Available online at www.sciencedirect.com
ScienceDirect
Current Opinion in Biotechnology

Equation discovery for systems biology: finding the structure and dynamics of biological networks from time course data

Sašo Džeroski[1] and Ljupčo Todorovski[2]

Reconstructing biological networks, such as metabolic and signaling networks, is at the heart of systems biology. Although many approaches exist for reconstructing network structure, few approaches recover the full dynamic behavior of a network. We survey such approaches that originate from computational scientific discovery, a subfield of machine learning. These take as input measured time course data, as well as existing domain knowledge, such as partial knowledge of the network structure. We demonstrate the use of these approaches on illustrative tasks of finding the complete dynamics of biological networks, which include examples of rediscovering known networks and their dynamics, as well as examples of proposing models for unknown networks.

Addresses
[1] Department of Knowledge Technologies, Jožef Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia
[2] Faculty of Administration, University of Ljubljana, Gosarjeva 5, SI-1000 Ljubljana, Slovenia

Corresponding author: Džeroski, Sašo (Saso.Dzeroski@ijs.si)

Current Opinion in Biotechnology 2008, **19**:360–368

This review comes from a themed issue on
Systems biology
Edited by Jaroslav Stark

Available online 5th August 2008

0958-1669/$ – see front matter

DOI 10.1016/j.copbio.2008.07.002

Introduction

The (re)construction of biological networks, including metabolic, regulatory, gene, and signaling networks, is of fundamental and immediate importance to the emerging field of computational systems biology [1]. The task to be addressed first in this context is the reconstruction of the structure of the network, that is, the variables of interest and their interactions. For metabolism, the networks focus on metabolites: the variables of interest are typically metabolite concentrations and the interactions among them biochemical reactions.

The dynamic behavior of biological networks is typically modeled by ordinary differential equation (ODE) models. Besides the dependences between compounds in the network, as specified by network structure, ODE models specify the exact nature of these dependencies through the functional form of the ODEs and their constant parameters (e.g. reaction rates). In a typical approach to ODE modeling of a biological network, a human domain expert[3] specifies the structure of the network and the functional form of the ODEs. Time course data about the behavior of the target biological network can be used to determine the values of the constant parameters in the ODEs.

Determining a set of ODEs from given time course data is referred to as system identification. The task of determining the functional form of a set of ODEs is referred to as structure identification. The task of determining appropriate values for the constant parameters is called parameter estimation. When the initial conditions of the ODE model are not known (which is often the case in biology), these have to be treated as additional parameters. In this article, we will discuss approaches to performing both of the above tasks simultaneously. The approaches we survey come from the area of machine learning [2,3], more specifically computational scientific discovery [4,5••], and are directly relevant to but not widely known in the systems biology community.

Computational scientific discovery of ODE models

Computational scientific discovery (CSD) [4,5••] is concerned with developing computer programs that automate or support some aspects of scientific discovery. The earliest and most prominent CSD systems, such as BACON [4], dealt with the problem of equation discovery, finding scientific laws in the form of equations. Although early CSD approaches considered algebraic equations, they were later extended to learn ODE models from time course data [6].

CSD is a subfield of machine learning [2,3] and artificial intelligence [7]. From this broader context, it inherits the fundamental approach of problem solving as heuristic search [8], which aims to find reasonably good (but not necessarily optimal) solutions to a given problem reasonably quickly. In particular, CSD programs for ODE discovery would search the space of ODE structures (functional forms) guided by heuristics related to how well the ODEs match the data. To judge how well an ODE structure matches given data, its parameters need to

[3] A domain is an area of study, that is, human physiology, and a domain expert is a person with considerable knowledge of the area.

be estimated. For ODE structures nonlinear in the parameters,[4] computationally expensive nonlinear optimization has to be used.

Another source of computational complexity is the size of the space of possible ODE model structures: this is typically huge and can easily be infinite. Of crucial importance is thus to define the space of ODE structures so as to keep it small and pertinent to capturing the dynamics of the modeled system. To achieve this, we have proposed the use of domain knowledge (about the area of study) in equation discovery [9••].

Different types of domain knowledge can be used in ODE discovery. We can start from existing ODE models for the system at hand (that are partial/incomplete/inaccurate) and revise/improve them in light of observed time course data. We can also provide a set of basic components as building blocks from which ODE models can be composed. Finally, we can provide a set of constraints that the ODE models we are willing to consider have to satisfy. Common to all of these is the explicit (declarative) statement of the modeling assumptions made concerning the space of ODE models considered. Below we briefly describe several CSD approaches to ODE discovery that can use domain knowledge of these types and illustrate them with examples related to biological networks.

Learning polynomial equations with constraints

The CSD system CIPER [5] [10,11] learns polynomial algebraic equations from data. Polynomial equations are linear in the parameters (cf. previous section), so CIPER can use linear regression to estimate the parameters efficiently. It performs heuristic search of the space of polynomial structures, with search proceeding from simple structures to more complex ones. The search starts with a structure consisting of a constant term only, which is gradually made more complex by adding new linear terms or multiplying existing terms with a variable.

For its search, the original CIPER uses a heuristic that combines model error and model complexity in an ad hoc fashion. The latest version of CIPER [12] uses the minimum-description length (MDL) principle [13] to combine these in a sound manner. It can take into account subsumption constraints, which specify partially known equation structures. Such a constraint might state (that is) that ax is to appear as a part/subpolynomial of the polynomial we are looking for: both $ax^2 + b$ and $ax + by$ satisfy this constraint.

[4] A structure is linear in a given parameter if the parameter appears as a multiplier in an additive term of the expression. that is, $y = ax^2 + b$ is linear in both a and b, while $y = x/(x + c)$ is nonlinear in c.

[5] The acronym CIPER stands for Constrained Induction of Polynomial Equations for Regression.

Note that CIPER has been designed for the discovery of algebraic equations: it handles ODEs by numerically introducing time derivatives of the system variables and treating these as dependent variables. CIPER can be applied repeatedly to produce equations for several dependent variables, that is, to produce a set of simultaneous equations. However, it can also search through the space of simultaneous equations directly [14]: this makes a lot of sense in modeling reaction networks, where variables that appear together in a reaction share terms in the corresponding equations.

Polynomial ODEs are often used to model biological networks, such as the cell-cycle regulation in fission yeast [15] or the cycling of Rho GTPases within protein complexes [16]. Constraints in CIPER are useful in the context of reconstructing such networks [11] from partial structures (see Figure 1). Given time course data obtained by simulating the ODEs and the constraints resulting from the partial network structure, CIPER successfully reconstructs the target ODE model. Without the constraints, however, the reconstruction is not completely successful: this illustrates the crucial role that domain knowledge can play.

Grammar-based equation discovery

To represent the possible space of ODE structures, we can view it as a language, with individual equation structures being sentences. A formal grammar can then be used to describe the language. The equation discovery system LAGRAMGE [17] uses the formalism of context-free grammars (CFG) for this purpose.

A CFG is defined by a set of terminals T, a set of nonterminals N, a set of productions P, and a starting symbol S from N. Terminals are the symbols that actually appear in the sentences of the language (e.g. words and punctuation in English). The terminal symbols S_3– S_7 in the CFG in Figure 2c represent system variables, I_1 and I_2 represent input variables. A special terminal symbol *const* denotes a constant parameter whose value (from the interval provided) is to be selected to match the data best. Nonterminals (or syntactic categories) represent classes of subexpressions or phrases in the language represented by the grammar (e.g. a nounphrase in English). The nonterminals R_2 and R_4 represent arithmetic expressions for modeling chemical reactions that involve two compounds Y_1 and Y_2. Productions (or rewrite rules) specify how a nonterminal can be replaced by a sequence of terminals and nonterminals (e.g. a nounphrase can be a determiner followed by a noun). The rewrite rules for R_2 and R_4 specify noncompetitive inhibition of the compounds Y_1 and Y_2 following the Michaelis–Menten kinetic model.

Given a CFG, we can check whether a sentence/expression belongs to the language defined by the grammar (parse task) or generate/derive expressions that belong to

Figure 1

(a) A reaction network:

$$\{x_5, \mathbf{x_7}\} \rightarrow \{x_1\}$$
$$\{x_1\} \rightarrow \{x_2, x_3\}$$
$$\{x_1, x_2, \mathbf{x_7}\} \rightarrow \{x_3\}$$
$$\{x_3\} \rightarrow \{x_4\}$$
$$\{x_4\} \rightarrow \{x_2, \mathbf{x_6}\}$$
$$\{\mathbf{x_4}, \mathbf{x_6}\} \rightarrow \{\mathbf{x_2}\}$$

(b) Partial equation structure:

$$\dot{x_1} = -c_1 \cdot x_1 + c_2 \cdot x_5 - c_3 \cdot x_1 \cdot x_2$$
$$\dot{x_2} = c_4 \cdot x_1 + c_5 \cdot x_4 - c_6 \cdot x_1 \cdot x_2$$
$$\dot{x_3} = c_7 \cdot x_1 + c_8 \cdot x_1 \cdot x_2 - c_9 \cdot x_3$$
$$\dot{x_4} = c_{10} \cdot x_3 - c_{11} \cdot x_4$$
$$\dot{x_5} = -c_{12} \cdot x_5$$

(c) Complete equations with reaction rates:

$$\dot{x_1} = 0.8 \cdot x_5 \cdot x_7 - 0.5 \cdot x_1 - 0.7 \cdot x_1 \cdot x_2 \cdot x_7$$
$$\dot{x_2} = 0.7 \cdot x_1 + 0.2 \cdot x_4 + 0.1 \cdot x_4 \cdot x_6 - 0.3 \cdot x_1 \cdot x_2 \cdot x_7$$
$$\dot{x_3} = 0.4 \cdot x_1 + 0.3 \cdot x_1 \cdot x_2 \cdot x_7 - 0.2 \cdot x_3$$
$$\dot{x_4} = 0.5 \cdot x_3 - 0.7 \cdot x_4 \cdot x_6$$
$$\dot{x_5} = -0.6 \cdot x_5 \cdot x_7$$
$$\dot{x_6} = 0.2 \cdot x_4 - 0.8 \cdot x_4 \cdot x_6$$
$$\dot{x_7} = -0.1 \cdot x_1 \cdot x_2 \cdot x_7 - 0.1 \cdot x_5 \cdot x_7$$

Current Opinion in Biotechnology

A reaction network **(a)**, consisting of six reactions, that was successfully reconstructed from simulated data and a partial specification of the network structure by constrained induction for polynomial regression (CIPER) [11]. The parts given in bold are assumed not to be known for the reconstruction task, but all the variables are assumed to be measured. The partial specification of the equation structure **(b)** is derived from the known part of the network: the polynomials in the partial structure have to be subpolynomials of the corresponding polynomials found by CIPER and are supplied to CIPER as subsumption constraints. Simulated data were obtained from the complete equations **(c)**, which were successfully reconstructed when CIPER was given both simulated data and a partial structure. When given only simulated data, CIPER searched a much larger space of equation structures and failed to reconstruct correctly the most complex equations, namely the ones for $\dot{x_1}$ and $\dot{x_2}$.

the language (generate task). For both purposes, we use the notion of a parse tree, which describes the way a certain expression can be derived using the grammar productions. Derivation always starts with the starting symbol S and applies production rules in an iterative manner until an expression that consists of only terminal symbols is reached. Figure 2c shows a CFG that defines a set of ODE structures used by Gennemark and Wedelin [18••] to model a reaction network due to Arkin and Ross [19](graphically shown in Figure 2a). Figure 2d shows a parse tree for the right-hand side of one of the original equations.

LAGRAMGE performs heuristic (or exhaustive) search over the space of ODE model structures defined by depth-bounded derivation trees for a given CFG. During the search, it keeps several alternative ODE structures found to be best so far, according to a criterion which combines the ODE model error and its complexity. To calculate the ODE model error with respect to given time course data, LAGRAMGE fits the constant parameters in the ODEs by nonlinear optimization using an algorithm [20] for solving the generalized nonlinear least-squares problem [21].

Representing process-based models

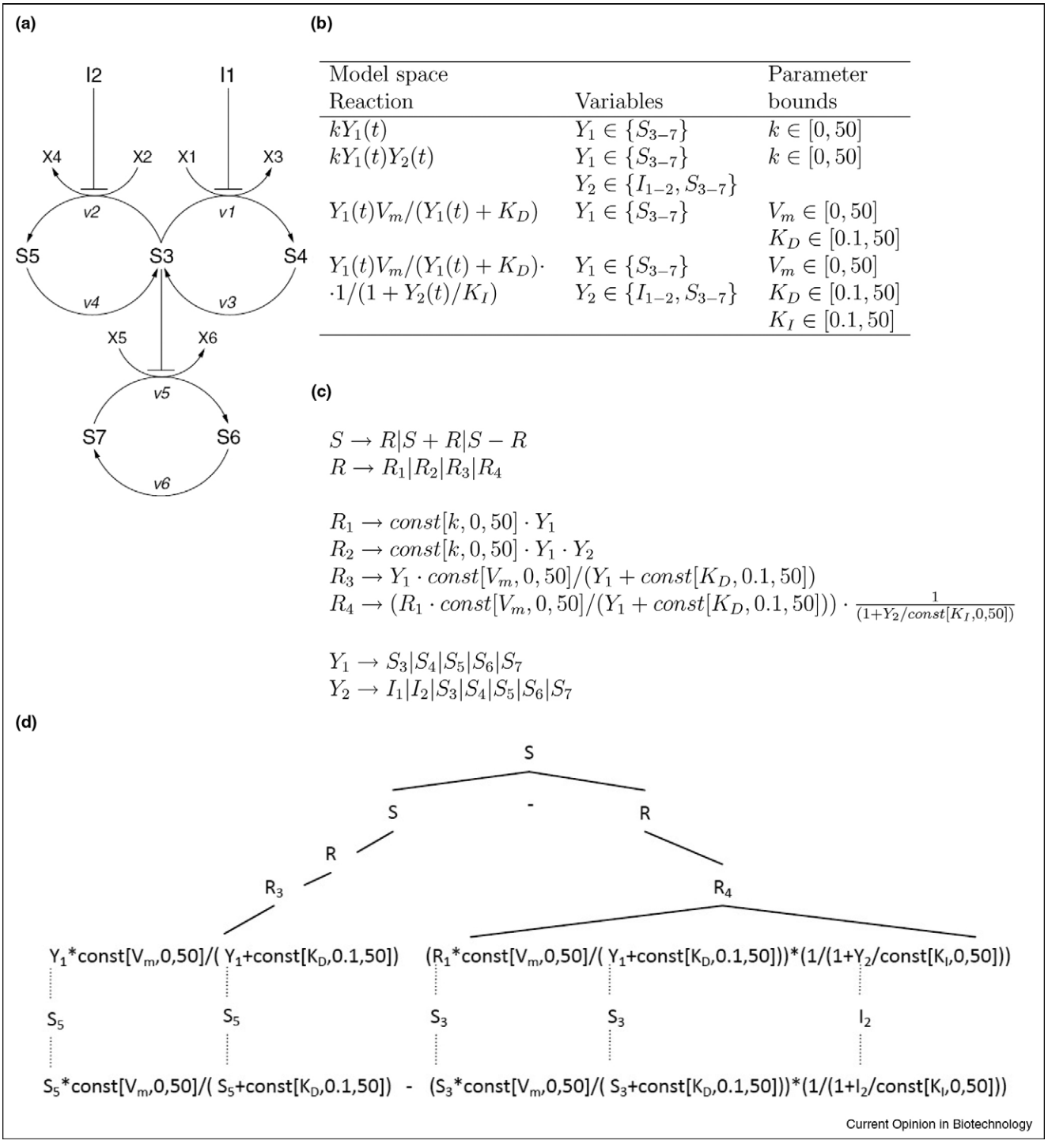
Grammars are an expressive formalism for representing many different types of domain knowledge, including existing models to be revised, incomplete/partial models, and knowledge-based building blocks for modeling in a particular domain [9••]. Note, however, that a grammar is specific to the modeling task at hand, that is, the grammar in Figure 2c is specifically intended for modeling the metabolic network of Arkin and Ross [19]. Also, grammar formalisms make little contact with the formalisms typically used by mathematical modelers and scientists and are thus difficult to use.

The representation of process-based models (PBMs) and process-based domain knowledge (PBDK) is more general and accessible to scientists and engineers, who often state their explanations in terms of processes that govern the behavior of an observed dynamic system. It also connects the explanatory and predictive aspects of modeling, by directly linking processes to the mathematical formulations cast in terms of equations. A basic set of generic processes or process classes can be identified for a domain of interest: together with some formulations cast in terms of equations, this constitutes domain knowledge than can be reused across different modeling tasks in the same domain. In population dynamics, processes include the growth and decay of a population or interactions between species [22]; in system biology, processes may correspond to biochemical reactions or regulatory influences.

Several formalisms for representing PBMs and PBDK have been proposed recently. Todorovski and Džeroski [23–25•,9••] propose a formalism for PBDK that comprises three components: a hierarchy of variable types, a hierarchy of process and function classes, and a combining scheme. The processes and functions relate variable types and specify model structures for individual processes, while the combining scheme specifies how the models of individual processes are combined into a model of the entire observed system. Figure 3b depicts PBDK for modeling biochemical reactions (in the S-system[6] style [26]) expressed in this formalism and a hypothetical

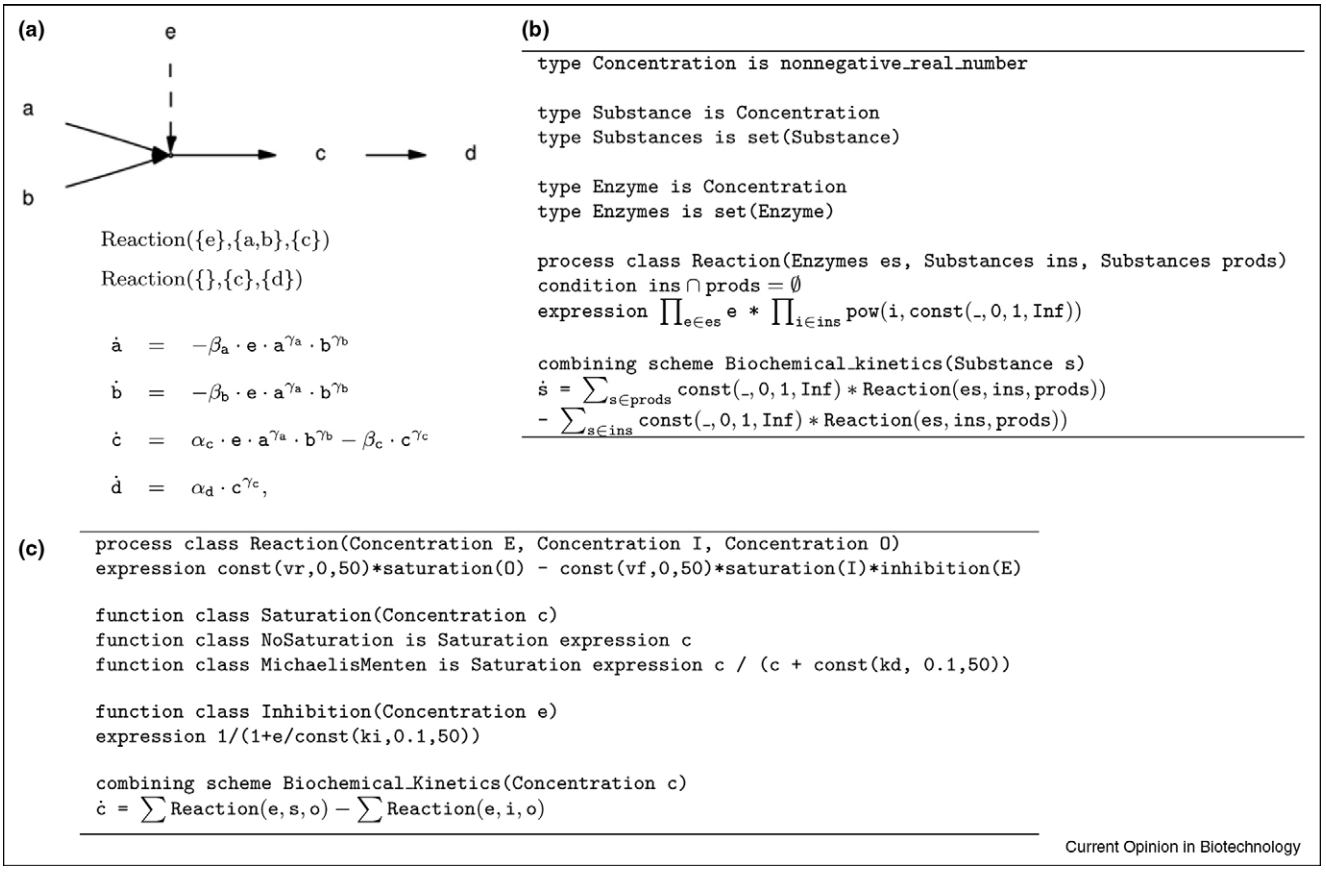
[6] The S-system represents biochemical processes using power-law expansions in the system variables.

Figure 2



A metabolic system **(a)** taken from Arkin and Ross [19] and used by Gennemark and Wedelin [18••] to evaluate their approach on the task of reconstructing the full dynamics of a system from simulated data. The corresponding ODEs are given in Appendix of [18••], Eqs. (3)–(12). The model space considered is specified through the possible reactions (i.e. the functional forms of terms to be included in the equations, the involved variables, and the ranges of possible values for the constant parameters **(b)**). The model space can also be represented by a context-free grammar **(c)** that can be used by LAGRAMGE together with time course data. LAGRAMGE searches through the space of possible equation structures represented by depth-limited parse trees that can be derived from the grammar, such as the one shown here **(d)**: this parse tree derives the equation for the derivative of $S_5(t)$ in the network shown in (a) by using production rules from the grammar shown in (c).

Figure 3



Process-based models and domain knowledge. **(a)** An example network of two reactions and its corresponding ODE model according to the *S*-system formalism. The network comprises two reactions, the first of which has two inputs (*a* and *b*) and one product *c* and is catalyzed by an enzyme *e*. **(b)** Process-based domain knowledge (PBDK) for ODE models based on the S-system formalism ready to use by LAGRAMGE2.0. There is only one class of processes, corresponding to reactions, where each reaction gives rise to a product of powers of the concentrations of participating substrates, products, and enzymes; the combining scheme adds up the influences of all reactions in which a component participates to obtain its rate of change. **(c)** LAGRAMGE 2.0 PBDK for modeling the metabolic system from Figure 2, simplified to consider only inhibited negative fluxes. The only class of processes (reaction) has an input *I* and output *O* and is catalyzed by an enzyme. Variants are possible, that is, saturated and unsaturated, expressed through the function definitions. The noncompetitive inhibition follows the Michaelis–Menten kinetic model.

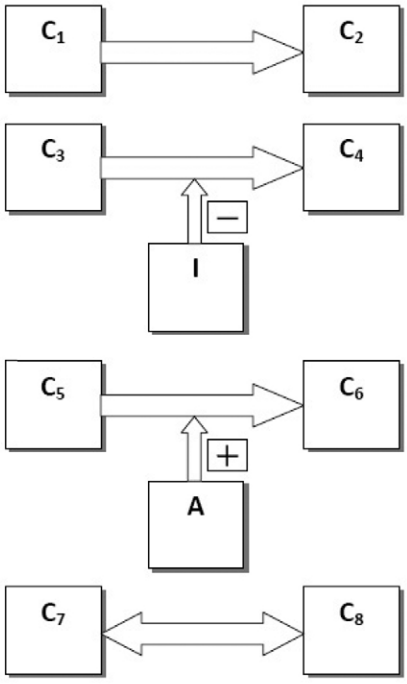
reaction network consisting of two processes (Figure 3a) with the corresponding equation structures. In the example shown there, variable types represent the class of chemical compounds (substances) and its subclass of enzymes. For simplicity, we have chosen the process class 'reaction' to correspond to a single chemical reaction. However, we can easily refine the hierarchy of processes by splitting the class 'reaction' in two subclasses, reversible and irreversible reactions. Finally, the combining scheme specifies how the influences of several chemical reactions that impact a single compound are combined in the differential equation for that compound.

Figure 3c depicts PBDK about the metabolic system from Figure 2a. Again, a single process class reaction is defined that involves an enzyme, an input and an output substance. Two variants of the process are possible that include the possibility of saturated or unsaturated reactions. The reaction includes noncompetitive inhibition of the compounds following the Michaelis–Menten kinetic model.

While the above formalism can be used to represent PBDK, Langley *et al.* [27,28••] propose a formalism for representing both PBDK and PBMs. This formalism uses generic processes to describe PBDK and specific processes (with specific variables and constant parameter values) to describe PBMs. PBDK for modeling chemical reactions, including irreversible and reversible ones, activation and inhibition, expressed in this formalism, is given in Figure 4a and b. Figure 4b defines the generic processes corresponding to each of the above classes of reactions, as well as the process of flux combination which combines the effects of different reactions on a single compound. Figure 4d gives an excerpt from a PBM for glycolysis, consisting of three specific processes: two of

Figure 4

(a)



process flux_combination;
variables $C\{conc\}, C_{+flux}\{flux\}, C_{-flux}\{flux\}, C_{+rate}\{rate\}, C_{-rate}\{rate\}$;
equations $d[C,t,1] = C_{+rate} * C_{+flux} + C_{-rate} * C_{-flux}$;
process irreversible;
variables $C1\{conc\}, C1_{-flux}\{flux\}, C2_{+flux}\{flux\}$;
parameters $kinetic\_order1[0,1], kinetic\_order2[0,1]$;
equations $C1_{-flux} = C1^{kinetic\_order1}$;
$C2_{+flux} = C1^{kinetic\_order2}$;
process inhibition;
variables $C3_{-flux}\{flux\}, C4_{+flux}\{flux\}, E\{conc\}$;
parameters $kinetic\_order1[0,1], kinetic\_order2[0,1]$;
equations $C3_{-flux} = E^{(-kinetic\_order1)}$;
$C4_{+flux} = E^{(-kinetic\_order2)}$;
process activation;
variables $C5_{-flux}\{flux\}, C6_{+flux}\{flux\}, E\{conc\}$;
parameters $kinetic\_order1[0,1], kinetic\_order2[0,1]$;
equations $C5_{-flux} = E^{kinetic\_order1}$;
$C6_{+flux} = E^{kinetic\_order2}$;
process reversible;
variables $C7_{+flux}\{flux\}, C7_{-flux}\{flux\}, C8_{+flux}\{flux\}, C8_{-flux}\{flux\}$,
$C7\{conc\}, C8\{conc\}$;
parameters $k\_o1\_1p[0,1], k\_o1\_2p[0,1], k\_o1\_1n[0,1]$,
$k\_o1\_2n[0,1], k\_o2\_1p[0,1], k\_o2\_2p[0,1]$,
$k\_o2\_1n[0,1], k\_o2\_2n[0,1]$;
equations $C7_{+flux} = C7^{k\_o1\_1p} * C8^{k\_o1\_2p}$;
$C7_{-flux} = C7^{k\_o1\_1n} * C8^{k\_o1\_2n}$;
$C8_{+flux} = C7^{k\_o2\_1p} * C8^{k\_o2\_2p}$;
$C8_{-flux} = C7^{k\_o2\_1n} * C8^{k\_o2\_2n}$;

(c)

(d)

model glycolysis_kinetics;
...
process flux_combination_2DHAP;
variables $2DHAP\{conc\}, 2DHAP_{+flux}\{flux\}, 2DHAP_{-flux}\{flux\}$;
equations $d[2DHAP,t,1] = 1.5514 * 2DHAP_{+flux} + 0.2402 * DHAP_{-flux}$;
...
process irreversible_2DHAP_G6P;
variables $2DHAP\{conc\}, 2DHAP_{-flux}\{flux\}, G6P_{+flux}\{flux\}$;
equations $2DHAP_{-flux} = G6P^{0}$;
$G6P_{+flux} = G6P^{0.6812}$;
...
process irreversible_2DHAP_3PG;
variables $2DHAP\{conc\}, 2DHAP_{-flux}\{flux\}, 3PG_{+flux}\{flux\}$;
equations $2DHAP_{-flux} = 3PG^{0}$;
$G6P_{+flux} = 3PG^{0.9455}$;
...

Current Opinion in Biotechnology

The use of inductive process modeling (IPM) [28••] in the domain of biochemical kinetics [29••], addressing the task of modeling glycolysis from measured data [30]. **(a)** Schematics for four types of biochemical reactions: activation, inhibition, irreversible, and reversible. **(b)** IPM process-based domain knowledge for each of the reaction types and their combination. Four generic processes correspond to the four types of reactions. In addition, the process of flux combination combines the effects of individual reactions on a compound. **(c)** The network proposed by Torralba *et al*. [30] and the network learned by IPM (right) from measured data and the domain knowledge above. The network fits the data well, but differs from the one proposed by Torralaba *et al*. No inhibition/activation reactions are considered by IPM, because no unobserved enzymes are given as candidate

these are irreversible reactions and one is a flux combination. They correspond to a small part of the network depicted on the right of Figure 4c (reactions 2DHAP → G6P and 2DHAP → 3PG). Note that the specific processes follow the templates set by the generic ones, but include specific values of the parameters (kinetic orders, reaction rates).

Learning process-based models

LAGRAMGE2.0 [23–25•,9••] learns PBMs by transforming PBDK into grammars and applying LAGRAMGE in turn. Given PBDK as described above (variable types, processes and functions, and combining schemes) and a specific modeling task (measured variables and their types), LAGRAMGE2.0 generates a grammar that defines the space of ODEs that correspond to PBMs specified by the PBDK. LAGRAMGE is then used to search this space and find an optimal model for the observed system behavior (time course data).

Inductive process modeling (IPM) [28••], on the contrary, performs heuristic search directly through the space of PBMs. Given a modeling task specification, IPM takes the template process models (i.e. generic processes from Figure 4b) from the PBDK and turns them into a number of specific process models that represent model components. that is, given three metabolites, G3P, 3PG, and 2DHAP, the reversible model template (generic process) from Figure 4a and b can result into three candidate chemical reactions, that is, G3P ↔ 3PG, 3PG ↔ 2DHAP, and G3P ↔ 2DHAP.

IPM then searches through the space of combinations of processes (model components), where each combination represents a complete model of the dynamic system, in order to find the optimal one. For each candidate model, IPM performs full simulation of the model equations and matches the simulated against the observed behavior. It is thus capable of learning models that include unobserved system variables, that is, variables whose values have not been directly measured/observed. This capability is important when only few metabolite concentrations or gene expression levels in the biological network under study are directly observable, which is typically the case in systems biology.

IPM was successfully applied in the domain of biochemical kinetics [29••] constructing a network model of glycolysis from measured data [30] and PBDK. Some details on this case study are given in Figure 4. IPM proposed a network that fits the measured data well, but differs from the network proposed by the researchers that collected the data [30]. Note that the IPM search through the space of all combinations of model components leads to a search space whose size grows exponentially with the number of processes included in the model. To make this strategy feasible for complex domains, one must add structural constraints, specifying, that is, which processes should be included in the model or which processes are mutually exclusive. The HIPM system [31] accepts structural constraints stated as a hierarchy of generic processes. Structural constraints specify basic modeling rules, such as 'two specific metabolites can only be involved in one type of reaction (reversible or irreversible)' or 'each metabolite should be involved in at least one reaction'.

Other recent work

Recent work in CSD related to the discovery of biological networks includes work on learning qualitative models of metabolic [32•] and genetic [33] networks. Garret *et al.* [32•] learn qualitative differential equations, which have the same functional form as ODE models for the *S*-system formalism, with products of variables (instead of powers thereof), but no specific values for the constant coefficients. Zupan *et al.* [33] reconstruct qualitative genetic networks from the outcomes of knockout and overexpression experiments and background knowledge (known gene-to-gene and gene-to-outcome interactions).

The above methods infer the structure of a network, without describing its dynamic behavior. Many methods address this task, a survey of which is given by Price and Schmulevich [1]. Network structure can be reconstructed by using information from the literature and databases, or by reverse engineering from genome-wide data on transcriptomics and proteomics [34••]. In the latter case, steady-state data [35] or time course data [36] can be used as input. An example is the method by Arkin and Ross [19], where a factor analysis of the correlations between time course data on measured variables is conducted and the results are manually interpreted to arrive at a network structure.

A few approaches have explicitly addressed the task of reconstructing both network structure and dynamics, two of which come from the area of evolutionary computation. Koza *et al.* [37•] use genetic programming to reconstruct a metabolic network, where simulated data and information on the types of reactions are taken into account. Kikuchi *et al.* [38] use a genetic algorithm to reconstruct a genetic network defined in the *S*-system formalism [26]. A recent approach by Gennemark and Wedelin [18••] performs heuristic search over an ad hoc defined space of ODE

(Figure 4 Legend Continued) catalysts for these. **(d)** An excerpt from the actual process-based model output by IPM. Two irreversible reactions (2DHAP → G6P and 2DHAP → 3PG) are included. The corresponding processes are specific forms of the generic process irreversible from (a), with specific variables listed, as well as the kinetic order constants specified. In addition, a specific flux combination process is included, which combines the influences of the two reactions on 2DHAP and specifies the reaction rates.

structures to rediscover a metabolic [19] and a genetic network [38].

Outlook

The task of finding the structure and dynamics of biological networks is of central interest to computational systems biology. In this article, we have given a survey of equation discovery methods that perform this task, taking as input time course data, as well as different types of domain knowledge (such as partial network structure). We have demonstrated the use of these approaches on illustrative tasks of finding the complete dynamics of biological networks, which included examples of rediscovering known networks and their dynamics, as well as examples of proposing models for unknown networks.

The task at hand is a difficult one, as data are scarce in typical systems biology endeavors. Even the task of identifying parameter values for a known ODE structure requires a number of measurements proportional to the number of parameters [39]. Searching through a large space of possible ODE structures makes the problem worse. However, the approaches discussed here are able to leverage the data with domain knowledge and thus potentially reduce the amount of data needed: this is a key feature of interest.

A promising direction for further work is to incorporate recently developed approaches to the task of parameter identification for ODE structures from short time courses [40•], within the equation discovery approaches discussed here. Many other challenges remain to be addressed for the successful use of equation discovery methods in systems biology. One is certainly the casting of domain knowledge for different formalisms that are frequently used to model biological networks (such as the *S*-system) into a form usable by equation discovery. The approaches described above can take advantage of the output of methods that reconstruct network structure; however, it still remains an open issue how to formulate network structures expressed in different formalisms as domain knowledge for discovering models of the full dynamic behavior of biological networks.

Acknowledgements

The authors would like to thank Kathy Astrahantsef, Benedikt Brors, Emmanuelle Caron, Brian Robertson, and Jaroslav Stark for their valuable comments on a draft of this article. Sašo Džeroski acknowledges the support by the EU-funded projects IQ (Inductive Queries for Mining Patterns and Models) and EETP (European Embryonal Tumor Pipeline).

References and recommended reading

Papers of particular interest, published within the annual period of review, have been highlighted as:

- • of special interest
- •• of outstanding interest

1. Price ND, Shmulevich I: **Biochemical and statistical network models for systems biology**. *Curr Opin Biotechnol* 2007, **18**:365-370.

2. Langley P: **Elements of Machine Learning**. San Francisco: Morgan Kaufmann; 1996.

3. Mitchell TM: **Machine Learning**. New York: McGraw Hill; 1997.

4. Langley P, Simon HA, Bradshaw GL, Żytkow JM: **Scientific Discovery**. Cambridge, MA: MIT Press; 1987.

5. •• Džeroski S, Todorovski L. (Eds.): **Computational Discovery of Scientific Knowledge**. Berlin: Springer; 2007.An overview of the state-of-the-art in computational scientific discovery, including several approaches to equation discovery and applications to finding the structure and dynamics of biological networks.

6. Džeroski S, Todorovski L: **Discovering dynamics: from inductive logic programming to machine discovery**. *J Intell Inf Syst* 1995, **4**:89-108.

7. Russel S, Norvig P: **Artificial Intelligence: A Modern Approach**. edn 2. Upper Saddle River, NJ: Prentice Hall; 2003.

8. Pearl J: **Heuristics: Intelligent Search Strategies for Computer Problem Solving** New York: Addison-Wesley; 1983.

9. •• Todorovski L, Džeroski S, Džeroski S, Todorovski L: **Computational Discovery of Scientific Knowledge**. Edited by Džeroski S, Todorovski L. Berlin: Springer; 2007:69–97. Describes how grammar-based equation discovery can be used to take into account different types of domain knowledge, including process-based domain knowledge about basic processes that govern the dynamics of systems in the area of study.

10. Todorovski L, Ljubič P, Džeroski S: **Inducing polynomial equations for regression**. *Lect Notes Comput Sci* 2004, **3201**:441-452.

11. Džeroski S, Todorovski L, Ljubič P: **Using constraints in discovering dynamics**. *Lect Notes Comput Sci* 2003, **2843**:297-305.

12. Pečkov A, Džeroski S, Todorovski L: **A minimal description length scheme for polynomial regression**. *Lect Notes Comput Sci* 2008, **5012**:284-295.

13. Grünwald PD: **The Minimum Description Length Principle**. Cambridge, MA: MIT Press; 2007.

14. Pečkov A, Džeroski S, Todorovski L. **Multitarget polynomial regression with constraints**. In *Proceedings of the ECML/PKDD International Workshop on Constraint-Based Mining and Learning*, Edited by Nijssen S, de Raedt L. Warsaw, Poland: Warsaw University; 2007:61–72.

15. Tyson JJ, Chen K, Novak B: **Network dynamics and cell physiology**. *Nat Rev Mol Cell Biol* 2001, **2**:908-916.

16. Goryachev AB, Pokhilko AV: **Computational model explains high activity and rapid cycling of Rho GTPases within protein complexes**. *PLOS Comp Biol* 2006, **2**:1511-1521.

17. Todorovski L, Džeroski S. **Declarative bias in equation discovery**. In *Proceedings of the Fourteenth International Conference on Machine Learning*, Edited by Fisher DH. San Mateo, CA: Morgan Kaufmann; 1997:376–384.

18. •• Gennemark P, Wedelin D: **Efficient algorithms for ordinary differential equation model identification of biological systems**. *IET Syst Biol* 2007, **1**:120-129.
An approach to equation discovery designed with finding the structure and dynamics of biological networks in mind; similar to, but less formalized than grammar-based equation discovery.

19. Arkin RP, Ross J: **Statistical construction of chemical reaction mechanisms from measured time-series**. *J Phys Chem* 1995, **99**:970-979.

20. Bunch DS, Gay DM, Welsch RE: **Algorithm 717: subroutines for maximum likelihood and quasi-likelihood estimation of parameters in nonlinear regression models**. *ACM Trans Math Soft* 1993, **19**:109-130.

21. Dennis JE, Gay DM, Welsch RE: **Algorithm 573: NL2SOL—an adaptive nonlinear least-squares algorithm**. *ACM Trans Math Soft* 1981, **7**:369-383.

22. Atanasova N, Todorovski L, Džeroski S, Rekar Remec S, Recknagel F, Kompare B: **Automated modelling of a food web in**

lake Bled using measured data and a library of domain knowledge. *Ecol Model* 2006, **194**:37-48.

23. Džeroski S, Todorovski L, Magnani L, Nersessian NJ, Pizzi C: Encoding and using domain knowledge on population dynamics for equation discovery. Logical and Computational Aspects of Model-Based Reasoning. Edited by Magnani L, Nersessian NJ, Pizzi C. Dordrecht: Kluwer; 2002:227–247.

24. Todorovski L, Džeroski S: **Using domain specific knowledge for automated modeling**. *Lect Notes Comput Sci* 2003, **2810**:48-59.

25. Todorovski L, Džeroski S: **Integrating knowledge-driven and**
• **data-driven approaches to modeling**. *Ecol Model* 2006, **194**:3-13.
Inducing process-based models through grammar-based equation discovery, demonstrating the utility of domain knowledge in environmental applications.

26. Voit EO: **Computational Analysis of Biochemical Systems**. Cambridge, UK: Cambridge University Press; 2000.

27. Langley P, Sanchez J, Todorovski L, Džeroski S. **Inducing process models from continuous data**. In *Proceedings of the Nineteenth International Conference on Machine Learning*, Edited by Sammut C, Hofmann A. San Mateo, CA: Morgan Kaufmann; 1997:347–354.

28. Bridewell W, Langley P, Todorovski L, Džeroski S: **Inductive**
•• **process modeling**. *Mach Learn* 2008, **71**:132.
Introduces a formalism for representing process-based models and domain knowledge, and a learning method that searches the space of process-based models directly.

29. Langley P, Shiran O, Shrager J, Todorovski L, Pohorille A:
•• **Constructing explanatory process models from biological data and knowledge**. *Artif Intell Med* 2006, **37**:191-201.
A successful application of the process-based modeling approach to the task of constructing a biological network model from time course data, describing the dynamics of glycolysis.

30. Torralba A, Yu K, Shen P, Oefner P, Ross J: **Experimental test of a method for determining causal connectivities of species in reactions**. *Proc Natl Acad Sci* 2003, **100**:1494-1498.

31. Todorovski L, Bridewell W, Shiran O, Langley P. **Inducing hierarchical process models in dynamic domains**. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, Edited by Veloso MM, Kambhampati S. Pittsburgh, PA: AAAI Press; 2005:892–897.

32. Garrett SM, Coghill GM, Srinivasan A, King RD: Learning
• qualitative models of physical and biological systems. Computational Discovery of Scientific Knowledge. Edited by Džeroski S, Todorovski L. Berlin: Springer; 2007:248–272. Describes an approach to learning qualitative models of dynamic systems and illustrates its relevance to finding biological networks on the glycolysis pathway.

33. Zupan B, Bratko I, Dems? ar J, Juvan P, Kuspa A, Halter JA, Shaulsky G: Discovery of genetic networks through abduction and qualitative simulation. Computational Discovery of Scientific Knowledge. Edited by Džeroski S, Todorovski L. Berlin: Springer; 2007:228–247.

34. Zapatka M, Koch Y, Brors B: **Ontological analysis and pathway**
•• **modelling in drug discovery**. *J Pharm Med* 2008, **22**:99-105.
Addresses the task of identifying significantly altered pathways in genome-wide data, discusses pathway representations for mathematical modelling and pathway analysis methods, and gives some examples of the successful modelling of signal transduction pathways.

35. Ideker T, Thorsson V, Ranish JA, Christmas R, Buhler J, Eng JK, Bumgarner R, Goodlett DR, Aebersold R, Hood L: **Integrated genomic and proteomic analyses of a systematically perturbed metabolic network**. *Science* 2001, **292**:929-934.

36. Sontag E, Kiyatkin A, Kholodenko BN: **Inferring dynamic architecture of cellular networks using time series of gene expression, protein and metabolite data**. *Bioinformatics* 2004, **20**:1877-1886.

37. Koza JR, Mydlowec W, Lanza G, Yu J, Keane MA: Automatic
• computational discovery of chemical reaction networks using genetic programming. Computational Discovery of Scientific Knowledge. Edited by Džeroski S, Todorovski L. Berlin: Springer; 2007:205–227. A genetic programming approach to finding biological networks, illustrated by reconstructing a part of a phospholipid cycle from simulated data.

38. Kikuchi S, Tominaga D, Arita M, Takahashi K, Tomita M: **Dynamic modeling of genetic networks using genetic algorithm and *S*-system**. *Bioinformatics* 2003, **19**:643-650.

39. Sontag ED: **For differential equations with *r* parameters,** $2r+1$ **experiments are enough for identification**. *J Nonlinear Sci* 2002, **12**:553-583.

40. Brewer D, Barenco M, Callard R, Hubank M, Stark J: **Fitting**
• **ordinary differential equations to short time course data**. *Philos Trans A Math Phys Eng Sci* 2008, **366(1865)**:519-544.
Introduces and evaluates a new efficient technique for estimating parameters in ordinary differential equations (linear in the parameters), particularly suited to situations where the number of data points is low.