend using more sophisticated methods to exclude (theoretical) non-observability (e.g. [32]) in addition.

In addition to $I_{X,m}$ and $I_{R,m}$, we denote by $I_{P,m} \subseteq \{1,\ldots,n_P\}$, $|I_{P,m}| = n_{P,m}$, and $I_{U,m} \subseteq \{1,\ldots,n_M\}$, $|I_{U,m}| = n_{U,m}$, the indices of the parameters and inputs of module m. Given $\mathbf{v_m}$, these indices can be determined by

$$\text{(2.5a)} \qquad i \in I_{P,m} \Leftrightarrow \chi\left(\frac{\delta}{\delta p_i}\mathbf{v_m}\left(\mathbf{x_m}(t),\mathbf{p},\mathbf{u}(t)\right)\right) \neq \mathbf{0}$$

$$\text{(2.5b)} \qquad i \in I_{U,m} \Leftrightarrow \chi\left(\frac{\delta}{\delta u_i}\mathbf{v_m}\left(\mathbf{x_m}(t),\mathbf{p},\mathbf{u}(t)\right)\right) \neq \mathbf{0}.$$

Eqs. 2.5a and 2.5a state that only parameters and inputs belong to a module which influence the rate of at least one of the reactions of the module. The definitions of $I_{P,m}$ and $I_{U,m}$ only become important for the determination of the computational complexity of our proposed algorithm (Section 2.7). Thus, to simplify notation, we formulated the ODEs of each module (Eq. 2.3) such that they depend on all parameters and inputs, and not only on the respective module's parameters and inputs.

Note that, different from other definitions of modules, the sets of species, reactions, and parameters of our modules might overlap. Even more importantly, not all species, reactions, or parameters necessarily belong to at least one module, for example, when certain species or reactions are not observable by any output. Parameters that only pertain to reactions outside any module are not identifiable. Thus, (some) non-identifiable parameters are automatically sorted out as a by-product of the modularization.

Given the modularization of the network, we can equivalently express Eq. 2.2 as an optimization problem with decision variables $\mathbf{p}$ and $\mathbf{u}(t)$:

$$\min_{\mathbf{p},\mathbf{u}} \sum_{m=1}^{n_M} \sum_{s=1}^{n_S} \left(\frac{y_m(t_s) - Y_{m,s}}{\sigma_{m,s}} \right)^2 \tag{2.6a}$$

subject to:

$$\begin{aligned} \frac{d}{dt}\mathbf{x_m}(t) &= \mathbf{N_m}\mathbf{v_m}(\mathbf{x_m}(t), \mathbf{p}, \mathbf{u}(t)),\ \mathbf{x_m}(0) = \mathbf{x_{m,0}} \\ y_m(t) &= \mathbf{c_m^T}\mathbf{x_m}(t) \\ \mathbf{p_{min}} &\leq \mathbf{p} \leq \mathbf{p_{max}} \\ \mathbf{0} &= \mathbf{u}(t) - \mathbf{y}(t) \qquad (2.6b) \end{aligned}$$

for all $m \in \{1, \dots, n_M\}$. Eq. 2.6b is conceptually similar to matching conditions [6,24] or continuity constraints [30] in multiple-shooting. It states that the trajectories of the inputs have to be equal to the trajectories of the outputs, which guarantees that the dynamics of the states of each module are equivalent to the corresponding states of the non-modularized model.

2.3. Linearized Modular Optimization Problem. To solve the modular optimization problem (Eq. 2.6), we apply a Gauss-Newton algorithm iteratively refining the parameters and inputs to find a (locally) optimal set of parameters and a corresponding set of inputs fulfilling the matching condition (Eq. 2.6b). In the following, we denote by a superscript the value of a variable in a given iteration $i = 0, 1, \dots$ of the Gauss-Newton algorithm. Starting from initial guesses for the parameter vector $\mathbf{p^0}$ and for the modules' input trajectories $\mathbf{u^0}(t)$, we determine in each iteration of the algorithm how the outputs of the individual modules change (i) in response to infinitesimal perturbations of the parameters while keeping the modules' inputs (i.e. the outputs of all other modules) fixed, and (ii) in response to infinitesimal perturbations of the inputs while keeping the parameters fixed. This approach closely resembles the discrimination between intra- and inter-level responses in MRA [10,18].

Keeping the modules' inputs fixed, the change of the output dynamics $\mathbf{dy_p^i}(t)$ in response to infinitesimal perturbations of the parameters $\mathbf{dp^i}$ is quantified by the *parameter sensitivities* $\mathbf{S_p^i}^T(t) = \left(\mathbf{S_{p,1}^i}^T(t), \dots, \mathbf{S_{p,n_M}^i}^T(t)\right)$, with $\mathbf{dy_p^i}(t) = \mathbf{S_p^i}(t)\mathbf{dp^i}$. The parameter sensitivities $\mathbf{S_{p,m}^i}(t)$ of each module $m = 1, \dots, n_M$ are obtained by integrating the initial value problems

$$\begin{aligned} \frac{d}{dt}\mathbf{S_{p,x,m}^i}(t) &= \mathbf{N_m}\frac{\delta}{\delta \mathbf{x_m^i}}\mathbf{v_m}(\mathbf{x_m^i}(t), \mathbf{p^i}, \mathbf{u^i}(t))\mathbf{S_{p,x,m}^i}(t) + \mathbf{N_m}\frac{\delta}{\delta \mathbf{p^i}}\mathbf{v_m}(\mathbf{x_m^i}(t), \mathbf{p^i}, \mathbf{u^i}(t)) \\ \mathbf{S_{p,m}^i}(t) &= \mathbf{c_m}^T\mathbf{S_{p,x,m}^i}(t), \end{aligned}$$

with $\mathbf{S_{p,x,m}}(0) = \mathbf{0}$.

Similarly, when keeping the parameters fixed, the change of the modules' output dynamics $\mathbf{dy^i_u}(t)$ in response to infinitesimal perturbations of the input trajectories $\mathbf{du^i}(t)$ is quantified by the *input sensitivities* $\mathbf{S^i_u}^T(t,s) = \left(\mathbf{S^i_{u,1}}^T(t,s), \ldots, \mathbf{S^i_{u,n_M}}^T(t,s)\right)$. These sensitivities $\mathbf{S^i_u}(t,s)$, with $\mathbf{dy^i_u}(t) = \int_0^t \mathbf{S^i_u}(t,s)\mathbf{du^i}(s)ds$, correspond to the impulse responses of time-varying linear systems [33] representing the individual modules linearized around the current parameters and inputs. In general, the impulse responses are computationally expensive to determine numerically. Therefore, we propose in the next section an alternative approach to estimate the responses of the modules' outputs to small changes in the input dynamics. Simultaneously perturbing the parameters and the inputs leads to the response in the output dynamics $\mathbf{dy^i}(t) = \mathbf{dy^i_p}(t) + \mathbf{dy^i_u}(t)$.

Given these definitions, the (locally) optimal parameter and input updates $\mathbf{\Delta p^i}$ and $\mathbf{\Delta u^i}(t)$ are the solutions of the linearized modular optimization problem

(2.7a)

$$\min_{\mathbf{\Delta p^i},\mathbf{\Delta u^i}} \sum_{m=1}^{n_M}\sum_{s=1}^{n_S}\left(\frac{y^i_m(t_s) - Y_{m,s} + \mathbf{S^i_{p,m}}(t_s)\mathbf{\Delta p^i} + \int_0^{t_s}\mathbf{S^i_{u,m}}(t_s,s)\mathbf{\Delta u^i}(s)ds}{\sigma_{m,s}}\right)^2$$

subject to:

$$\text{(2.7b)} \qquad \begin{aligned} \frac{d}{dt}\mathbf{x^i_m}(t) &= \mathbf{N_m v_m}(\mathbf{x^i_m}(t), \mathbf{p^i}, \mathbf{u^i}(t)),\ \mathbf{x^i_m}(0) = \mathbf{x_{m,0}}, \\ y^i_m(t) &= \mathbf{c_m}^T\mathbf{x^i_m}(t) \\ \mathbf{p_{min}} - \mathbf{p^i} &\le \mathbf{\Delta p^i} \le \mathbf{p_{max}} - \mathbf{p^i} \end{aligned}$$

$$\text{(2.7c)} \qquad \mathbf{u^i}(t) + \mathbf{\Delta u^i}(t) = \mathbf{y^i}(t) + \mathbf{S^i_p}(t)\mathbf{\Delta p^i} + \int_0^t \mathbf{S^i_u}(t,s)\mathbf{\Delta u}(s)ds.$$

Note that after the linearization of the system, it is not guaranteed anymore that the output trajectories equal the input trajectories. We denote the difference between these trajectories as the *inconsistency* $\mathbf{J^i}$, defined by

$$\mathbf{J^i}(t) = \mathbf{y^i}(t) - \mathbf{u^i}(t).$$

A non-zero inconsistency can not only result from inconsistently chosen initial parameter sets $\mathbf{p^0}$ and input trajectories $\mathbf{u^0}(t)$, but also from the non-linearity of the network: The sensitivities $\mathbf{S^i_{p,m}}(t)$ and $\mathbf{S^i_{u,m}}(t,s)$ quantify the change in the output dynamics for infinitesimal parameter and input perturbations $\mathbf{dp^i}$ and $\mathbf{du^i}(t)$, while the (locally) optimal updates $\mathbf{\Delta p^i}$ and $\mathbf{\Delta u^i}(t)$ are not infinitesimal. However, while we require $\mathbf{J^i}(t)$ to be (close to) zero after convergence of our algorithm, this condition does not have to be satisfied during intermediate iterations.

2.4. Relaxed Matching Condition and Input Trajectory Update. The linearized optimization problem (Eq. 2.7) is infinite-dimensional since the input trajectory update $\mathbf{\Delta u^i}(t)$ is a function of time, and depends on the impulse responses $\mathbf{S^i_u}(t,s)$ which are in general computationally expensive to determine. To formulate a finite-dimensional "condensed" version of the modular optimization problem in terms of variables that are numerically better accessible, we relax the matching condition (Eq. 2.7c) and restrict the input trajectory update to a specific functional form. These two simplifications are mathematically justified by proving local convergence of the final condensed optimization algorithm *a posteriori* (see Supplementary Information).

First, we define n_τ monotonously increasing *matching times* $0 \leq \tau_1 < \ldots < \tau_{n_\tau} = t_f$, and relax the matching condition (Eq. 2.7c) of the linearized system such that it only has to be fulfilled at these matching times. Then, we fix the input trajectories' update to

$$\boldsymbol{\Delta}\mathbf{u}^{\mathbf{i}}(t) = \mathbf{J}^{\mathbf{i}}(t) + \sum_{k=1}^{n_\tau} \boldsymbol{\Delta}\lambda^{\mathbf{i}}_{\mathbf{k}}\phi_k(t), \tag{2.8}$$

with n_M new parameters $\boldsymbol{\Delta}\lambda^{\mathbf{i}}_{\mathbf{k}} \in \mathbb{R}^{n_M}$ for each matching time τ_k, $k = 1, \ldots, n_\tau$. The *basis functions* $\phi_k(t)$ are defined by

$$\phi_k(t) = \begin{cases} \frac{t-\tau_{k-1}}{\tau_k-\tau_{k-1}} & \tau_{k-1} < t \leq \tau_k \\ \frac{\tau_{k+1}-t}{\tau_{k+1}-\tau_k} & \tau_k < t < \tau_{k+1} \\ 0 & \text{otherwise,} \end{cases} \tag{2.9}$$

with $\tau_0 = 0$ and $\tau_{n_\tau+1} = t_f$.

The relaxed matching condition can then be expressed for $j = 1, \ldots, n_\tau$ as

$$\boldsymbol{\Delta}\lambda^{\mathbf{i}}_{\mathbf{j}} = \mathbf{S}^{\mathbf{i}}_{\mathbf{p}}(\tau_j)\boldsymbol{\Delta}\mathbf{p}^{\mathbf{i}} + \mathbf{S}^{\mathbf{i}}_{\mathbf{J}}(\tau_j) + \sum_{k=1}^{n_\tau} \mathbf{S}^{\mathbf{i}}_{\lambda,\mathbf{k}}(\tau_j)\boldsymbol{\Delta}\lambda^{\mathbf{i}}_{\mathbf{k}}, \tag{2.10}$$

with $\mathbf{S}^{\mathbf{i}}_{\mathbf{J}}(t) = \int_0^t \mathbf{S}^{\mathbf{i}}_{\mathbf{u}}(t,s)\mathbf{J}^{\mathbf{i}}(s)ds \in \mathbb{R}^{n_M}$ and $\mathbf{S}^{\mathbf{i}}_{\lambda,\mathbf{k}}(t) = \int_0^t \mathbf{S}^{\mathbf{i}}_{\mathbf{u}}(t,s)\phi_k(s)ds \in \mathbb{R}^{n_M \times n_M}$ the local sensitivities of the outputs with respect to infinitesimal changes of the input trajectories in the directions of $\mathbf{J}^{\mathbf{i}}(t)$ and $\phi_k(t)$. Similar to the parameter sensitivities, we obtain the sensitivities $S^i_{J,m}(t)$, ${\mathbf{S}^{\mathbf{i}}_{\mathbf{J}}}^T(t) = \left(S^i_{J,1}(t), \ldots, S^i_{J,n_M}(t)\right)$, independently for each module m by solving the initial value problems

$$\begin{aligned} \frac{d}{dt}\mathbf{S}^{\mathbf{i}}_{\mathbf{J,x,m}}(t) =& \mathbf{N_m}\frac{\delta}{\delta \mathbf{x}^{\mathbf{i}}_{\mathbf{m}}}\mathbf{v_m}(\mathbf{x}^{\mathbf{i}}_{\mathbf{m}}(t), \mathbf{p}^{\mathbf{i}}, \mathbf{u}^{\mathbf{i}}(t))\mathbf{S}^{\mathbf{i}}_{\mathbf{J,x,m}}(t) \\ &+ \mathbf{N_m}\frac{\delta}{\delta \mathbf{u}^{\mathbf{i}}}\mathbf{v_m}(\mathbf{x}^{\mathbf{i}}_{\mathbf{m}}(t), \mathbf{p}^{\mathbf{i}}, \mathbf{u}^{\mathbf{i}}(t))\mathbf{J}^{\mathbf{i}}(t) \\ S^i_{J,m}(t) =& {\mathbf{c_m}}^T\mathbf{S}^{\mathbf{i}}_{\mathbf{J,x,m}}(t), \end{aligned}$$

with $\mathbf{S}^{\mathbf{i}}_{\mathbf{J,x,m}}(0) = \mathbf{0}$. The sensitivities ${\mathbf{S}^{\mathbf{i}}_{\lambda}}^T(t) = ({\mathbf{S}^{\mathbf{i}}_{\lambda,\mathbf{1}}}^T(t), \ldots, {\mathbf{S}^{\mathbf{i}}_{\lambda,\mathbf{n_M}}}^T(t))$ are obtained by solving analogous initial value problems for each input and each basis function.

The relaxed matching condition (Eq. 2.10) corresponds to a set of $n_\tau n_M$ equations that can be solved for $\boldsymbol{\Delta}\lambda^{\mathbf{i}}$:

$$\boldsymbol{\Delta}\lambda^{\mathbf{i}} = (\mathbf{I} - \mathbf{S}^{\mathbf{i}}_{\lambda\tau})^{-1}\left(\mathbf{S}^{\mathbf{i}}_{\mathbf{p}\tau}\boldsymbol{\Delta}\mathbf{p}^{\mathbf{i}} + \mathbf{S}^{\mathbf{i}}_{\mathbf{J}\tau}\right), \tag{2.11}$$

with

$$\begin{aligned} {\boldsymbol{\Delta}\lambda^{\mathbf{i}}}^T &= \left({\boldsymbol{\Delta}\lambda^{\mathbf{i}}_{\mathbf{1}}}^T, \quad \ldots, \quad {\boldsymbol{\Delta}\lambda^{\mathbf{i}}_{\mathbf{n}_\tau}}^T\right), & {\mathbf{S}^{\mathbf{i}}_{\mathbf{p}\tau}}^T &= \left({\mathbf{S}^{\mathbf{i}}_{\mathbf{p}}}^T(\tau_1), \quad \ldots, \quad {\mathbf{S}^{\mathbf{i}}_{\mathbf{p}}}^T(\tau_{n_\tau})\right), \\ {\mathbf{S}^{\mathbf{i}}_{\mathbf{J}\tau}}^T &= \left({\mathbf{S}^{\mathbf{i}}_{\mathbf{J}}}^T(\tau_1), \quad \ldots, \quad {\mathbf{S}^{\mathbf{i}}_{\mathbf{J}}}^T(\tau_{n_\tau})\right), & {\mathbf{S}^{\mathbf{i}}_{\lambda\tau}}^T &= \left({\mathbf{S}^{\mathbf{i}}_{\lambda}}^T(\tau_1), \quad \ldots, \quad {\mathbf{S}^{\mathbf{i}}_{\lambda}}^T(\tau_{n_\tau})\right). \end{aligned}$$

The matrix $(\mathbf{I} - \mathbf{S}^{\mathbf{i}}_{\lambda\tau})$ is conceptually similar to the negative of the so called interaction map in MRA [10]. It can only be inverted if it has full rank, i.e. $\mathbf{S}^{\mathbf{i}}_{\lambda\tau}$ must not have an eigenvalue equal to one. To illustrate this condition, consider $\mathbf{S}^{\mathbf{i}}_{\lambda\tau}$ has an eigenvalue equal to one with corresponding eigenvector $\boldsymbol{\Delta}\lambda^{\mathbf{i}}_{\mathbf{1}}$. Then, if $\boldsymbol{\Delta}\lambda^{\mathbf{i}}_{\mathbf{0}}$ is a solution of Eq. 2.11, also $\boldsymbol{\Delta}\lambda^{\mathbf{i}}_{\mathbf{0}} + \alpha\boldsymbol{\Delta}\lambda^{\mathbf{i}}_{\mathbf{1}}$ is a solution for every $\alpha \in \mathbb{R}$. Such a $\boldsymbol{\Delta}\lambda^{\mathbf{i}}_{\mathbf{0}}$ corresponds to a perturbation of the inputs which results in a change of the output trajectories being equal to the input perturbation at the matching times. Even though the individual modules are causal and do not contain feed-throughs, such an effect is in principle possible if the distance between adjacent matching times is too large. To prevent such cases, we propose to monitor the condition number of $(\mathbf{I} - \mathbf{S}^{\mathbf{i}}_{\lambda\tau})$, and eventually to increase the number of matching times if necessary. In the following, we assume that the distance between the matching times was chosen small enough such that $(\mathbf{I} - \mathbf{S}^{\mathbf{i}}_{\lambda\tau})$ is well conditioned.

2.5. Condensed Modular Parameter Identification Problem. As described in the previous section, by relaxing the matching condition and restricting the input trajectory update $\boldsymbol{\Delta}\mathbf{u}^{\mathbf{i}}(t)$ to a specific functional form, $\boldsymbol{\Delta}\mathbf{u}^{\mathbf{i}}(t)$ becomes a function of the parameter vector update $\boldsymbol{\Delta}\mathbf{p}^{\mathbf{i}}$ (Eqs. 2.8 and 2.11). This also implies that $\boldsymbol{\Delta}\mathbf{u}^{\mathbf{i}}(t)$ satisfies the relaxed matching condition (Eq. 2.10).

With these simplifications, we can state a *condensed* modular parameter identification problem having as decision variables only $\boldsymbol{\Delta}\mathbf{p}^{\mathbf{i}}$:

$$\min_{\boldsymbol{\Delta}\mathbf{p}^{\mathbf{i}}} \sum_{m=1}^{n_M} \sum_{s=1}^{n_S} \left(\frac{y^i_m(t_s) - Y_{m,s} + \mathbf{T}^{\mathbf{i}}_{\mathbf{p},\mathbf{m}}(t_s)\boldsymbol{\Delta}\mathbf{p}^{\mathbf{i}} + \mathbf{T}^{\mathbf{i}}_{\mathbf{J},\mathbf{m}}(t_s)}{\sigma_{m,s}} \right)^2 \tag{2.13a}$$

subject to

$$\begin{aligned} \frac{d}{dt}\mathbf{x}^{\mathbf{i}}_{\mathbf{m}}(t) &= \mathbf{N}_{\mathbf{m}}\mathbf{v}_{\mathbf{m}}(\mathbf{x}^{\mathbf{i}}_{\mathbf{m}}(t), \mathbf{p}^{\mathbf{i}}, \mathbf{u}^{\mathbf{i}}(t)),\ \mathbf{x}^{\mathbf{i}}_{\mathbf{m}}(0) = \mathbf{x}_{\mathbf{m},\mathbf{0}}, \\ y^i_m(t) &= \mathbf{c}_{\mathbf{m}}{}^T\mathbf{x}^{\mathbf{i}}_{\mathbf{m}}(t) \\ \mathbf{p}_{\mathbf{min}} - \mathbf{p}^{\mathbf{i}} \leq \boldsymbol{\Delta}&\mathbf{p}^{\mathbf{i}} \leq \mathbf{p}_{\mathbf{max}} - \mathbf{p}^{\mathbf{i}}, \end{aligned} \tag{2.13b}$$

with $\mathbf{T}^{\mathbf{i}}_{\mathbf{p},\mathbf{m}}$ and $\mathbf{T}^{\mathbf{i}}_{\mathbf{J},\mathbf{m}}$ defined by

$$\mathbf{T}^{\mathbf{i}}_{\mathbf{p},\mathbf{m}}(t) = \mathbf{S}^{\mathbf{i}}_{\mathbf{p},\mathbf{m}}(t) + \mathbf{S}^{\mathbf{i}}_{\lambda,\mathbf{m}}(t) \left(\mathbf{I} - \mathbf{S}^{\mathbf{i}}_{\lambda\tau}\right)^{-1} \mathbf{S}^{\mathbf{i}}_{\mathbf{p}\tau} \tag{2.14a}$$

$$\mathbf{T}^{\mathbf{i}}_{\mathbf{J},\mathbf{m}}(t) = \mathbf{S}^{\mathbf{i}}_{\mathbf{J},\mathbf{m}}(t) + \mathbf{S}^{\mathbf{i}}_{\lambda,\mathbf{m}}(t) \left(\mathbf{I} - \mathbf{S}^{\mathbf{i}}_{\lambda\tau}\right)^{-1} \mathbf{S}^{\mathbf{i}}_{\mathbf{J}\tau}. \tag{2.14b}$$

Several efficient algorithms exist (see e.g. [11]) to determine the parameter update $\boldsymbol{\Delta}\mathbf{p}^{\mathbf{i}}$ by solving Eq. 2.13a. Given the parameter update, the input trajectory update $\boldsymbol{\Delta}\mathbf{u}^{\mathbf{i}}(t)$ can be derived using Eqs. 2.8 and 2.11.

Because we relax the matching condition and restrict the input trajectory update to a specific functional form, the condensed identification problem (Eq. 2.13) is not equivalent to the linearized one (Eq. 2.7). To mathematically justify these simplifications, we derived an upper bound $\Delta\tau_{max}$ for the distance between adjacent matching times such that (local) convergence is guaranteed given (local) identifiability of the parameters (see Supplementary Information). This upper bound $\Delta\tau_{max}$ scales with the square root of the reciprocal of the absolute value of the second derivative of the sensitivities of the outputs with respect to changes in the inputs and the parameters. This indicates that the matching times should be distributed over the simulation time according to the dynamics of the interface species. As a rule of thumb, the faster the dynamics of the interface species are during a given period, the denser should be the matching times in this period. As demonstrated in the Results section, for many optimization problems it seems sufficient to estimate the timescales of the interface species' dynamics based on the experimental data, and to distribute the matching times accordingly.

Finally, far away from a local optimum, additional provisions have to be made to prevent divergence of the algorithm (step-width adjustment) or negative input trajectories (basis function adjustment). Details about these additional steps, together with the holding condition for our algorithm, are described in the Supplementary Information.

2.6. Embedded Modular Algorithm. Since all interface species are by definition experimentally measured, typically rather good initial estimates $\mathbf{u^0}(t)$ for the input trajectories can be derived directly from the experimental data using adequate spline approximations (see Results section). In contrast, parameters (and, thus, initial parameter estimates $\mathbf{p^0}$) in biomolecular research are often only poorly constrained by prior knowledge. For such poorly constrained initial parameter guesses, the initial input trajectories $\mathbf{u^0}(t)$ are typically closer to the real network dynamics than the output trajectories $\mathbf{y^i}(t)$ during early iterations i; the latter might not even have the same order of magnitude as the experimental data. Given that the parameters converge to a good (local) optimum for which the output dynamics are in sufficient agreement with the experimental data, this also implies that $\mathbf{u^0}(t)$ is typically also closer to the final (optimized) output trajectories than the output trajectories $\mathbf{y^i}(t)$ at early iterations i. Thus, in the joint space of parameters and input trajectories, updating the input trajectories at too early iterations to reduce the inconsistencies $\mathbf{J^i(t)}$ more likely increases than decreases the distance to a good (local) optimum.

Given these considerations, we embed our modular parameter identification approach into a broader algorithm with two phases. In the first phase, we use a simplified version of our algorithm without updating the input trajectories, with $\mathbf{T^i_{Jt}}(t) = \mathbf{0}$ and $\mathbf{T^i_{pt}}(t) = \mathbf{S^i_{pt}}(t)$ in Eq. 2.13a, corresponding to optimizing each module independently. In this phase, only the parameter sensitivities $\mathbf{S^i_p}(t)$, but not the input sensitivities $\mathbf{S^i_\lambda}(t)$ and $\mathbf{S^i_J}(t)$ have to be numerically integrated, thus also reducing computational time per iteration. Furthermore, the separate optimization of the modules in the first phase also allows for efficient parallelization. We switch to the second phase, in which the full (condensed) modular algorithm is applied, either if the output trajectories are close to the experimental data (i.e., if the cost decreases below some threshold), or if the simplified algorithm converged.

2.7. Computational Complexity. Assuming a constant integration step width, equally distributed matching times, and equal cost to integrate a state or a sensitivity equation, the time complexity per iteration of our full modular parameter identification approach is $O(E_M)$, with

$$E_M = \sum_{m=1}^{n_M} \left[(n_{P,m} + 1) n_{X,m} + \chi(n_{U,m}) n_{X,m} + \frac{n_\tau + 1}{2} n_{U,m} n_{X,m} \right] \tag{2.15}$$

the number of state and sensitivity equations which have to be numerically integrated. The first term in the sum in Eq. 2.15 represents the cost to integrate the states $\mathbf{x_m}(t)$ and the parameter sensitivities $\mathbf{S^i_{p,m}}(t)$ of module m; sensitivities with respect to parameters not belonging to a module are zero and do not have to be integrated. The second term represents the cost to integrate the sensitivities with respect to the inconsistencies $\mathbf{J}^i(t)$; they only have to be integrated if the module has at least one input ($\chi(n_{U,m}) = 0$ if $n_{U,m} = 0$, and $\chi(n_{U,m}) = 1$ otherwise). The third term captures the cost to integrate the sensitivities with respect to each of the module's inputs and each basis function $\phi_k(t)$, $k = 1, \dots, n_\tau$; not all of these sensitivities have to be integrated over the whole time domain. Instead, the integrations can start when the respective basis functions are non-zero for the first time, i.e. $\sum_{k=1}^{n_\tau} \frac{k}{n_\tau} = \frac{n_\tau+1}{2}$.

Under the same assumptions, the time complexity per iteration of the corresponding non-modular parameter identification approach is $O(E_O)$, with

$$E_O = n_X(n_P + 1). \tag{2.16}$$

If we additionally assume that the states and parameters are distributed equally between non-overlapping modules sharing no common parameters, our full modular parameter identification approach is computationally less expensive ($E_M < E_O$) per iteration if $\Omega(n_\tau + 3) < 2\frac{n_P}{n_M}$. The mean relative connectivity $\Omega = \frac{\sum_{m=1}^{n_M} n_{U,m}}{n_M(n_M-1)} \in [0, 1]$ between the modules is $\Omega = 1$ ($\Omega = 0$) if the interface species of all (no) modules serve as inputs to one another. Thus, our approach seems especially suited for optimization problems with many parameters that are roughly equally distributed over similar sized modules, for modules with low inter-module connectivity, or a combination thereof. Note, however, that we compared the cost of our full modularized and a corresponding non-modularized algorithm *per iteration*. As described in the previous section, we apply a simplified algorithm in the first phase of our embedded approach without updating the input trajectories. For optimization problems with little or no overlap between modules, the number $\sum_{m=1}^{n_M} n_{X,m}(n_{P,m} + 1)$ of necessary integrations per iteration of the simplified algorithm is typically much lower than the corresponding number of integrations in a non-modular approach.

Ideally, the decision between a modular or a non-modular algorithm should rather depend on the number of integrations *per optimization run* than *per iteration*, as well as on the probability to converge to good (local) optima. For general biomolecular networks, both measures cannot be easily estimated analytically and have to be determined numerically.

3. Results.

3.1. Example 1: Repressilator. As a first example, we apply our embedded modular parameter identification approach to a model of a synthetic oscillatory network, the so-called *Repressilator* [13].

The model (available at the BioModels Database, [25]) consists of the three transcriptional repressors TetR, cI, and LacI, with protein concentrations $y_{1\dots3}$ and their corresponding mRNA species with concentrations $x_{1\dots3}$ (Fig. 2A). The dynamics of the *Repressilator* are described by the following set of ODEs:

$$\frac{d}{dt}x_i(t) = a_{0,tr} + a_{tr}\frac{K_M^n}{K_M^n + y_{i-1}^n(t)} - k_{dm}x_i(t) \tag{3.1a}$$

$$\frac{d}{dt}y_i(t) = k_{tl}x_i(t) - k_{dp}y_i(t) \tag{3.1b}$$

with $i \in \{1, 2, 3\}$ and $y_0 = y_3$. The nominal parameters of the model are $K_M = 40$, $a_{0,tr} = 0.03$, $a_{tr} = 29.97$, $k_{tl} = 6.93$, $k_{dm} = 0.347$, $k_{dp} = 0.0693$, and $n = 2$. After the decay of the initial conditions, the model with the nominal parameters shows stable limit-cycle oscillations.

In the following, we assume these nominal parameters to be unknown, and that they should be identified given experimental data. We generated artificial experimental data by assuming that all three protein concentrations $y_{1\dots3}$ can be experimentally

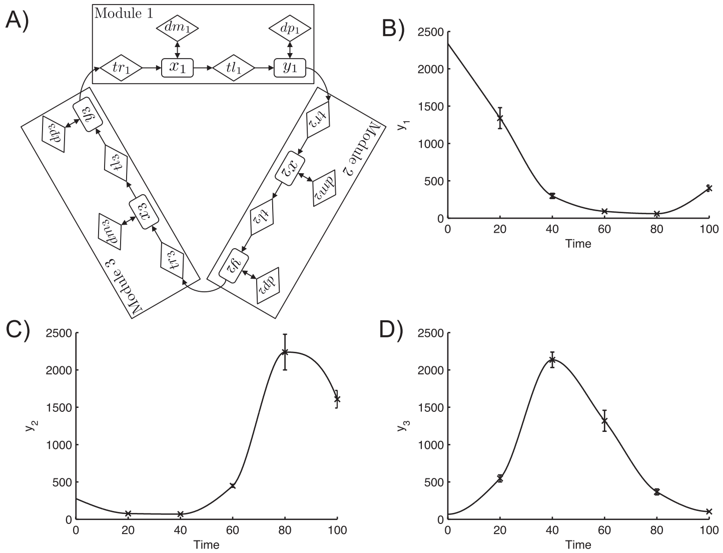

FIG. 2. *Network structure, artificial experimental data, and initial input trajectories for the Repressilator example [13]. A) Species-reaction graph. Species are represented by rectangular nodes, and reactions by diamonds. Reactions denoted by tr represent transcription, tl translation, dm mRNA degradation, and dp protein degradation. A directed edge is drawn from a species to a reaction if the reaction rate depends on the species' concentration, and an edge from a reaction to a species if firing of the reaction changes the species' concentration. B-D) Error bars represent the mean and standard deviations for the artificially generated experimental data of B) TetR, C) cI and D) LacI concentrations over time. The data was generated from the model with nominal parameters as described in the text. The curves represent splines (PCHIP, see [14]) fitted through the data and used as initial input trajectories for our modular parameter identification algorithm.*

measured. For each protein species, artificial mean concentrations and variances at five time points at a temporal resolution of $\Delta t = 20$ were generated. First, we determined the corresponding state values of the model with nominal parameters by numerical integration. Then, we created three artificial "replicas" for each time point by assuming normally distributed measurement noise with a coefficient of variation of $c_v = \frac{\sigma}{\mu} = 10\%$ around the simulated species concentrations. Similar to real measurement data, we finally calculated the mean and the unbiased variance of the species concentrations from these replicas.

We modularized the *Repressilator* as described in Section 2.2 (see Fig. 2A). Each of the three modules has the concentrations of one protein and its corresponding mRNA species as states, the protein species as an output, and the output (the protein species) of its preceeding module as an input. For the modular optimization, the initial input trajectories $\mathbf{u^0}(t)$ were set to piecewise cubic hermite interpolating polynomials (PCHIP, see [14]) of the artificial experimental data, using the PCHIP implementation part of the curve fitting toolbox of Matlab (The MathWorks, Natick, MA). In the first phase of our embedded modular algorithm, we optimized the parameters of the modules without updating the input trajectories as described in the previous section. As soon as the root-mean-square (RMS) error dropped below $F_{init} = 2$, or when the simplified algorithm converged, we switched to the full optimization algorithm. In the full optimization algorithm, $n_\tau = 10$ matching times were equally distributed at $\tau_1 = 10, \ldots, \tau_{10} = 100$.

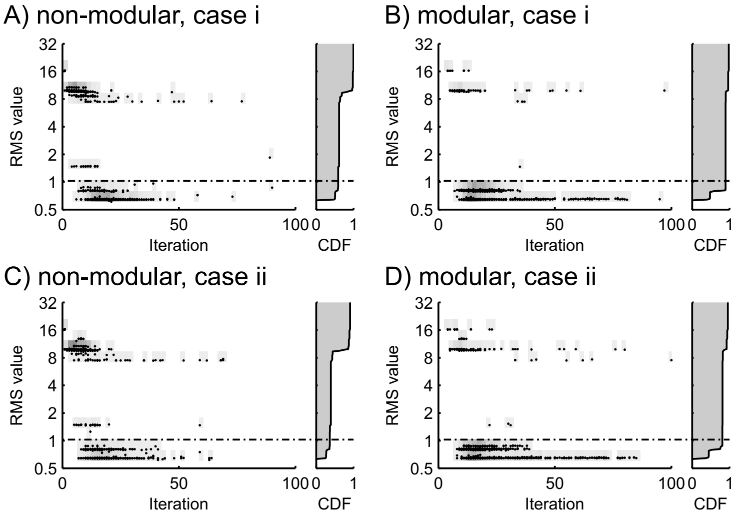

FIG. 3. *Probability density and cumulative distribution function (CDF) of the RMS value and the number of model evaluations for optimizing the Repressilator model of [13]. In total,* 1000 *different initial parameter sets were sampled with a Monte-Carlo approach, with K_M, $a_{0,tr}$, a_{tr}, k_{dm}, and k_{dp} log-uniformly distributed between* 0.01 *and* 100*-fold their original value, and the Hill factor n uniformly distributed between* 1 *and* 4*. The plots represent the results for the non-modular (left column) and our modular optimization algorithm (right column), either with (case i, top row) or without (case ii, bottom row) additionally constrained parameters. Each black dot shows the outcome of one optimization run, the background gray level the relative density (arbitrarily scaled), and the black dotted line the RMS error of the original parameter set with which the artificial experimental data was generated. Only optimization runs that converged are displayed.*

In each iteration, we determined the parameter update by solving a constrained linear least-squares problem consisting of the respective condensed modular parameter identification problem with an additional constraint on the parameter update

$$\left(\frac{1}{\pi_{max}} - 1\right)\mathbf{p^i} \leq \mathbf{\Delta p^i} \leq (\pi_{max} - 1)\,\mathbf{p^i}, \tag{3.2}$$

with $\pi_{max} = 2$ the maximal fold change of each parameter value per iteration (that is, in iteration $i+1$, each parameter can have a value maximally twice, and minimally half its value in iteration i). The constrained linear least-squares problem (Eqs. 2.13 and 3.2) was solved with an implementation of a reflective Newton method [11], part of the optimization toolbox of Matlab. The parameters of the backtracking algorithm determining the step-with (see Supplementary Information) were set to $\alpha = 0.1$ and $\beta = 0.5$, and the parameters for the holding condition (see Supplementary Information) to $A_1 = A_2 = A_3 = B_1 = B_2 = 1\%$.

To evaluate the convergence properties of our modular parameter identification algorithm, we compared its performance with a non-modular (local) algorithm operating on the whole network. For a comparison unbiased by different implementation details, we modified our modular algorithm to perform a non-modular parameter identification by allowing it to operate on a single "module" containing all states. Besides minor adjustments (e.g., removing unrequired convergence criteria for inconsistency), both algorithms were configured identically. We then used both algorithms in the local phase of a global stochastic multistart optimization [29] (see caption of Fig. 3 for details). During the optimization, we (i) either additionally constrained the parameters to stay in the region of the parameter space from which we sampled, or (ii) we did not impose any additional constraints. For both algorithms, the same randomly generated initial parameter sets were used. The final cost for a parameter set identified by our modular approach was determined *a posteriori* by simulation of the original, non-modularized network. We considered an optimization run to not have converged either after a maximum of 100 iterations, or when during an iteration the parameters were such that numerical integration failed for one or more of the modules (we used the solver $ode15s$ of Matlab with a maximal integration step width of 0.1).

With additionally constrained parameters (case i), both the non-modular (99% of the initial parameter sets) and our modular (98%) approach lead to convergence within 100 iterations (Fig. 3). For the non-modular approach, converged optimizations had an average RMS error of 4.17 and only 59% of the initial parameter sets converged to an RMS value of one or less; the nominal parameter set–used to generate the artificial experimental data–results in an RMS error of 1.03. In contrast, our modular parameter estimation method resulted in an average RMS error of 1.60, corresponding to 90% convergence to RMS values of less or equal to one. Both algorithms showed worse performance when parameters were not constrained to the sampling region (case ii). However, our modular approach (96% convergence, average RMS of 2.24, 81.1% final RMS below or equal to one) was less sensitive to the removal of the constraints than the non-modular approach (90.8% convergence, 5.91 average RMS, 36.7% with RMS less or equal to one). We also observed higher robustness of the modular approach when we use extended regions in parameter space for initial sampling (with constraints, case i; three orders of magnitudes around the nominal parameter set and $[1, 5]$ for the Hill factor n), with 36% convergence rate to RMS values of less or equal to one of the non-modular algorithm, and 70% for the modular algorithm.

Fig. 3 shows that the lower average RMS value of our modular algorithm is mainly achieved by avoiding certain sub-optimal local minima. Manual inspection showed that most of these sub-optimal parameter sets correspond to strongly damped oscillations or a non-oscillatory model behavior, and that a minority correspond to oscillations with a significantly higher frequency than oscillations obtained with the nominal parameter set. We hypothesize that the comparatively good performance of our modular algorithm results from the modules being "entrained" by the experimental data used to initialize the inputs to the modules. By allowing for inconsistencies between the modules' dynamics during (but not at the end of) the optimization, each module is exposed to input dynamics close to the dynamics of the "real" biomolecular network already in early iterations. In contrast, in a non-modular optimization, a module is only exposed to dynamics close to the real-world system after the parameters converged to the proximity of a good local optimum–if such a convergence happens at all.

3.2. Example 2: JAK2/STAT5 Signaling. As a second example, we estimated the parameters of a model [1] of the JAK2/STAT5 signaling pathway (available at the BioModels Database, [25]) that has 26 states and 36 reactions. The model describes the fast erythopoetin (Epo) induced signal transduction pathway in mammalian cells. Pathway activation results in phosphorylation and nuclear import of STAT5, as well as two slow feedback mechanisms via the production of SOCS3 and CIS under transcriptional control of nuclear localized STAT5. A wide range of different experiments were performed in [1] for parameter identification and model val-

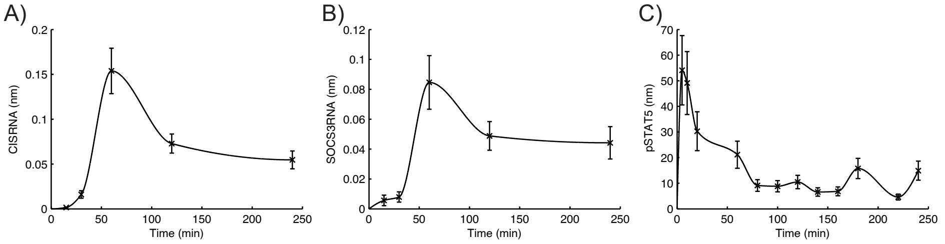

Fig. 4. *Experimental data for the JAK2/STAT4 signaling pathway used for optimization, and initial input trajectories. The error bars represent the mean and standard deviations for A) CIS RNA, B) SOCS3 RNA and C) phosphorylated STAT5 concentrations over time after induction with* 5 *U/ml Epo; they were derived from the experimental data given in [1] as described in the main text. The curves represent splines (PCHIP, see [14]) fitted through the data and used as initial inputs for our modular parameter optimization algorithm.*

idation. However, even with this extensive data set, not all of the 21 free parameters of the model are practically identifiable [1], which is a common–if not ubiquitous [26]–phenomenon for larger models in systems biology. Since in [1] the concentrations of several different species were simultaneously measured for several experiments, the JAK2/STAT5 signaling pathway model represents an interesting test case for our modular parameter identification approach.

This section is intended to demonstrate our modular identification approach on a realistic biomolecular network, and not to serve as an exhaustive biological study. Therefore, we restricted ourselves to a subset of the experimental data reported in [1] for the modular identification of the network. This subset consisted of data for the phosphorylated STAT5 protein (Table S3 in Supplementary Information of [1]), and for the SOCS3 and CIS mRNA abundances (Table S5) in the first 240 min after induction of the pathway with 5 U/ml Epo (Fig. 4). We transformed the raw experimental data to concentration units by using the relationships stated in Supplementary Information of [1] (Eqs. 96 and 123-128). For the mRNA levels, we calculated the mean and the standard deviations from the three available replicas. For phosphorylated STAT5, only one replica per time point was reported in [1]. We used these values as the mean and assumed a coefficient of variation of 25% in agreement with the average coefficient of variation of the mRNA data. With these experimentally measured species, we modularized the JAK2/STAT5 signaling model (see Fig. 5) as described in the Methods section.

All relevant 21 free parameters of the model for the chosen subset of the experimental data (those parameters that were neither fixed to a certain value already in [1] or correspond to initial species' concentrations) were optimized with an initial value set to 1. The units of the parameters and their upper and lower bounds were set according to Table S16 in [1]. This corresponds to initial parameter values up to more than eight orders of magnitude away from the ones identified in [1].

We distributed $n_\tau = 10$ matching times at $\{5, 10, 15, 20, 30, 60, 90, 120, 180, 240\}$ minutes after induction according to a rough visual inspection of the experimental data with respect to the time intervals showing fast interface species' dynamics (see Fig. 4). The switching condition between the two phases of our modular algorithm, the holding criterion as well as the maximal relative parameter update per iteration were set as in the previous example. To evaluate the performance of our algorithm, we also optimized the parameters using the non-modular counterpart of our algorithm as described above.

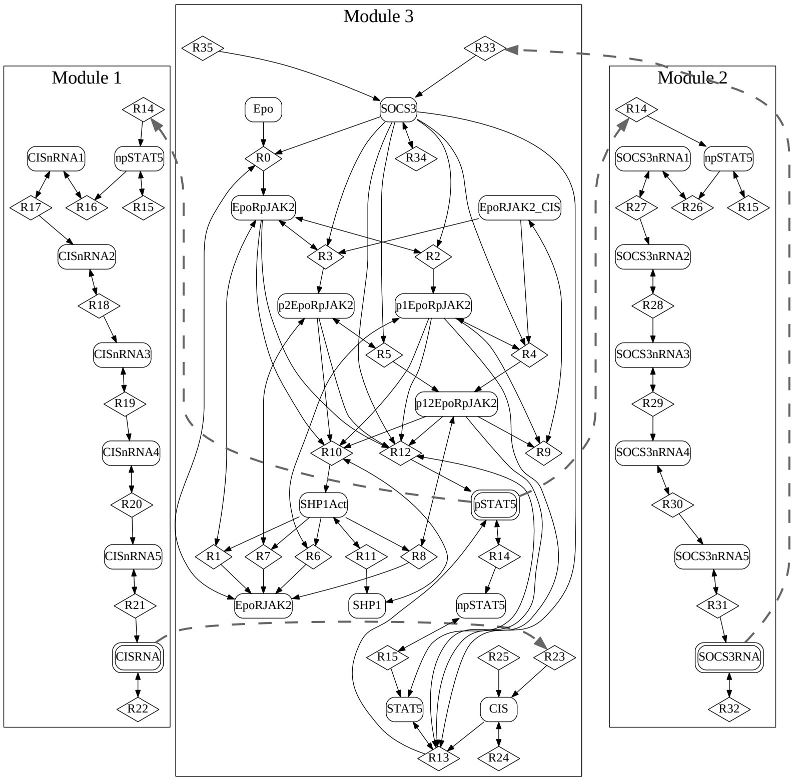

FIG. 5. *Species-reaction graph of the modularized JAK2/STAT5 signaling pathway [1]. The nodes (see Fig. 2A for notation) are grouped according to the module the corresponding reaction or species belongs to. Species and reactions belonging to more than one module (R14, R15 and npSTAT5) are represented by separate nodes in each module. Experimentally measured species are highlighted by a double border, and edges representing inter-module communication are dashed and slightly thicker. Graph drawn with Graphviz [12].*

For both the modular and the non-modular algorithm, Fig. 6 displays the RMS error between the experimental data and the corresponding states of the model, as well as the indicator functions used for the holding criteria. Our modular algorithm showed quick convergence and it terminated already after 21 iterations with an RMS value of 1.31 (determined *a posteriori* by integrating the non-modularized model). In contrast, the non-modular algorithm already terminated after the first iteration since the expected relative gain in the RMS value was less than the corresponding holding criterion (see Supplementary Information). To allow for a better comparison of the two algorithms, we removed the holding criterion for the non-modular algorithm, and ran it for a total of 100 iterations. During these 100 iterations, the non-modular algorithm temporarily signaled convergence two additional times at iteration 6 and $22-23$. Finally, it converged around iteration 50 to a local sub-optimal minimum with a final RMS value of 1.57.

The comparatively fast convergence properties of our embedded modular parameter optimization algorithm can be explained as a result of using the experimental data (Fig. 4) as proxies for the input dynamics during the initial phase of our embedded optimization algorithm (compare also to Example 1). It does not seem to be necessary that these inputs are very close to the final trajectories. Indeed, especially

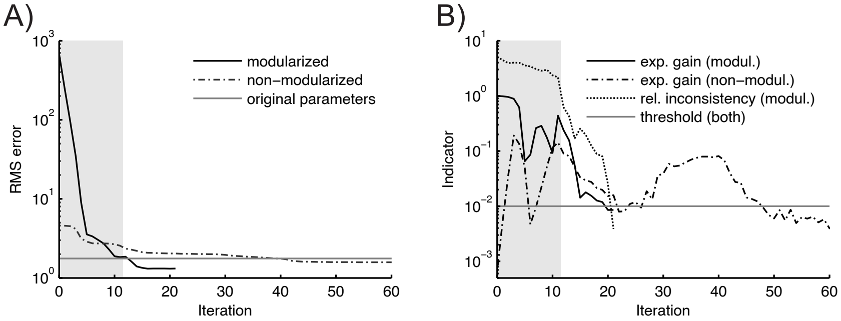

Fig. 6. *Comparison of the convergence properties of the modular and non-modular parameter identification algorithms for the JAK2/STAT5 signaling model of [1]. A) Root-mean-square error F between experimental data and simulations over iteration number. B) Expected relative RMS gain* $1-\frac{F(\mathbf{p^i},\mathbf{u^i},\mathbf{\Delta p^i},1)}{F(\mathbf{p^i},\mathbf{u^i},\mathbf{0},1)}$ *(modular, see Supplementary Information), respectively* $1-\frac{F(\mathbf{p^i},\mathbf{\Delta p^i},1)}{F(\mathbf{p^i},\mathbf{0},1)}$ *(non-modular), of updating the parameters; and the scaled maximal inconsistency* $\sqrt{\sum_{m=1}^{n_M} \frac{\max_t |J_m^i(t)|}{\mathrm{mean}_t(u_m^i+y_m^i)}}$ *(modular only) used as holding conditions (see Supplementary Information). The gray areas correspond to the iterations in which the simplified modular optimization algorithm was used.*

the initial input trajectory of phosphorylated STAT5 (Fig. 4C) could be improved with respect to prior knowledge of the expected dynamics. This suggests that it is sufficient that at least some dynamical properties are roughly captured and that concentrations have the correct order of magnitude. These inputs likely serve in the initial phase of the modular algorithm to "entrain" the individual modules, resulting in a quick convergence to dynamics similar to the final, (locally) optimal ones. Our full modular algorithm used in the second phase of the embedded algorithm, which started at iteration 11, first reduced the inconsistency between the inputs and the outputs while simultaneously adjusting the parameters to preserve the quality of the fits (ca. iterations $12-14$); subsequently, it even decreased the cost (iterations $15-20$). Since at the end of iteration 20, the convergence criterion for the parameters but not yet the one for the inconsistencies was satisfied (see Supplementary Information), the parameter update was set to zero for the last iteration (iteration 21). This lead to a rapid decrease of the inconsistency, such that the holding condition of the algorithm–converged parameters and small inconsistencies–became satisfied at the end of iteration 21.

4. Discussion. In this article, we presented a modular parameter identification algorithm for biomolecular networks. In our approach, we partition a biomolecular network into several, partly overlapping modules based not solely on the network's structure, but also on the set of species for which experimental data is available. These modules are defined such that the set of experimentally measured species also forms the interface between the modules. In contrast to existing approaches to modular identification, this module definition allows us to deal with highly interconnected networks, as demonstrated for the JAK2/STAT5 example. For the optimization, the modules and their parameter sensitivities are independently numerically integrated by using fixed "input" trajectories at every iteration, proxies for the inter-module communication initialized to splines through the experimental data. To achieve consistency between the modules at the end of–but not necessarily during–the optimization, these input trajectories are updated in each iteration such that they eventually converge to the actual interface species' dynamics. By allowing for inconsistencies between the interface species' dynamics and the input trajectories during the optimization, the computational requirements per iteration can be decreased for larger networks, the convergence properties are improved (at least for the tested examples), and the risk of convergence to local sub-optimal minima is reduced.

Future research might extend or adjust our modular parameter identification approach in different directions: one might allow for more than one experimentally measured species per module, or one might extend the algorithm to allow for experimental data in which not all interface species are measured under all experimental conditions. Furthermore, it would be interesting to define algorithms to automatically identify optimal matching times, and eventually to adaptively adjust their position and number during optimization. One could also analyze the convergence properties of our modular parameter identification framework for other sets of basis functions, for example, basis functions defined in the frequency rather than in the time domain. Interestingly, our modular parameter identification approach is in principle compatible with multiple-shooting [7, 24, 30]. Thus, it might be possible to combine both approaches, that is, to modularize a biomolecular network and to split its time domain into several intervals. We believe that such a combination would be promising to combine the advantages of both approaches and to further increase performance (see Supplementary Information for a detailed discussion). Finally, integrating our deterministic local parameter identification approach with more sophisticated (stochastic or deterministic) global approaches than multistart seems promising. Such modern hybrid approaches have been proposed to provide a high overall efficiency and robustness [34].

REFERENCES

[1] J Bachmann, A Raue, M Schilling, M E Böhm, C Kreutz, D Kaschek, H Busch, N Gretz, W D Lehmann, J Timmer, and U Klingmüller, *Division of labor by dual feedback regulators controls JAK2/STAT5 signaling over broad ligand range*, Molecular Systems Biology, 7 (2011).

[2] E Balsa-Canto, J R Banga, J A Egea, A Fernandez-Villaverde, and G M de Hijas-Liste, *Global optimization in systems biology: stochastic methods and their applications*, Advances in Experimental Medicine and Biology, 736 (2012), pp. 409–424.

[3] E Balsa-Canto, M Peifer, J R Banga, J Timmer, and C Fleck, *Hybrid optimization method with general switching strategy for parameter estimation*, BMC Systems Biology, 2 (2008), pp. 1–9.

[4] A Bar-Even, E Noor, Y Savir, W Liebermeister, D Davidi, D S Tawfik, and R Milo, *The moderately efficient enzyme: evolutionary and physicochemical trends shaping enzyme parameters*, Biochemistry, 50 (2011), pp. 4402–4410.

[5] R Bellman, *Dynamic Programming*, Princeton University Press, Princeton, New Jersey, 1957.

[6] H G Bock, *Numerical treatment of inverse problems in chemical reaction kinetics*, in Modelling of Chemical Reaction Systems, K H Ebert, P Deuflhard, and W Jäger, eds., vol. 18 of Springer Series in Chemical Physics, Springer Berlin Heidelberg, 1981, pp. 102–125.

[7] ———, *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 1987. In German.

[8] C G E Boender and A H G Rinnooy Kan, *Bayesian stopping rules for multistart global optimization methods*, Mathematical Programming, 37 (1987), pp. 59–80.

[9] C G E Boender, A H G Rinnooy Kan, G T Timmer, and L Stougie, *A stochastic method for global optimization*, Mathematical Programming, 22 (1982), pp. 125–140.

[10] F J Bruggeman, J L Snoep, and H V Westerhoff, *Control, responses and modularity of cellular regulatory networks: a control analysis perspective*, IET Systems Biology, 2 (2008), pp. 397–410.

[11] T F Coleman and Y Li, *A reflective Newton method for minimizing a quadratic function*

subject to bounds on some of the variables, SIAM Journal on Optimization, 6 (1996), pp. 1040–1058.

[12] J Ellson, E R Gansner, L Koutsofios, S C North, and G Woodhull, *Graphviz – open source graph drawing tools*, Graph Drawing, (2001), pp. 483–484.

[13] M B Elowitz and S Leibler, *A synthetic oscillatory network of transcriptional regulators*, Nature, 403 (2000), pp. 335–338.

[14] F N Fritsch and R E Carlson, *Monotone piecewise cubic interpolation*, SIAM Journal on Numerical Analysis, 17 (1980), pp. 238–246.

[15] W Huyer and A Neumaier, *Global optimization by multilevel coordinate search*, Journal of Global Optimization, 14 (1999), pp. 331–355.

[16] H-M Kaltenbach and J Stelling, *Modular analysis of biological networks*, Advances in Systems Biology, 736 (2012), pp. 3–17.

[17] B Kholodenko, M B Yaffe, and W Kolch, *Computational approaches for analyzing information flow in biological networks*, Science signaling, 5 (2012), pp. 1–14.

[18] B N Kholodenko, F J Bruggeman, and H M Sauro, *Mechanistic and modular approaches to modeling and inference of cellular regulatory networks*, in Systems Biology, Springer, 2005, pp. 143–159.

[19] G Koh, D Hsu, and P S Thiagarajan, *Component-based construction of bio-pathway models: The parameter estimation problem*, Theoretical Computer Science, 412 (2011), pp. 2840 – 2853.

[20] G Koh, H F C Teong, M-V Clément, D Hsu, and P S Thiagarajan, *A decompositional approach to parameter estimation in pathway modeling: a case study of the Akt and MAPK pathways and their crosstalk*, Bioinformatics, 22 (2006), pp. e271–e280.

[21] O Kotte and M Heinemann, *A divide-and-conquer approach to analyze underdetermined biochemical models*, Bioinformatics, 25 (2009), pp. 519–525.

[22] H Kuwahara, M Fan, S Wang, and X Gao, *A framework for scalable parameter estimation of gene circuit models using structural information*, Bioinformatics, 29 (2013), pp. i98–107.

[23] M Lang, S Summers, and J Stelling, *Cutting the wires: Modularization of cellular networks for experimental design*, Biophysical Journal, 106 (2014), pp. 321–331.

[24] S M Lenz, H G Bock, J P Schlöder, E A Kostina, G Gienger, and G Ziegler, *Multiple shooting method for initial satellite orbit determination*, Journal of guidance, control, and dynamics, 33 (2010), pp. 1334–1346.

[25] C Li, M Donizelli, N Rodriguez, H Dharuri, L Endler, V Chelliah, L Li, E He, A Henry, M I Stefan, J L Snoep, M Hucka, N Le Novere, and C Laibe, *BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models*, BMC Systems Biology, 4 (2010), pp. 1–14.

[26] G Lillacci and M Khammash, *Parameter estimation and model selection in computational biology*, PLoS Computational Biology, 6 (2010), p. e1000696.

[27] M Locatelli and F Schoen, *Random linkage: a family of acceptance/rejection algorithms for global optimisation*, Mathematical Programming, 85 (1999), pp. 379–396.

[28] C G Moles, P Mendes, and J R Banga, *Parameter estimation in biochemical pathways: a comparison of global optimization methods*, Genome Research, 13 (2003), pp. 2467–2474.

[29] P M Pardalos, H E Romeijn, and H Tuy, *Recent developments and trends in global optimization*, Journal of Computational and Applied Mathematics, 124 (2000), pp. 209–228.

[30] M Peifer and J Timmer, *Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting*, Systems Biology, IET, 1 (2007), pp. 78–88.

[31] T P Prescott and A Papachristodoulou, *Signal propagation across layered biochemical networks*, in Proceedings of the American Control Conference (ACC), 2014, pp. 3399–3404.

[32] A Sedoglavic, *A probabilistic algorithm to test local algebraic observability in polynomial time*, Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation, (2001), pp. 309–317.

[33] Y Shmaliy, *Continuous-Time Systems*, Springer Netherlands, Dordrecht, 2007, ch. Linear Time-Varying Systems, pp. 349–423.

[34] J Sun, J M Garibaldi, and C Hodgman, *Parameter estimation using metaheuristics in systems biology: a comprehensive review*, Computational Biology and Bioinformatics, IEEE/ACM Transactions on, 9 (2012), pp. 185–202.