DETERMINISTIC AND STOCHASTIC PROCESS-BASED MODELING AND DESIGN OF DYNAMICAL SYSTEMS IN BIOLOGY

Jovan Tanevski

Doctoral Dissertation
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Supervisor: Prof. Dr. Sašo Džeroski, Jožef Stefan Institute, Ljubljana, Slovenia
Co-Supervisor: Prof. Dr. Ljupčo Todorovski, University of Ljubljana,
Faculty of Administration, Ljubljana, Slovenia

Evaluation Board:
Prof. Dr. Đani Juričić, Chair, Jožef Stefan Institute, Ljubljana, Slovenia
Prof. Dr. Blaž Zupan, Member, University of Ljubljana,
Faculty of Computer and Information Science, Ljubljana, Slovenia
Prof. Dr. Michael Stumpf, Member, Imperial College London, London, United Kingdom

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL

Jovan Tanevski

DETERMINISTIC AND STOCHASTIC PROCESS-BASED MODELING AND DESIGN OF DYNAMICAL SYSTEMS IN BIOLOGY

Doctoral Dissertation

DETERMINISTIČNO IN STOHASTIČNO PROCESNO MODELIRANJE IN NAČRTOVANJE DINAMIČNIH SISTEMOV V BIOLOGIJI

Doktorska disertacija

Supervisor: Prof. Dr. Sašo Džeroski

Co-Supervisor: Prof. Dr. Ljupčo Todorovski

Ljubljana, Slovenia, July 2016

Acknowledgments

I would like to express my gratitude to my supervisor Sašo Džeroski and co-supervisor Ljupčo Todorovski for their guidance and support during the entire course of research that led to this dissertation. I am especially thankful for the motivating discussions on various topics related to this work as well as machine learning, scientific discovery and academia in general. Additionally, I would like to thank the members of the evaluation board: Đani Juričić, Blaž Zupan and Michael Stumpf for the careful reading and the constructive feedback on this work.

I acknowledge the financial support from the Slovene Human Resources and Scholarship Fund (11011-49/2011), received through their postgraduate scholarship program, and the research funding received through the EU projects PHAGOSYS — Systems biology of phagosome formation and maturation - modulation by intracellular pathogens (FP7-HEALTH 223451), SUMO — Super modeling by combining imperfect models (FP7-ICT 266722) and HBP — The Human Brain Project (FP7-ICT 604102), which I took part in. This work would not be possible without the availability and the access to the computer clusters provided by the Centre of Excellence for Integrated Approaches in Chemistry and Biology of Proteins (CIPKeBiP) and the Academic and Research Network of Slovenia (ARNES), through the Slovenian National Grid Initiative, for which I am also grateful.

I would next like to thank Michael Stumpf, Cristopher Barnes and Daniel Silk. Their work in the area of modeling and design of dynamical biological systems had a great impact on me and my work. I am thankful for the motivation and the constructive discussions during the early stages of this work. I would also like to thank Yannis Kalaidzidis for the invaluable input and critical advice on the Rab5-Rab7 switch and on the topic of modeling endocytosis during our collaboration.

I would like to thank my officemate Darko Cherepnalkoski, for sharing his view on process-based modeling with me. I also appreciate the numerous tours and hours spent cycling together throughout Slovenia, which made difficult times more bearable. Furthermore, I would like to thank my colleagues that provided me with professional and personal support: Dragi Kocev, Aleksandra Rashkovska Koceva, Panče Panov, Ivica Slavkov and Dragana Miljković. Thank you also to my colleagues with whom I closely shared the wonderful experience of being a PhD student: Nikola Simidjievski, Vladimir Kuzmanovski and Nejc Trdin. I am happy to call all of you my friends.

I am most grateful to my closest family for their encouragement and support. Especially, I wish to express my gratitude and thanks to my mother Marija Tanevska. None of this would have been possible without her unconditional love and persistent confidence in me. Finally, I am grateful to my dear Marija Crvenkovska for her love, for being there every step of the way and for sharing all things good and bad with me.

Abstract

The work presented in this thesis is situated at the intersection of three research areas: systems and synthetic biology, mathematical modeling and machine learning. Systems biology is concerned with understanding the principles of life and the emergence of its complexity from a systems-level perspective. Synthetic biology is the application of knowledge learned from systems biology for the purpose of design of novel living systems that behave in an expected and beneficial manner. Mathematical modeling is an important aspect of systems and synthetic biology. It is an established approach to acquiring knowledge about the structure, function and behavior of dynamical systems. Process-based modeling is a machine learning approach to computational scientific discovery that constructs complete mathematical models of dynamical systems from knowledge and data.

The major limitations of process-based modeling are the coarse and deterministic formal representation of models and domain knowledge, the limited use of knowledge for model selection and the confinement to completely data-driven scenarios. In this work, we overcome these limitations of process-based modeling approaches and make them more suitable for application to tasks of modeling and design of dynamical biological systems. In particular, we extend process-based modeling towards representing and learning of stochastic models, the use of complex and domain-specific model selection criteria and, finally, towards process-based design.

We present an improved formalism for process-based modeling, which uses a finer grained representation of models and domain specific modeling knowledge. This formalism represents interactions, i.e., processes as sets of reaction equations. The new formalism allows for both deterministic and stochastic interpretation of models and modeling knowledge.

We address the problem of model selection by adapting regularization to the specifics of process-based modeling. We further propose to strengthen the evaluation bias of the learning process by introducing domain-specific criteria and demonstrate that this outperforms standard approaches to model selection in scenarios with limited observability.

We extend the scope of process-based modeling towards the task of designing dynamical biological systems that can achieve a desired behavior. We relax the dependence of the approach to data, make a shift towards completely knowledge-driven learning and take advantage of methods for simultaneous optimization of multiple problem-specific objective functions. This allows for process-based design of novel biological systems with desired properties and behaviors.

The presented improvements are evaluated on multiple tasks of modeling and design from the domains of systems and synthetic biology. The evaluation shows that the process-based approach can successfully reconstruct, in an automated manner, results from the literature obtained by manual or related computational approaches. Our approach performs well on both synthetic and real world problems of modeling and design of deterministic and stochastic dynamical systems.

Using the new formalism, our process-based modeling approach was successfully applied to the reconstruction of the structure and dynamics of gene regulatory networks with global and local kinetic rates and the learning of compartmental epidemiological models for the Eyam plague outbreak and the Tristan da Cunha influenza outbreak. The strengthening of the evaluation bias by domain-specific criteria was applied to solve the model selection problem in the task of modeling the dynamics of the Rab5-Rab7 switch in endocytosis. Finally, our process-based approach was also successfully evaluated by applying it to two design tasks of constructing a stochastic toggle switch without cooperativity and a deterministic oscillator.

Povzetek

Delo, ki je predstavljeno v tej tezi, se umešča na stičišče treh raziskovalnih področij: sistemske in sintezne biologije, matematičnega modeliranja in strojnega učenja. Sistemska biologija se ukvarja z razumevanjem načel življenja in z nastankom njihove kompleksnosti na ravni sistema. Sintezna biologija uporablja znanje, pridobljeno iz sistemske biologije, za snovanje novih živih sistemov, ki se obnašajo v skladu z našimi napovedmi in nam koristijo. Matematično modeliranje je pomemben aspekt sistemske in sintezne biologije. To je dobro uveljavljen postopek pridobivanja znanja o strukturi, funkciji in obnašanju dinamičnega sistema. Procesno modeliranje je eden od načinov, kako lahko s strojnim učenjem pridemo do znanstvenih odkritij, saj lahko s pomočjo podatkov in znanja avtomatično zgradi popoln matematični model dinamičnega sistema.

Največje omejitve procesnega modeliranja so grobi in deterministični formalni opisi modelov in znanja o domeni, omejena uporaba znanja pri izbiranju modela in uporaba postopkov, ki so povsem podatkovno vodeni. V tem delu presežemo omenjene omejitve procesnega modeliranja, s čimer tak pristop postane primernejši za uporabo pri modeliranju in načrtovanju bioloških dinamičnih sistemov. Natančneje, v procesno modeliranje vpeljemo stohastične modele, kompleksne kriterije izbiranja modelov odvisne od domene, in procesno načrtovanje dinamičnih sistemov.

Predlagamo izboljšan formalizem procesnega modeliranja s podrobnejšo predstavitvijo modelov in znanja o modeliranju, ki je odvisno od domene. S takim formalizmom opišemo procese oziroma interakcije kot množice reakcijskih enačb, ki opisujejo te interakcije. Novi formalizem dopušča tako deterministično kot stohastično razlago modelov in znanja o modeliranju.

Nalogo izbiranja modelov obravnavamo z regularizacijo, ki je prilagojena posebnostim procesnega modeliranja. Nadalje predstavimo domensko odvisne kriterije, ki krepijo pristranskost evaluacije učnega procesa. Pokažemo, da opisani pristop preseže standardne načine izbiranja modelov v primeru omejenih možnosti opazovanja dinamičnega sistema.

Povečamo domet procesnega modeliranja, ki ga zdaj lahko uporabimo za načrtovanje bioloških dinamičnih sistemov, ki se obnašajo po naših željah. Odpravimo odvisnost procesnega modeliranja od podatkov in naredimo korak proti učenju, ki ga vodi zgolj znanje, pri čemer izkoristimo metode za sočasno optimizacijo več funkcij koristnosti odvisnih od naloge. S tem omogočimo procesno načrtovanje novih bioloških sistemov z želenimi lastnostmi in obnašanjem.

Predstavljene izboljšave so bile preizkušene na več nalogah modeliranja in načrtovanja iz domen sistemske in sintezne biologije. Preizkusi pokažejo, da lahko s procesnim pristopom pridemo po avtomatizirani poti do enakih rezultatov, kot jih srečamo v literaturi in so bili pridobljeni ročno ali s sorodnimi računalniškimi pristopi. Naš pristop se dobro odreže tako pri sintetičnih kot tudi pri resničnih problemih modeliranja in načrtovanja deterministčnih in stohastičnih dinamičnih sistemov.

Svoj pristop k procesnemu modeliranju z izboljšanim formalizmom uspešno uporabimo pri razpoznavanju strukture in dinamike genskih regulatornih mrež z globalnimi in lokalnimi reakcijskimi hitrosti ter pri učenju kompartmentalnih epidemioloških modelov za izbruha kuge v Eyamu in gripe na otoku Tristan da Cunha. Okrepljena pristranskost pri evalvaciji s pomočjo domensko odvisnih kriterijev je bila uporabljena pri izbiranju modela za dinamiko preklopa Rab5-Rab7 v endocitozi. Naš pristop je bil uspešno preizkušen tudi na dveh nalogah načrtovanja, kjer je bilo treba konstruirati stohastično preklopno stikalo brez kooperativnosti in deterministični oscilator.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The work presented in this thesis lies at the intersection of mathematical modeling, machine learning and biology. The presented methodology for modeling and design is based on systems thinking. It is motivated by the ideas of understanding highly complex and dynamical phenomena, by analysis of the relationships and interactions of constituent parts of a complex system, using the systems approach (von Bertalanffy, 1968). The methodology is implemented as a machine learning (Mitchell, 1997) algorithm, i.e., the computational simulation of the empirical scientific process of modeling as a tool for construction of descriptive, representational models of physical systems (Langley, Simon, Bradshaw, & Zytkow, 1992), intended for better understanding of the nature of the relationships and interactions in them. We apply the proposed methodology in the domain of biology, more precisely, within systems and synthetic biology, primarily at the molecular level.

1.1 Background

1.1.1 Systems biology

Systems biology takes a systems-theoretic approach to the study of biological systems. In contrast to the traditional reductionist approach, systems biology goes beyond the study of individual parts and towards system-level understanding of complex systems using a holistic/emergentistic approach to biology.

The main approach to achieving a system-level understanding of biological processes is the transfer of ideas and approaches from systems theory and systems engineering to applications in biology (Wolkenhauer, 2001). Systems biology as a discipline within this frame of reasoning can be considered to be formally established in the 1960s (Mesarović, 1968). However, modeling efforts and perspectives that can be considered as systems approaches to biology date back to the 1950s. Notable examples are the development and analysis of a dynamical model of neuronal action potential by Hodgkin and Huxley (1952), for which they were awarded the Nobel Prize in Physiology or Medicine and the discovery of feedback inhibition of amino acid biosynthetic pathways (Umbarger & Brown, 1957; Yates & Pardee, 1957).

The rise to prominence and the subsequent rapid expansion of systems biology begins with the 21st century, due to the great success of the Human Genome Project and the development of a critical mass of high-throughput technologies in molecular biology, enabling the generation of massive amounts of biological data (briefly summarized in the historical perspective by Westerhoff and Palsson (2004)). The large amount of generated data and the subsequent shift of focus within the community, from the analysis of single constituents of a biological system towards a more general and integrative system-level view

of the dynamics and the emergent properties of biological systems, motivated the outreach to and the tighter integration with the fields of formal systems analysis, information theory and computer science. This is evident from the accounts from this period by Kitano (2000, 2002b) and Ideker, Galitski, and Hood (2001). The positive feedback between these fields led to the rapid development of computational tools and their use for formalization and exchange of knowledge, computational modeling, simulation and analysis of models, knowledge discovery, and data analysis (Kitano, 2002a).

1.1.2 Modeling dynamics in biology

From the earliest works in systems biology on, modeling as a tool for systems-level understanding of complex systems, has been, and still is, paramount to the field and its further development. Within the scope of this work, a model is assumed to provide an exact mathematical description of a dynamical system. We refer to the process of modeling of a dynamical system, as descriptive scientific modeling.

A mathematical model provides insight into the constituents of the system and the network of interactions between them (the structure) that give rise to the dynamical behavior over time and under different conditions. At the lowest level of abstraction, which allows for adequate interpretation (simulation of the dynamical system), the behavior of the system is represented by the changes of its state as a function of time, which is often nonlinear and complex. Mathematically this dynamical behavior is most appropriately captured by differential equations.

Traditionally, the model is represented a set of coupled ordinary differential equations (ODEs), describing continuous and deterministic system evolution. An alternative representation of models uses a set of coupled stochastic differential equations (SDEs), explicitly modeling the various sources of fluctuations, or more generally, taking into account the discrete nature of the state of the system. Another alternative is the representation that uses reaction equations with stochastic kinetic rates.

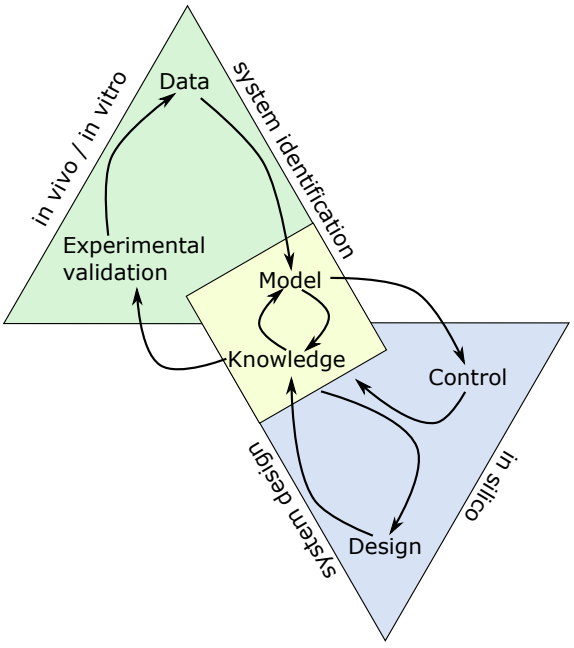

Figure 1.1: The cycle of descriptive modeling, control and design of dynamical systems in biology.

The descriptive modeling approach is central to the discovery of knowledge about the structure, function and behavior of a complex dynamical system. The structure of a complex dynamical system, most often, cannot be directly and completely observed. It is, however, somewhat easier to observe and measure the behavior of the dynamical system.

Given measured behavior, the descriptive modeling approach, referred to as system identification (Ljung, 1999) and depicted in the upper part of Figure 1.1, is an efficient tool for the discovery and validation of knowledge about an observed dynamical system. The induction of knowledge follows the next steps: First, postulate hypotheses about the structure of the system and the functional form of the interactions within the structure, i.e., *models*. Second, test each model against the measured *data*. Finally, select as best the model that most adequately explains the measurements. The *knowledge* about the system captured by a model is subject to *experimental validation* which can be performed in vivo or in vitro. The obtained knowledge can be further used to guide the design of new experiments that result in new observations and measurements, which can be subsequently used to revise existing or generate new knowledge/models.

The insights gained through the identification loop are crucial for the application of *control* mechanisms to the system. The identification of the structure of the system and the specific form of its interactions allows for targeting specific system properties towards the goal of achieving control over its behavior. The construction of appropriate controllers for the control loop is thus another mechanism for the revision and generation of knowledge. The application of system identification and control approaches within systems biology has resulted in an improved understanding of the basic functional building blocks and design principles of dynamic biological systems (Alon, 2007; Tyson & Novak, 2010).

1.1.3 Synthetic biology

The construction (*design*) of novel systems that exhibit desired behavior, through the application of knowledge gained from the system-level understanding of biological systems (via the modeling and control loops), is the goal of synthetic biology. While the focus of systems biology is on the problem of identification of dynamical systems, in synthetic biology the focus is shifted towards the design of dynamical systems (lower part of Figure 1.1).

Synthetic biology develops in parallel to the expansion of systems biology (Cameron, Bashor, & Collins, 2014). Important first milestone applications of synthetic biology are the development of the first synthetic toggle switch (Gardner, Cantor, & Collins, 2000) and the repressilator (Elowitz & Leibler, 2000). These applications triggered a period of development and analysis of parts (basic elements) and modules which are composed of parts and have specific functions. This, in turn, led to many efforts aimed at combining parts and modules, towards real-world applications, resulting in one of the major achievements of this period: the development of a registry of standard parts (Canton, Labno, & Endy, 2008). However, the predominantly manual (wet-lab) development of these parts by using the available genetic engineering techniques (Carr & Church, 2009; Gaj, Gersbach, & Barbas III, 2013), although achievable, is still a tedious and costly task. Further development, towards the combination and integration of these parts into more complex system-level circuits (named the second wave of synthetic biology (Purnick & Weiss, 2009)), is therefore conditioned by the move towards integrating the in-silico loop within the process, i.e., the development and use of computational modeling approaches (Marguet, Balagadde, Tan, & You, 2007; Kaznessis, 2007).

From a more recent perspective, as evident from the targeted applications of systems and synthetic biology, especially within biotechnology and biomedicine (Khalil & Collins, 2010; Wolkenhauer, 2014), the task of modeling for knowledge discovery becomes more demanding. The manual approach does not scale well with the current developments and

needs, given the increasing complexity of the systems being considered and the need for model refinement in light of increased availability of data. Furthermore, synthetic biology is witnessing rapid development of novel technologies for genome engineering and editing (e.g., the recent identification and exploitation of the CRISPR/Cas system (Jinek et al., 2012; Cong et al., 2013; Doudna & Charpentier, 2014)). This leads to an increase in the number and the complexity of designs that can achieve a desired behavior for the targeted application (Kelwick, MacDonald, Webb, & Freemont, 2014), and need to be considered and implemented for a given synthetic biology task.

The use of computational approaches would allow for many designs (models) to be considered in-silico, while only a few selected candidate designs would make it to the wet-lab stage.

1.1.4 Machine learning

Recent computational approaches to knowledge discovery automate the task of modeling dynamical systems and bear a lot of potential for further advancement in the areas of systems and synthetic biology. The process of knowledge discovery through descriptive modeling, as previously described, can be cast into a learning task: Learn a model of a dynamical system that adequately explains the observations of the behavior of the system. This exactly matches the general paradigm of machine learning. Within computer science, machine learning explores the development of algorithms that solve problems by learning, i.e. algorithms that improve their measurable performance with experience (Mitchell, 1997). Within machine learning, the algorithms that address the induction of functions from observations belong to the paradigm of supervised learning. The task of modeling from observations can be cast into a supervised learning task. In particular, this task involves learning the structure (function) and the parameters of a model, guided (supervised) by the observed outcome of the model in the form of numerical time-series.

However, the modeling of dynamical systems is a special form of supervised learning, since it does not fit the standard framework for two reasons. First, the observations of a system in machine learning are assumed to be independent and identically distributed point observations of system properties or behavior (examples) clearly labeled with the expected value of a conditional variable (target property or behavior). When considering dynamical systems, this general independence assumption breaks down. The state of a dynamical system changes as a function of time, dependent on the previous state(s) of the system. Its observations therefore come in the form of time-series of measurements.

Second, in machine learning, the classes of functions that are most commonly considered correspond to broad families of functions, such as the family of linear or non-linear functions, the family of logic formulas (rules/trees) or the family of discrete or continuous probability functions (Hastie, Tibshirani, & Friedman, 2009). These broad families of functions are preferred due to their ability to easily capture common patterns in observations from different domains, and their relatively low cost of training (optimization) and evaluation. When considering the task of modeling dynamical biological systems, these general classes of functions are not well suited. It is important to take into account that the processes within the dynamical biological systems are intrinsically and physically guided by a very specific form of interactions. Furthermore, these interactions that comprise the system (and represent its structure), and their explicit functional forms (which guide the dynamics of the system) need to establish a formal basis of knowledge about the system and as such, they need to be easily understandable, interpretable and communicable by and to domain experts.

Therefore, machine learning approaches to modeling dynamics rely on methods for computational scientific discovery approaches (Langley et al., 1992). They are designed

to allow for learning from both knowledge and time-series of observations. The use of task specific knowledge provides a way to induce understandable and communicable models. Computational approaches to scientific discovery follow closely (mimic) and assist, by automation, the manual empirical approach to modeling dynamical systems. These approaches are based on the simulation of human cognitive processes dealing with problem solving (Newell & Simon, 1972). Two basic concepts are at the foundation of problem solving, upon which all further approaches build upon: The formal representation of the problem space and search for the problem solution within this space as a selective process guided by heuristics, as opposed to random trial and error.

Approaches to computational scientific discovery have been used in numerous domains for diverse problems (Džeroski & Todorovski, 2007). Historically, they have been applied to rediscovery of laws in physics, chemistry, and ecology from observations. Later they have been applied to model revision and discovery of new knowledge. Most recently, process-based modeling (Džeroski & Todorovski, 2002; Todorovski, Bridewell, Shiran, & Langley, 2005; Bridewell, Langley, Todorovski, & Džeroski, 2008; Čerepnalkoski, Taškova, Todorovski, Atanasova, & Džeroski, 2012; Simidjievski, Todorovski, & Džeroski, 2016) has emerged as a quantitative computational scientific discovery approach specifically tailored to modeling dynamical systems from knowledge and data. It has been successfully applied to several case studies of reconstructing biochemical reaction networks (Džeroski & Todorovski, 2008), modeling aquatic ecosystems (Čerepnalkoski et al., 2012; Simidjievski, Todorovski, & Džeroski, 2015) and watershed modeling (Škerjanec, Atanasova, Čerepnalkoski, Džeroski, & Kompare, 2014).

1.2 Motivation

When modeling the dynamics of biological systems we are faced with several sources of uncertainty. Regardless of whether the modeling goal is to explain a set of observations, generate new insights, revise existing beliefs or achieve expected behavior, the uncertainty may be fundamental and come from the (lack of) knowledge, i.e., the level of understanding of the structure and the dynamical behavior of the biological system. The two fundamental types of uncertainty in modeling dynamical systems concern the structure of the model and its parameters.

Another source of uncertainty comes from the intrinsic properties of the system, such as the stochastic nature of reactions at the molecular level. Since the modeling task is usually guided by the available observations, further sources of uncertainty may come from the experimental procedure used to obtain them. These may include noise due to variability of external experimental conditions, measurement noise and limited observability of the properties of the model. All of these sources of uncertainty have to be taken into account during the modeling process in order to produce valid models: This makes modeling a hard task.

In the following, we briefly present the different formalisms for representing models and the fundamental uncertainties in model structure and parameters together with existing approaches to resolving these uncertainties. We identify the limitations of the existing approaches. We further identify the process-based modeling approaches as a good candidate for further development, aimed at addressing these limitations. A more detailed overview of these formalisms and approaches, including process-based modeling, is given in Chapter 2.

1.2.1 Representing uncertainty

There is a lack of flexible formalisms for capturing the available knowledge and uncertainties which at the same time are general enough to allow for different interpretations of the encoded models. The process-based modeling formalism is an example of a formalism that allows for representing both uncertainty in the structure and in the parameter values of a model. As such, it represents a good candidate for further development, mainly in the direction of encoding stochastic models.

A mathematical description of a model of a dynamical system takes the form of a system of coupled differential equations $\frac{dx_i}{dt} = f_i(x, \theta_i, t)$, where x represents the set of system variables(i.e., variable properties of the constituent entities of the dynamical system) at time t, functions f describe the structure of the system in terms of its processes, and θ is a set of constant parameters. The formal representation of the structure of a model defines the assumptions about the nature of the interactions between constituents, capturing the intrinsic properties of the system. The fundamental structural uncertainty can be declaratively represented by hypotheses about the constituent entities, their properties and the interactions between them in the system being modeled. The formal representation of structural uncertainty requires means for defining the different possible forms of functions f_i that reflect the available knowledge about the system being modeled. The larger the space of possibilities for f, the larger the structural uncertainty.

The model parameters additionally define the specific interactions within the model. Their values are also related to the model behavior. The second fundamental uncertainty is therefore related to the values of the constant parameters in the models. The formal representation of uncertainty in the parameter values requires means for defining the possible range of values for each parameter ($a_i^j \leq \theta_i^j \leq b_i^j$, where $a_i^j, b_i^j \in \mathbb{R}, j = 1..|\theta_i|$). The wider the range, the larger the uncertainty. In order to best capture the available knowledge, a formalism for representing models of dynamical systems in biology should be able to capture the fundamental uncertainty in model structure and parameter values.

An important property of a formalism is also its ability to represent the effect of the stochasticity of the model interactions that are part of its structure. For example, to allow for including the appropriate terms within f_i that approximate the various sources of intrinsic or extrinsic noise. Even further, at the intracellular level, the biochemical reactions can be considered to be a result of random collisions of a discrete number of constituents. The stochastic fluctuations in these systems are responsible for the emergence of specific phenotypes or genetic activities on one hand, or divergence of behavior on the other (McAdams & Arkin, 1997; Arkin, Ross, & McAdams, 1998; Samoilov, Plyasunov, & Arkin, 2005). Deterministic representations fail to account for the underlying stochasticity of biological systems, especially if the modeled system is observed at the molecular level and contains only a small number of molecules.

At the level of representation of models with fixed structure and parameter values, the most widely used formalisms are differential equations and the more abstract representation of reaction equations. However, the low-level representation using a system of coupled differential equations or a set of reaction equations cannot be used to directly and explicitly represent structural uncertainty nor uncertainty in parameter values. Furthermore, the direct representation using these mathematical formalisms is often rigid and complex, and fails to communicate the common structural patterns within a dynamical system.

Many higher-level formalisms have been proposed focusing on different aspects of the structure of the modeled system, capturing different properties of the system (Machado et al., 2011; Bartocci & Lió, 2016). They are usually complementary to the mathematical representation using differential equations. They aim at alleviating the process of modeling complex systems by imposing a more intuitive, understandable and constructive approach.

The formalisms used for systems biology applications focus primarily on the optimal representation of a single model with a fixed structure. For example, a number of formalisms have emerged that deal with the need for compact representation arising from the complexity of the space of the combinations of elementary interactions of a similar form between entities observed in a given biological system. These formalisms allow for specifying rules (constraints) that limit the space of potential interactions between entities on the basis of their properties. There exist several classes of these formalisms, most notably the class of rule-based modeling formalisms (Feret, Danos, Krivine, Harmer, & Fontana, 2009; Faeder, Blinov, & Hlavacek, 2009) and process algebras (Priami & Quaglia, 2004; Ciocchetta & Hillston, 2009). Note that the encoded constraints do not address the issue of structural uncertainty in any way.

For synthetic biology applications, a number of formalisms allow for the formal specification of rules or constraints imposed on the space of possible model components and their composition (Pedersen & Phillips, 2009; Bilitchenko et al., 2011). The model components considered correspond to standardized biological parts with specific behaviors, such as those defined in the registry of standard parts (Canton et al., 2008). Although combinations of different parts can be explored in order to satisfy the constraints, the composition-oriented formalisms do not allow for explicit specification of structural uncertainty. Additionally, these parts have fixed interaction properties and parameter values.

Even though the process of descriptive modeling of dynamical systems requires the generation and testing of multiple competing structural hypotheses, only few computational approaches formalize and systematically address this issue for systems and synthetic biology applications. Some approaches come from the area of network reconstruction where the structural uncertainty is defined by the probability of presence of edges (interactions) between nodes (constituents) in a network (Bansal, Belcastro, Ambesi-Impiombato, & di Bernardo, 2006; Penfold & Wild, 2011). Other, optimization-based approaches, encode the structural uncertainty in the form of uncertainty in the parameter values of a more general model (Rodrigo, Carrera, & Jaramillo, 2007; Dasika & Maranas, 2008; Marchisio & Stelling, 2009; Sendin, Exler, & Banga, 2010) or in the form of a single integer parameter that represents different manually enumerated structural hypotheses (Toni, Welch, Strelkowa, Ipsen, & Stumpf, 2009; Barnes, Silk, Sheng, & Stumpf, 2011). It is worth noting that these approaches, in addition to the ability to consider multiple structural hypotheses, allow for the definition of uncertainty in the parameter values. However, only a small number of them are flexible enough to allow for both deterministic and stochastic interpretation of the models.

Process-based modeling formalisms use domain specific knowledge to represent constituent entities and interactions between them (processes). The process-based formalisms are complete, in the sense that they can efficiently encode uncertainty in both model structure and parameters. The formalism used by process-based approaches is based on the notion that knowledge of a specific domain of application can be specified in the form of templates of system constituents and interactions, clearly defined by the domain, which can be further organized in taxonomies. ProBMoT (Process-Based Modeling Tool), developed by Čerepnalkoski (2013), is the most recent contribution to the area of process-based modeling. Its formalism allows for the introduction of incompletely specified process-based models, also based on domain knowledge. The incomplete models are a way to define the space of possible candidate model structures. They impose additional constraints on the uncertainty in the structure and the numerical parameters encoded in a general library of domain knowledge.

However, the existing process-based formalisms are currently limited to only deterministic interpretation of the dynamics of the modeled system. The description of the

interactions between the system constituents within the process-based formalism relies on the coarse representation using fragments of ordinary differential equations. A finer grade representation of processes is a desirable feature. It can broaden the possibilities of interpretation in the direction of capturing the inherent stochasticity of dynamical systems in biology.

1.2.2 Resolving uncertainty

Process-based modeling (PBM) offers an approach to identifying the structure and the parameter values of a formally represented model. However, the existing PBM approaches are data-driven. Additionally, they rely on a single goodness-of-fit criterion to resolve the uncertainty in the structure and the parameter values. The process of resolving the uncertainty has to be guided by a well defined modeling goal. As such, it should take into account all the sources of knowledge and uncertainty in a most efficient manner.

At the level of structural uncertainty, if a formal representation thereof is available, it can be resolved by automated enumeration or search, guided by problem-specific constraints imposed on the formally defined general knowledge. By automation, this process can be easily scaled in terms of the number of considered candidate structures.

Composition-oriented approaches do not require an explicit representation of uncertainty. Instead, an objective of the task of resolving the uncertainty is defined based on the expected input-output behavior of the system. The approaches then infer a series of intermediary logic gates and connections, necessary to produce the desired behavior. The resulting logical circuit is then transformed into a model by using rule-based or composition-oriented formalisms, which produce a valid composition of well-characterized parts from a library.

The class of approaches that cast the problem of resolving structural uncertainty into a task of network reconstruction are primarily concerned with the reconstruction of interactions based on the statistically significant dependence between observations of different constituents. They transform each candidate model structure into a system of differential equations. However, the type of interactions between the constituents is considered to have a single predetermined form.

The class of optimization-based approaches are able to consider a broader range of objectives for resolution of the structural uncertainty. The objectives can consider the goodness of fit of the model to the available observation, but also more general properties of the behavior of the model. The optimization-based approaches cast models into the formalism of ordinary differential equations or into a set of reaction equations. They reduce the structural uncertainty to the level of uncertainty in the parameter values and further solve the problem by applying parameter estimation.

If there is no formal representation of the structural uncertainty, the space of structural hypotheses can be also defined manually by enumeration of specific candidate model structures. The set of structures can also be a result of the continuous manual revision of a single candidate structure through trial and error. The manual enumeration or revision are tedious tasks, which can be performed for a relatively small number of diverse candidate model structures.

At the level of parametric uncertainty, a model cast into a system of differential equations or reaction equations requires the determination of the constant parameter values (parameter estimation). The parameters are estimated by using a chosen objective function(s), which guides the optimization process towards predefined modeling goals. Many different optimization methods have been used for parameter estimation in systems biology (Moles, Mendes, & Banga, 2003; Sun, Garibaldi, & Hodgman, 2012). In systems biology, the objective function used for parameter optimization is based on the comparison of the

observed time-course data to the simulations produced by the model structure with a set of candidate parameter values. The aim of parameter estimation is then to minimize the difference between the two. In contrast to systems biology, where observed behaviors are available and used in the optimization process, in synthetic biology the desired behaviors are not directly observed, but are rather specified by using a criterion or a set of criteria that the desired behavior has to satisfy (Rodrigo et al., 2007; Dasika & Maranas, 2008; Barnes et al., 2011; Silk et al., 2011).

From a machine learning point of view, the task of resolution of uncertainty can be solved by using a two-level learning approach. The resolution of uncertainty relies on the formally encoded knowledge and the measured data (or a desired behavior), which are used to refine the space of structure hypotheses into candidate model structures at the first level. The optimization of the model parameter values for the candidate model structures takes place at the second level. At the end of the learning process, a model is selected from the list of candidates that most adequately fits the designated modeling goals (optimization objectives).

The learning strategy of PBM approaches is search within a well defined model space constrained by domain knowledge, coupled with numerical optimization guided by a single objective function. The objective function is based on goodness-of-fit to observations of the modeled system, represented by the sum of squared errors. This objective function does not always reflect the goal of the modeling task, which can result in a hard model selection problem. This is especially true for the task of design of dynamical systems where this objective function cannot be used (due to the lack of observed data). Observations should be replaced by a description of the criteria that need to be satisfied in order for the desired behavior to be achieved. Note that a suitable design might have to simultaneously fulfill multiple design objectives (expected properties of behavior), which, in general, can be independent or even conflicting. A multi-objective optimization approach guided by goal specific criteria would be the most general and adequate approach.

1.2.3 Model selection

The process of resolving uncertainty and its existing limitations (in general, and in PBM in particular) is directly related to the ensuing problem of model selection. Within PBM, especially in the most recent approaches, the problem of model selection has received little attention. Model selection can be performed to choose the most suitable model based on different available criteria (Cedersund & Roll, 2009; Kirk, Thorne, & Stumpf, 2013). The selection of the most appropriate model within PBM is performed only on the basis of the achieved optimized performance of the candidate models obtained during the learning process. This approach to model selection, especially when considering structurally diverse candidate models, inevitably leads to over-fitted models.

A classical approach to over-fitting avoidance in machine learning and statistics is the selection of a model with the least generalization error. If we consider the most common tasks in machine learning, this approach requires an abundance of observations coming from a large number of experiments. These observations can be used to both learn a model and test its predictive performance. In a train-test type of approach, the models are learned (trained) using a subset of observations. The power of generalization is then measured (tested) by the error over observations that have not been used in the learning process. The smaller the error, the better the model captures the relevant information within the observations, and the better the generalization power of the model. However, this kind of task is rarely considered within the domains of system and in synthetic biology, usually due to the lack of quality observations, the price of repeating and maintaining experiments for a longer time, or both.

What remains as an important source of information for descriptive modeling tasks, that can be used to approximate the generalization power of a model, is captured within the bias introduced in the learning process. Formally, Mitchell (1997) defines bias as "any basis for choosing one generalization over another, other than strict consistency with instances". There exist two types of bias: representational (language) bias and evaluation (procedural) bias. The specific functional form of the interactions within the system, the uncertainty in both structure and parameter values, and the constraints over the model space that the uncertainties define for a specific instance of a problem, correspond to the representational bias. The measurements used for the evaluation of the suitability of a certain candidate model, given observations or a target behavior, correspond to the evaluation bias.

Ideally, the knowledge encoded in both types of bias during the learning process can be used to compensate for the lack of complete observations, or for observations with lower-grade quality, and guide the model selection process. It is typical for model selection to make use of additional knowledge about the desired properties of the structure in conjunction with the fit of the model in order to strengthen the evaluation bias and improve the estimate of the generalization error of a candidate model. This approach to model selection is typically based on the parsimony principle. Generally, it is based on regularization of a goodness-of-fit or a likelihood function, such as the Akaike information criterion (AIC) (Akaike, 1974), the Bayesian or Schwarz information criterion (BIC) (Schwarz, 1978) or Minimum Description Length (MDL) (Grünwald, 2007). However, the efficiency of model selection (using these or related methods) is proportional to the amount of relevant assumptions and information about the model structure and its behavior captured by the biases.

While the process-based modeling formalisms can be efficiently used to encode domain knowledge, i.e., introduce the most relevant representational bias to the learning process, it does not make use of the complete information in the process of optimization and model selection. The evaluation of the performance of each candidate model is limited to the single goodness of fit function of its simulation and the observations used for parameter estimation. Therefore, the introduced evaluation bias is weak. The first step towards an improved model selection is the strengthening of bias, following the parsimony principle (Tanevski, Todorovski, Kalaidzidis, & Džeroski, 2013). The next step is the strengthening of the evaluation bias by the introduction of problem specific criteria related to the desired properties of the simulated behavior of the system. In order to estimate the complete structure and parameter values of a model of a complex biological system, one criterion in the form of an error function is not sufficient. This is especially the case for applications in the domain of synthetic biology, where the modeling task is completely knowledge-driven.

1.2.4 Synthesis

The area of computational biology lacks a unified approach to automated modeling and design of dynamical systems. We thus turn to machine learning approaches to computational scientific discovery, i.e., process-based modeling. Even though the current PBM approaches can be used to encode and resolve uncertainty in both structure and parameters in an automated manner, they are limited in their applicability to the domains of systems and synthetic biology. We have identified the following issues that need to be addressed:

- The PBM formalisms are coarse and limited to representing deterministic models. A good process-based formalism should be close to the basic mathematical formulation, and easily understandable by biologists/modeling experts, on one hand, and allow for both deterministic and stochastic interpretation, on the other hand.

- Model selection within PBM is an important problem that has received limited attention. The model selection problem in PBM stems from the explicit representation of structural uncertainty. Known approaches to model selection should be adapted and applied to the specifics of the PBM task. Furthermore, the use of domain knowledge in the different stages of the modeling process can be used to strengthen the bias and alleviate the model selection problem.

- PBM approaches are completely data-driven and are not suitable for design applications. Based on the domain-specific strengthening of the evaluation bias, the shift from a completely data-driven towards a predominantly knowledge-driven approach can lead to process-based design of dynamical biological systems.

The purpose of the dissertation is the development of a complete approach to automated process-based modeling and design of dynamical systems in systems and synthetic biology by overcoming the limitations of the existing approaches.

1.3 Goals

The goals of the dissertation relate to the specifics of the design, implementation and evaluation of the process-based approach to the automated construction of deterministic and stochastic models in systems and synthetic biology:

- *Improved process-based formalism.* The first goal is concerned with the design of an improved formalism for process-based modeling that extends the current formalisms towards the modeling and interpretation of the stochastic, reaction-based nature of the dynamical systems in the targeted application domains. This can be achieved by extending the existing PBM formalism to support reaction equations. The interactions between the entities in the system will remain in the form of template processes, but will be extended to contain biochemical reactions. The reactions will represent the discrete change of the properties of the constituent entities, parameterized with a stochastic rate.

- *Model selection and bias strengthening.* The second goal is related to the problem of model selection for process-based modeling. The model selection based on the goodness-of-fit of each candidate leads to selection of overfitted models. Model selection can be approached with standard regularization methods and will be further improved by strengthening the evaluation bias using information about the structure and the simulated behavior of the model.

- *Process-based design.* The third goal is to extend PBM approaches to completely enable knowledge-driven modeling. We will develop an approach that does not require measured data, but is in turn based on objective functions that describe a desired behavior, fulfilling the crucial precondition for the design of novel dynamical systems. Depending on the complexity of the modeled system, the implementation of a simultaneous optimization of multiple objective functions is required in order to target specific properties of the behavior of a dynamical system. Subsequently, the selection of a design has to be performed on the basis of the information available from the multi-objective optimization of each candidate.

The improvements to the process-based approach to meet all of the aforementioned goals require the development of a software tool. The tool should be aimed at allowing biological experts to easily manipulate component combinations and develop explanatory

models of biological systems from measured data, as well as discover/design of (new) biological networks based on descriptions of their desired behavior. This can be achieved by the implementation of an internal stochastic representation of the process-based reaction models and adaptation of state-of-the art stochastic simulators to the improved representation; integration of heuristic optimization of the parameter values of candidate models based on adequate arbitrary objective functions for specific modeling tasks; and integration of multi-objective optimization algorithms within the developed tool.

Finally, the proof of concept can be achieved by evaluation. The software tool can be used to illustrate the utility and evaluate the performance of the developed approach. This requires: the encoding of process-based knowledge for various real and benchmark modeling problems in systems and synthetic biology by using the equation based and the newly developed reaction based formalism, the definition of modeling tasks based on performed experiments from the literature and benchmark modeling tasks; and the comparison of the results obtained by the process-based approach to the published results, in a real-world scenario, or to a previously chosen ground truth model in a benchmark scenario.

1.4 Contributions

The work presented in this thesis comprises several contributions to the areas of machine learning (process-based modeling), systems biology and synthetic biology. A complete list of publications related to this thesis is available in the back matter in the Bibliography section. Each one of the following main contributions of the thesis is related to one of the goals outlined above:

- *A process-based formalism, for representing models and domain specific modeling knowledge, that allows for both deterministic and stochastic interpretation*; an algorithm for learning models represented in this formalism and its implementation within a process-based modeling tool; its evaluation in terms of reconstruction of complete models of dynamical systems from both synthetic and real world data has been published in a journal article (Tanevski, Todorovski, & Džeroski, 2016a).

  We present an improved and extended higher-level formalism for process-based modeling and design of dynamic biological systems, addressing the limitations of previous formalisms. The major improvement comes from the representation and interpretation of the stochastic nature of processes, i.e., interactions among constituent entities in systems. The representation of process dynamics is based on reaction equations, a powerful and flexible formalism for relating the temporal evolution of the properties of reactant and product entities. The reaction-equation based representation is directly related to the basic understanding of the behavior of the constituent entities in an interaction from a biological perspective. In addition to allowing for a deterministic representation of the dynamical behavior (by reduction of process-based models to systems of ordinary differential equations), models represented in the formalism are reducible to two different mathematical representations that capture the stochasticity of a biological system, i.e., to a chemical master equation or to a system of stochastic differential equations.

- *An approach to model selection in PBM that combines domain-independent and domain-specific criteria* and its application to the domain of modeling the dynamics of the Rab5-Rab7 switch in endocytosis. This work has been presented at conferences (Tanevski et al., 2013; Tanevski, Todorovski, & Džeroski, 2013) and published in a journal article (Tanevski, Todorovski, Kalaidzidis, & Džeroski, 2015).

We demonstrate that the model selection issue, so far largely neglected within PBM, can be approached by regularization, i.e., by considering the complexity of the structure of a process-based model in addition to the goodness of fit criteria. Additionally, we show that the strengthening of the evaluation bias of the learning process by the introduction of domain-specific criteria outperforms the standard regularization based approach to model selection for PBM. To this end, we consider a real world problem of modeling the Rab5-Rab7 dynamics in endocytosis from noisy data and limited observability. We encode domain-knowledge within the evaluation criteria, in the form of objective functions that are not completely data-driven and accordingly adapt the heuristic parameter estimation.

- *A methodology for process-based design*, its implementation and evaluation on task of designing deterministic and stochastic dynamical biological systems. These have been presented in conference proceedings (Tanevski, Simidjievski, & Džeroski, 2012) and a journal submission (Tanevski, Todorovski, & Džeroski, 2016b).

  We develop a methodology for process-based design of dynamical biological systems by combining the reaction-equation based formalism and a multi-objective optimization approach. This approach allows for learning within the process-based modeling formalism by taking advantage of a completely data-free evaluation bias in combination with a domain-knowledge based representational bias. We have shown that it can be used to design novel dynamical systems that can achieve desired behaviors. Taking advantage of methods for parameter estimation by simultaneous optimization of multiple, potentially conflicting, problem specific objectives, we develop an approach to select among alternative designs. This approach uses the hyper-volume indicator of the overall (multi-objective) quality and the complexity of the designs.

1.5 Organization

This introductory chapter has provided an overview of the development and the basic concepts in the different subject areas at the intersection of which the topic of this thesis is situated. It has further identified the limitations of the existing approaches that have motivated our research on this topic, the goals we have set ourselves, related to overcoming these limitations, and the scientific contributions resulting from the achievement of these goals.

Chapter 2 gives an overview of the related work. Three different aspects of the related work are presented. First, we review the modeling formalisms that are commonly used for representing dynamical models in the areas of systems and synthetic biology. Second, we summarize the related methods that have been considered for inferring models of dynamical biological systems. Third, we look at the process-based modeling approaches that are closely related to the work presented in this thesis.

The following three chapters are dedicated to the three scientific contributions of the thesis. They describe the work performed towards the achievement of each of the above goals and the contribution resulting from the achievement of each goal.

Chapter 3 presents the formalism for stochastic process-based modeling of dynamical systems, which bridges an existing gap between the process-based modeling approaches and the standard approaches for modeling in systems biology. We present its use within a process-based modeling tool, which is evaluated on four tasks of reconstructing system structure and dynamics from synthetically generated and real world data. The tasks come from two domains: two from the domain of gene regulatory networks and two from the domain of epidemiology.

Chapter 4 presents the bias strengthening approach to model selection, which uses domain-specific criteria in addition to domain-independent ones. The evaluation of the approach is performed by focusing in more depth on the model selection problem using for the task of modeling the mechanism of Rab5 - Rab7 conversion switch during the maturation of the endosome. The difficulty of this specific model selection problem is due to the limited observability and presence of noise in the data which are frequent issues in real-world modeling tasks from the domain of systems biology. We consider and compare different model selection criteria for this task, both domain-independent and domain-specific, as well as combinations thereof.

Chapter 5 presents a methodology for process-based design of dynamical biological systems that brings together process-based modeling and multi-objective optimization. We describe the use of domain-specific knowledge for representing the uncertainty in the structure and parameters of candidate models/designs. We also use such knowledge for the description of different objectives that can be simultaneously considered for optimization and selection of models/designs that lead to the desired behavior. We illustrate the advantages of process-based design on the tasks of designing a stochastic toggle switch (without cooperativity) and a deterministic oscillator.

Finally, Chapter 6 concludes the dissertation. It sums up the performed research, presents a summary of the contributions, and outlines several directions for further work.

Chapter 2

Related Work

2.1 Modeling Formalisms for Systems and Synthetic Biology

A formalism is a well defined set of syntactic rules for representing entities and the interactions between them. Apart from the syntax, formalisms convey the means of interpretation of its representation. Various formalisms can represent the same biological model. The interpretation of a formally represented model of a dynamical system is a realization of its behavior trough time, often referred to as simulation. The modeling formalisms for systems and synthetic biology differ both in their syntax and semantics, due to the underlying assumptions about the physical properties of the biological system and the intended application of the models that they represent.

The first difference is at the level of abstraction of the interpretation of the formally represented model. The models can be qualitative or quantitative. Qualitative models represent systems and their dynamical behavior in an abstract and descriptive manner. Although they are most often built empirically, they do not require precise information about the numerical properties of constituents and the interaction between them. Quantitative models offer a more precise and detailed view of the dynamical system. They incorporate complete mechanistic and kinetic details of the dynamical system and precisely represent the physical and chemical processes underlying the interactions. Their use requires detailed knowledge about the system being modeled.

Another difference is in the properties of the state-space defined by the model. The state space is the valid space of values for the variable properties of the interacting entities. These properties can be represented and interpreted as discrete or continuous.

Dynamical systems change their state over time, so the interpretation of formally represented models can differ in the treatment of the temporal properties of the interactions between the entities. The interactions within the dynamical system can be considered as occurring in discrete time intervals or continuously.

Finally the formalisms differ by the interpretation of the determination of the state-space as a function of time. The state of a system can be interpreted deterministically, i.e., its temporal evolution is completely determined by the defined interactions and their properties. It can also be interpreted stochastically i.e., its evolution is influenced by stochastic interactions, probabilistically defined by their properties.

In the following subsections we make a brief overview of a selection of modeling formalisms frequently used in systems and synthetic biology in terms of the aforementioned differences and their application. More detailed overviews are given by Machado et al. (2011) and Bartocci and Lió (2016). We start by describing the important features of purely qualitative modeling formalisms such as Boolean networks, and continue towards the intersection of qualitative and quantitative formalisms, where we consider Petri nets

and process algebras with their various extensions. Finally, towards the quantitative end, we describe rule-based formalisms, reaction equations and differential equations.

It can be seen that the extensions of the initially qualitative formalisms aim at enabling quantitative generalizations. The quantitative representations are further extended to cover stochastic interactions. At the lowest level of abstraction, these generalizations result in the representation of a dynamical system in the form of Markov processes, which allow for numerical simulation (execution) under more realistic assumptions.

Quantitative modeling formalisms that enable fine grained representation of the dynamics of the interactions are preferred for synthetic biology applications. These applications require detailed understanding of the mechanisms of interaction within the dynamical system in order to make realistic predictions about the behavior of the design and generate potential candidates for wet lab realization. Finally, enabled by the effort to standardize biological parts, formal representations of synthetic biology constructs (designs) can be highly abstract. Each part contains specific details about its compatibility and characterization of its functional properties, therefore the formal representation of a dynamical system may consist only of parts and their composition, resulting in "composition-oriented" formalisms.

2.1.1 Boolean networks

Boolean networks a as formalism for modeling genetic regulatory networks has been introduced by Kauffman (1969). It since has been applied also for modeling different types of regulatory systems in biology. A Boolean model is a network representation of a dynamical system, where each constituent (entity) is a node in a network represented by a Boolean (1/0) state. For different entities, such as genes, proteins or stimuli, the state represents qualitative properties, such as expression/inexpression, concentration above/below threshold, modified state (ex. change due to phosphorylation/dephosphorylation), or presence/absence of stimulus. The state of each node is a function of the states of other nodes that are connected to it, constructed from basic logical operators, such as NOT, AND and OR, defining the interactions within the network.

The Boolean networks are qualitative and discrete in terms of the representation of state-space and time. Each state of a node can be updated at a discrete time interval, resulting in a trajectory that represents the dynamical behavior of the node through time. Although there exist modifications of Boolean networks that introduce a level of stochasticity (Akutsu, Miyano, & Kuhara, 2000; Shmulevich, Dougherty, Kim, & Zhang, 2002; Chaves, Albert, & Sontag, 2005), their interpretation is principally deterministic.

Boolean networks are appropriate for large scale systems with hundreds of entities and interactions between them. Most often they are considered as first step of modeling when there is an abundance of experimental data about a large set of entities obtained by high-throughput omics methods, but lack of detailed knowledge of possible mechanisms of interaction and kinetics. As such, they are appropriate for generation of hypotheses by data-driven network inference or for representing global system properties arising from the structure of the inferred networks. These properties can relate to the node centrality, clustering and modularity of the network, or its stability, e.g., steady-states (attractor states) (Jeong, Mason, Barabasi, & Oltvai, 2001; Basso et al., 2005; Klamt, Saez-Rodriguez, Lindquist, Simeoni, & Gilles, 2006; Saez-Rodriguez et al., 2009).

Although Boolean networks can be used for conceptual planning and system design in synthetic biology, their application is predominantly reserved for top-down modeling approaches in systems biology. An overview of Boolean modeling and its applications is given by Wang, Saadatpour, and Albert (2012).

2.1.2 Petri nets

Petri nets, named after their creator Carl Gustav Petri, are a graphical and mathematical formalism for representing concurrent systems (Petri, 1962).

A basic Petri net represents a system as a bipartite digraph consisting of two types of elements, places and transitions. Places correspond to interacting entities or variable physical quantities. Transitions correspond to interactions. Visually, places are represented by circles, transitions by squares. The state-space is represented by a discrete distribution of tokens across places in the graph. Tokens represent the current value of a variable represented by a place. The places are connected (place to transition, transition to place) to transitions by weighted arcs. The weights represent the number of tokens needed to cross the arc (transit, fire the reaction), and the number of transfered tokens as a result of the transit. For applications to modeling molecular biological systems, the weights of the arcs on the input and output of a transition correspond to the stoichiometry of the reactants and products. The transfers are assumed to be able to be executed concurrently, however in fixed time intervals.

The models represented by basic Petri nets have discrete state-space and time and their interpretation is deterministic. The application of models represented by basic Petri nets is therefore limited to qualitative analysis of the modeled system (Reddy, Liebman, & Mavrovouniotis, 1996; Küffner, Zimmer, & Lengauer, 2000; Zevedei-Oancea & Schuster, 2003). However, many extensions have been developed, bringing the formalism towards the representation of quantitative models of dynamical systems (Hardy & Robillard, 2004).

Stochastic Petri nets, one of the first quantitative extensions, differ from basic Petri nets by the interpretation of the nature of the transitions within a model. They can be used to adequately capture the stochastic nature of processes, influenced by intrinsic or extrinsic noise. The transitions within a model, represented by a stochastic Petri net, are considered to occur at time intervals drawn from an exponential distribution with predetermined rate parameter. The rate parameter can also be calculated as a function of the state of the input places, thus better capturing the stochastic kinetics of the interaction. The stochastic Petri nets can therefore be considered as an alternative to the direct representation by using a set of stochastic reaction equations.

Other extensions that allow for quantitative modeling using Petri nets are the continuous, hybrid and functional Petri nets. The continuous and hybrid extensions affect the state-space representation. In continuous Petri nets, the places can be assigned continuous state values, representing continuous variables such as concentrations. Hybrid Petri nets can have discrete or continuous states assigned to its places. Functional Petri nets allow for the explicit definition of the speed of transition, corresponding to a kinetic rate, usually as a function of the state of the input symbols. The continuous or hybrid and functional Petri nets have therefore continuous representation of state-space and continuous representation of time. They are interpreted deterministically and can be considered as an alternative to the direct representation by using a system of coupled ordinary differential equations.

Due to their versatility, Petri nets can be used to perform both qualitative and quantitative analysis. They have been used to model different types of dynamical systems in biology and have been proposed as a formalism for design of novel systems for synthetic biology (Chaouiya, 2007; Heiner, Gilbert, & Donaldson, 2008).

2.1.3 Process algebras

Another family of formalisms initially developed for representing concurrent systems, later adopted for modeling dynamical systems in biology, is the family of process algebras (Priami & Quaglia, 2004). The focus of process algebras (Milner, 1980) is the representation of the distributed communication between independent agents (processes) interacting through shared channels of communication by executing atomic actions. The sharing of channels between processes defines the topology of the network of interactions.

One of the first formalisms to be adopted for use in biology was the π-calculus (Milner, Parrow, & Walker, 1992). Regev, Silverman, and Shapiro (2001) demonstrated that the π-calculus can be used to represent and qualitatively analyze the behavior of molecular signalization pathways. In the π-calculus, the processes represent biological entities. The channels through which a process can communicate represent the properties of these entities. Finally, the communication between processes that share channels represents the dynamical interactions within the system.

In contrast to other formalisms, process algebras have several features that allow for the efficient representation of models. First, they allow for the representation of sequence of interactions, which a process (entity) can take part in. Second, they allow for the representation of concurrent interactions between different entities. Furthermore, they allow for defining competitiveness by representing different, uniformly distributed exclusive interaction options. They also allow for representing interactions that effectively modify the possibilities for interaction of different entities, by means of sending and receiving communication channels to and from other entities. Finally, they allow for representation of compositionality. The behavior of a complex system can be represented by rules of interaction, based on the properties of the entities, and their composition.

Similar to Petri nets, process algebras were initially used to represent qualitative models of dynamical systems in biology. The π-calculus is used to represent systems with discrete state-space. The actions within the system are instantaneous and the time is orthogonal to them. Process algebras require further extensions to be more suitable for broader range of applications within biology, and account for representing biological phenomena that are not supported by the basic process algebras. These extensions are related to the representation of compartments, bidirectional communications, reversibility and affinity of actions.

Among approaches that extend the descriptiveness of process algebras are the Brane Calculi (Cardelli, 2005), Beta binders (Priami & Quaglia, 2005) and CCS-R (Danos & Krivine, 2007). One of the most important extensions is the encoding of quantitative information related to the communication between processes. Therefore, a number of approaches such as the BioSPI (Priami, Regev, Shapiro, & Silverman, 2001), and Bio-PEPA (Ciocchetta & Hillston, 2009) have been developed that rely on stochastic variants of process algebras such as the stochastic π-calculus and the Performance Evaluation Process Algebra (PEPA). The stochastic process algebras rely on the continuous representation of time and incorporate the temporal dimension within the communication by assigning a rate of a performed action. The rate represents a parameter of an exponential distribution that accounts for the duration of the action. As a result the available actions, given the current state of the model, are assumed to be in a race condition in which only the fastest action will be selected to proceed. This allows the generation of a continuous time Markov chain from the system representation that can be further used for quantitative analyses.

The process algebras are powerful formalisms that can be used to represent complex systems in biology in an abstract but compact manner. On the other hand, the main downside is their understandability. The abstract but relatively complex representation of dynamical systems offered by process algebras may be intuitive to computer scientists or telecommunications engineers, however it may prove difficult for adoption by biologists.

2.1.4 Rule-based formalisms

Closely related to the process algebras and more recent are the rule-based formalisms. They are used for indirect detailed specification of (usually) large dynamical systems (Chylek et al., 2014). The main concept is that instead of specifying the entire set of interactions that comprise a dynamical system, a significantly smaller set of rules can be specified in the form of classes of interactions (reaction generators), from which they can be reconstructed. The development of rule-based formalisms is motivated by the tedious task of representing a network of interactions between entities for which every conformation or state needs to be explicitly enumerated and accounted for (e.g., by a different variable). Their use is therefore indicated for modeling dynamical systems for which the size of the set of rules of interactions is significantly smaller than the set of all interactions, such as signaling pathways and protein-protein interaction networks.

The most general description of the representation of dynamical systems using rule-based formalisms is based on graphs (Blinov, Yang, Faeder, & Hlavacek, 2006; Lemons, Hu, & Hlavacek, 2011). The structure of each constituent of a dynamical system (or complex) is represented by a colored and attributed graph (or connected set of graphs). The nodes are the molecular components (e.g., protein domain) colored with the type of the molecule they belong to, each associated with attributes that represent its state (e.g., conformation, phosphorylation). The edges of the graph represent the noncovalent bindings between the constituents. The interactions in the system are encoded by rules comprised of two parts: a "reactants" side, a set of graph patterns to be matched, which represent the sufficient conditions for an interaction to happen, and a "products" side, a set of graph patterns, which represent the result from the interaction. A mapping of nodes between the reactant and product patterns is assumed. The consequence of a rule is therefore the rewriting of reactant and product graphs (physical change in the structure of the constituents) as a result of an interaction. Each rule is followed by a constant rate which applies for all interactions that will be generated as a result of rule matching.

In most domain specific instances, such as the κ (Feret et al., 2009), BioNetGen (Faeder et al., 2009) and BIOCHAM (Fages, Soliman, & Chabrier-Rivier, 2004) languages, rule-based modeling formalisms are used for quantitative modeling. The state of the system is considered to be discrete and defined by the set of all graphs at a given time. The interpretation of rules can be both deterministic and stochastic depending on the task. Dynamical systems represented by using rule-based formalisms can be resolved using an initial (seed) state and iterative application of rules, resulting in a network of reactions or a system of ordinary differential equations.

2.1.5 Composition-oriented formalisms

Composition-oriented formalisms are high level formalisms applied to problems in synthetic biology, designed to obscure as much detail as possible from the process of modeling. The motivation comes from the computer aided design languages available for designing electronic circuits. The focus is at the level of representing a construct by composition of functional parts, trading-off the concern of detailed definition of the structure or kinetics of the interactions. The composition-oriented formalisms are based on the assumption that a comprehensive list of well characterized design parts exist, such that can be used interchangeably. Different implementations trade off the level of abstraction of different aspects of the design. Syntactically, the formalisms are diverse and range from domain-specific implementations of formal languages to object-oriented programming languages.

The formalism employed by GenoCAD (Cai, Hartnett, Gustafsson, & Peccoud, 2007), one of the first tools for synthetic biology, is based on a context-free grammar. Within

the grammar, biological parts (e.g., part identification number or genetic sequence of a promoter, RBS, protein coding sequence, spacer, terminator) are defined as terminals, while a complete hierarchy of constructs (e.g., Cistron, Operon, Device, System) can be defined at the level of non-terminals in the grammar. At this level of abstraction, GenoCAD can be used to only define syntactically correct construct (valid ordering to enable correct gene expression), while the function of the construct remains undefined. A further improvement of the GenoCAD formalism includes attribute grammars, which contain information about the structure-function relationship of the different parts within the construct (Cai, Lux, Adam, & Peccoud, 2009). This improvement allows for conversion of the construct into a quantitative model with continuous state-space and interactions that can be interpreted deterministically.

Eugene (Bilitchenko et al., 2011) is influenced by high-level hardware description languages. At the conceptual level it allows for definition of a list of Parts and their composition into Devices which can be collected into a System. Eugene enables definition of specific Parts by instantiation of the part object with different lower level properties (e.g., part id, sequence, orientation, kinetic properties), which can be directly mapped to a library of physical parts. The Devices are defined by a combination of compositional rules and conditionals imposed on the list of Parts. Subsequently, the devices can define different final constructs, composed of different Parts that adhere to the rules, enabling the possibility of exploration of the design space.

The GEC formalism (Pedersen & Phillips, 2009) adds another level of abstraction by not requiring knowledge of specific parts and introduces the concept of part types. The assumption is that two different databases are available. First, a database of well defined parts belonging to one of the possible types (promoter, RBS, protein coding regions, terminator). The low-level properties are also defined for each part and may serve as constraints (type of regulation, target of regulation, kinetic properties). Second, a database of prototype reactions in the form of reaction equations that take into account the properties of the parts. The construct representation is then based on defining the desired physical composition based on part types and imposing property-based constraints.

2.1.6 Differential equations and reaction equations

Good high-level formalisms tend to capture, at a lower level, quantitative representation of the dynamical behavior that is reducible to an exact mathematical representation. Under the assumption that biological systems are well-stirred chemical systems in thermal equilibrium, models in biology are represented mathematically as a set of continuous variables that evolve through time deterministically, according to a system of coupled ordinary differential equations. Modeling using a system of differential equations allows for encoding all the details needed for their interpretation. Models represented using differential equations are models represented at the lowest abstraction level, where every detail is fully specified. The formalism of ordinary differential equations (ODEs) is well accepted in the biological community due to its historical relationship with related fields such as (bio)chemistry, ecology and epidemiology (Murray, 1993).

However, the representation using a system of ODEs is not adequate for molecular systems with a small number of copies (only few orders of magnitude above one) of the reactive entities. The deterministic representation fails to account for the underlying stochasticity of natural systems (Wilkinson, 2006; Lecca, Laurenzi, & Jordan, 2013). A more general alternative to using a deterministic and continuous representation of these systems is a representation by using reaction equations. The latter can treat the system as discrete in terms of entities, but also stochastic in terms of the interactions between them. By using a set stochastic reaction equations as a baseline representation, the system is considered

to be composed of a set of molecular entities with a discrete number of copies that interact through a number of coupled reactions. Together with the appropriate kinetic rates, the model defines a network of possible transitions between system states with assigned probabilities. The stochastic dynamical behavior of the system, resulting from the possible state transitions through time, is captured by the evolution of the probability distribution over all possible discrete states that the system can be in (Gillespie, 1992). This exact representation of the dynamics of the system is referred to as the chemical master equation (CME) of the system (McQuarrie, 1967). It can be shown that under increasingly restrictive assumptions the CME can be approximated by a system of stochastic differential equations (SDEs), representing the Langevin Equation of the system, and further by a system of ODEs (Turner, Schnell, & Burrage, 2004; Gillespie, 2000).

Most of the approaches for systems and synthetic biology rely on modeling using reaction equations and differential equations. The use of these formalisms allows for the complete and most realistic analysis of the properties of the structure and the behavior of a dynamical system, taking into account potential aberrations and the application of control mechanisms.

2.2 Inferring Models of Dynamics in Biological Systems

Recall that the resolving of the fundamental structural and parametric uncertainties are the two parts of the process of modeling and design of dynamical biological systems. Regardless of whether the two types of uncertainties are formulated as a single problem that can be solved to resolve both of them simultaneously, or as two consecutive and dependent problems, they can be cast into the frame of finding the best solution among a set of possibilities. Within this frame, the problem can be approached by either search or optimization strategies.

Both search and optimization have a long history, branching into a large number of subfields, focusing on specific types of problems and the development of strategies for their solution. Here, we focus only on a relatively small subset of approaches to resolving uncertainty, applied to problems coming from the domains of systems and synthetic biology, represented through examples of related work relevant to the topic of this thesis.

Search is the predominant problem solving strategy used in artificial intelligence and machine learning (Russell & Norvig, 2010). A search problem is symbolically represented using the following components: a state space S; an initial state $s_0 \in S$; a mapping t, which maps a state s to a subset of actions $A = \{a_i | a_i : S \to S\}$ applicable to s, representing a transition model; a goal test function $g : S \to \{true, false\}$, which determines if a state fulfills the predefined properties of the goal of the search, by making use of information from the already explored states; and a cost function $f : (S, t) \to \mathbb{R}$, used to guide the search towards a goal state. Multiple states in the search space can be goal states of the search. The solution of a search problem is the path from the initial to a goal state, i.e., the set of actions taken to reach the goal.

Optimization (mathematical programming) is the task of finding the best possible solution to a problem, subject to an objective function, from a (constrained) space of alternatives. Optimization is commonly applied to problems in various domains ranging from science and engineering to finance and management.

Mathematically, optimization is defined as the minimization or maximization of a function $f : S \to \mathbb{R}$, that maps the elements from a non-empty set of solutions $S = \{s_i | s_i \in D\}$, where $D = D_1 \times \cdots \times D_k$ is the domain of the variables comprising a solution, to a real valued number, by finding a solution $s*$ such that $\forall s \in S : f(s*) \leq f(s)$, if f is subject to minimization or $f(s*) \geq f(s)$, if f is subject to maximization. The set of solutions S is

also known as a solution or a search space of the problem. Each solution $s \in S$ is subject to constraints $\{g_i(s) \leq 0, i = 1, \ldots, n\} \cup \{h_i(s) = 0, i = 1, \ldots, m\}$, where $n, m \geq 0$.

The function f is also known as objective, fitness, cost or utility function. The choice of an objective function is central to the optimization problem. The shape of f in the objective space given a set S defines the difficulty of an optimization problem and the type of optimization algorithm that can be applied to solve the problem.

2.2.1 Approaches to complete model inference

The modeling formalisms for systems and synthetic biology are commonly used to encode a model of a dynamical system with fixed structure and parameter values. Surprisingly small amount of attention has been given to the problem of direct and formal encoding of uncertainties in the structure and the parameter values of quantitative models. Among the approaches to complete model inference where a reconstruction is performed of both the causal interactions within a system, their specific form and the resulting dynamical behavior, the following classes can be identified: the class of composition-oriented approaches, the class of network reconstruction based approaches, and the class of optimization-based approaches.

The class of composition-oriented approaches are used for the task of design of biological systems. These approaches cast the problem of complete model inference into a combinatorial search problem. Given a collection of defined parts, constraints on their composition can be imposed using rule-based formalisms, such as κ and BioNetGet (Marchisio, Colaiacovo, Whitehead, & Stelling, 2013; Wilson-Kanamori, Danos, Thomson, & Honorato-Zimmer, 2015) or composition-oriented formalisms, such as GEC and Eugene. These constraints define a space of candidate compositions with valid physical implementations (DNA sequence) for a single structure that can be enumerated by combinatorial search and further analyzed by experts. Approaches such as Proto (Beal, Lu, & Weiss, 2011) and Cello (A. A. K. Nielsen et al., 2016) make use of composition-oriented formalisms and implicitly consider the possibility of structural uncertainty within the composition. They define and resolve structural uncertainty through a high-level description of design objectives. The design objectives are formulated as a Boolean function of parts that are identified as input and output of the system by using a functional Lisp-like language (Proto) or a high-level hardware description language (Cello). These approaches first infer an abstract network representation of a composition of standard parts that implements all intermediary logic functions needed to achieve the objective. These compositions are then resolved into a specific physical construct using Eugene or more advanced methods, such as MatchMaker (Yaman, Bhatia, Adler, Densmore, & Beal, 2012). All of these approaches infer models by resolving only the structural uncertainty. The type of interactions among the model components and the values of the parameters of these interactions are predefined and fixed.

A lot of work has been done on the topic of large-scale biological network reconstruction from expression data (Bansal, Belcastro, et al., 2006; Penfold & Wild, 2011). However, only few of the approaches to network reconstruction are exception from the standard approaches that consider only the statistically significant dependence between the observations as the evidence for interaction and neglect the dynamics of the interactions. The approaches to complete model inference based on network reconstruction cast the networks into a system of ODEs and try to infer causal interactions between the entities. These methods are applied to problems of structure identification for large-scale networks. Therefore additional assumptions and simplifications are taken into account that make these approaches computationally feasible. All of these methods follow the assumption that the same form of kinetics applies to all interactions, most frequently linear kinetics with additional terms for degradation or external influence (Gardner, di Bernardo, Lorenz,

& Collins, 2003; Bonneau et al., 2006; Bansal, Gatta, & di Bernardo, 2006). Since a model represented by a system of differential equations requires computationally expensive numerical integration, some approaches take into account only the steady states of the entities (Gardner et al., 2003; Bonneau et al., 2006), and solve only for the steady state of the system. Others try to discretize and reduce the dimensionality of the system (Bansal, Gatta, & di Bernardo, 2006). When it comes to constraining the space of possible model structures, some methods employ data-driven heuristics (Wahl, Haunschild, Oldiges, & Wiechert, 2006; Bonneau et al., 2006), while some of them additionally limit the space of structures based on the interactions already documented in the literature (Wahl et al., 2006; Henriques, Rocha, Saez-Rodriguez, & Banga, 2015). In general, these methods are adequate for generation of initial model structures, candidates for revision upon availability of data and hypotheses about the specific types of interaction dynamics.

The third class of optimization based approaches require the rigid representation of structural uncertainty as parameters. They also allow for specification of broader class of objectives related to the properties of the behavior of the model (Marchisio & Stelling, 2009). The different possible structures can be encoded by rules for evolution of the structure of a dynamical system (François & Hakim, 2004), by simple manual enumeration of equation-based models, or as parameters within a single, general equation-based model that indicate the presence of individual fragments (interactions) within its structure. Once formulated as an optimization problem, different methods can be applied towards its solution. The target of the optimization can be a single objective (Rodrigo et al., 2007; Dasika & Maranas, 2008; Toni et al., 2009) or multiple objectives (Sendin et al., 2010; Barnes et al., 2011; Higuera, Villaverde, Banga, Ross, & Morán, 2012; Otero-Muras & Banga, 2014). These approaches are further discussed in the subsection that follows.

A recent work by Villaverde and Banga (2013) gives a general overview of the different methodological perspectives of the problem of model inference coming from different areas. It presents approaches coming from the aspect of network reconstruction and the aspect of identification of complete dynamical models that can be applied towards its solution, the overlap of goals and ideas and the possibility of convergence.

In contrast to other approaches, process-based modeling approaches are compositional and formulate the task of model inference as a two-level approach towards learning a model of a dynamical system, i.e., search in a formally defined and domain-knowledge constrained space of candidate model structures guided by the results from the optimization of the constant parameters. These approaches are presented in more detail in the following section.

2.2.2 Parameter estimation

Almost all approaches to complete model inference require resolving the uncertainty in the constant parameter values of the model, by their estimation from observations (Jaqaman & Danuser, 2006; Kirk, Silk, & Stumpf, 2016). The treatment of the parameters that require estimation may differ according to the approach taken for parameter estimation. Namely, two classes of approaches that are most commonly considered are the Bayesian (probabilistic) and the frequentist (e.g., Maximum Likelihood (ML)) approaches. Both classes treat the constant parameters (θ) of the model as variables for the task of estimation. However, the Bayesian approaches treat the parameters as random variables with corresponding likelihood (probability density) over their value given the observations (d), while the ML approaches treat the parameters as variables that can be assigned only a single value that maximizes the likelihood of the observations.

Formally, the Bayesian approaches combine prior knowledge about the values of the parameters $P(\theta)$ and the observations d, using a likelihood function $f(d|\theta)$ in order to

obtain a posterior distribution of the parameter values after the observation of the data $P(\theta|d)$ following the Bayes' rule:

$$P(\theta|d) = \frac{P(\theta) \cdot f(d|\theta)}{P(d)} \approx P(\theta) \cdot P(d|\theta) \tag{2.1}$$

However, Equation 2.1 does not always have a closed form solution and the process of estimation must be performed by iterative updating of the posterior distribution by using Markov Chain Monte Carlo sampling algorithms. Optimal candidate sets of parameter values $\hat{\theta}$ for the model can then be obtained by sampling the posterior. A common choice of candidate set of parameters is the mode of the posterior (maximum a posteriori estimate).

The ML approaches, as a popular example of a frequentist approach, are concerned with obtaining a fixed set of values for the parameters $\hat{\theta}$ that are most likely to have produced the observations d, by maximizing the likelihood function $L(d|\theta) = f(d|\theta) = \prod_{i=1}^{n} f(d_i|\theta)$, where d is assumed to be consisted from independent and identically distributed observations:

$$\hat{\theta} = argmax_\theta L(d|\theta) \tag{2.2}$$

For example, under the assumption that the observations follow a normal distribution, the well known Least Squares (LS) estimator can be obtained by considering $f(d_i|\theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma^2}} \cdot exp(-\frac{r_i^2}{2 \cdot \sigma^2})$, where r_i is the residual $r_i = \hat{d}_i - d_i$ and $\hat{d}_i$ is obtained by simulating the model using parameters θ:

$$\widehat{\theta_{LS}} = argmax_\theta L(d|\theta) = argmax_\theta \ln L(d|\theta) \approx argmax_\theta - \frac{1}{2} \sum_{i=1}^{n} \frac{r_i^2}{\sigma^2} \approx argmin_\theta \sum_{i=1}^{n} r_i^2 \tag{2.3}$$

However, in biology, due to the size and complexity of the models, the scarcity of observations and the different sources of noise for obtaining the data, the likelihood function can be difficult to calculate or even define. Therefore, in general, the methods used for parameter estimation rely on different objective functions that are either approximations of the likelihood function or on heuristic functions that capture the qualitative or quantitative properties of the behavior of the models, with regards to the observations or expectations, and guide the optimization.

Parameter estimation of dynamical biological models is in general a nonlinear optimization problem with differential-algebraic constraints. The estimation of parameters of dynamical biological models is a hard problem. The problem is frequently high dimensional, the objective space is commonly multi-modal, and the evaluation of the objective function can be time-consuming since it relies on the simulation of the model. Different Bayesian and frequentist approaches to parameter estimation that deal with these issues have been applied to problems of modeling dynamical systems in biology.

Bayesian methods are flexible in terms of representation of the uncertainty in the parameter values and the minimal amount of assumptions required for their execution. Furthermore, the output from the Bayesian approaches capture the most amount of information about the estimated parameters in the form of a posterior distribution over the values of the parameters, which can be used for further analysis and comparison of models. However, the application of Bayesian methods comes with the cost of high computational complexity and have therefore been applied only on a limited number of relatively small problems (Wilkinson, 2007). More recently approaches based on approximate Bayesian computation have been applied to estimate parameters from observations for the task of modeling (Toni et al., 2009) and design (Barnes et al., 2011; Silk et al., 2011) of deterministic and stochastic dynamical biological systems.

Frequentist approaches are more prominent for parameter estimation applications for modeling and design of dynamical biological systems. Traditionally the parameter estimation has been performed using gradient descent methods, especially the ones based on optimization of the least squares function as shown in Equation 2.3, such as the Gauss-Newton or the Levenberg–Marquardt method (Mendes & Kell, 1998; Ashyraliyev, Fomekong-Nanfack, Kaandorp, & Blom, 2009). The main drawback of these methods is their local nature. The objective function, for complex models with many parameters, can have several local minima in which these optimization methods may get trapped. This drawback resulted in their prompt replacement with global optimization methods (Moles et al., 2003).

More recently, global stochastic optimization and hybrid algorithms, especially the ones based on metaheuristics, have been considered as the most promising (Chou & Voit, 2009; Ashyraliyev et al., 2009). Metaheuristic approaches are able to handle efficiently a wide range of problems subject to optimization of arbitrary objective functions, by quickly converging to a near-global optimum. Out of the many different parameter estimation approaches, the ones based on Evolutionary Strategies and Differential Evolution have been recommended as the most successful for systems biology applications (Sun et al., 2012).

It is worth noting that most of the parameter estimation methods applied to the problem of modeling dynamical biological systems, although able in general to optimize arbitrary objective functions, consider frequently only the sum of squared errors between the model simulation and the observations as a single objective function. There are also, although few, examples of simultaneous optimization of multiple conflicting objective functions as an approach to design of dynamical biological systems. Most of them are optimization-oriented approaches to inferring complete models of dynamics, (Handl, Kell, & Knowles, 2007; Sendin et al., 2010; Barnes et al., 2011; Higuera et al., 2012; Otero-Muras & Banga, 2014) that consider resolving structural and parametric uncertainty as a single optimization problem. The same multi-objective optimization methods can be applied to the problem of parameter estimation for a single structure.

For the task of design of biological systems, due to the inability to consider observations within the objective function used for optimization of parameter values, the objectives become more specific and relate to the behavior that the dynamical system has to optimally satisfy. They range from adapting the sum of squared errors objective function by replacing the observations with a target trajectory represented in a form of a function (Rodrigo et al., 2007; Rodrigo & Jaramillo, 2013) to more specific objectives based on the qualitative and quantitative properties of the behavior of the system. These objectives can target, for example, the required expression levels of transcripts as response to different inducers (Dasika & Maranas, 2008) or the properties of the stability represented by the Lyapunov exponents (Silk et al., 2011), eigenvalues and components from the Fourier spectrum (Barnes et al., 2011).

2.3 Process-Based Modeling

Process-based modeling (also referred to as process modeling or inductive process modeling) is an approach to modeling dynamical systems that combines domain specific knowledge and data (time-series of observations of a behavior). It has been defined as a machine learning problem by Langley, Sanchez, Todorovski, and Džeroski (2002).

This learning problem is motivated by related approaches to computational scientific discovery (Langley et al., 1992), namely equation discovery, by automated learning of numerical laws, both in the form of general algebraic equations and differential equations

from time-series data. As an improvement to these approaches, PBM addresses the important issue of the explanatory power and communicability of the derived models, by exploiting domain-specific knowledge in the specification of the model space as a replacement to the specification using generic algebraic terms. This domain specific learning task makes use of a knowledge generalized in the form of generic causal relations, i.e., representations of physical (mechanistic) phenomena specific for the domain of interest, as a means to strengthen the representational bias. The resulting models are therefore based on established modeling concepts and theories in the domain of interest, formulated in a modular fashion using the easily understandable notions of *entities* (model constituents) and *processes* (set of interactions between entities representing a common occurrence). The notions of entities and processes used to formally represent a model are ontologically well grounded as continuants and occurrents, commonly and intuitively (Forbus, 1984) used in the scientific and engineering domains.

The formalism used for PBM is orthogonal to the previously described formalisms. The quantitative information conveyed by the processes can in principle be represented using a range of formalisms that are able to quantify causal relations. The entities in PBM are represented by a named set of variables and constants that represent their properties in a hierarchy of types. The processes in PBM use a set of fragments of algebraic and ordinary differential equations to relate the change of the variable properties of the entities to the continuation of a process, assuming a deterministic and continuous state space represented by the values of the entity variables in a continuous time frame.

The task of process-based modeling takes at input: domain-knowledge cast into a library of domain-specific generic entities and processes, set of entities in the observed system, optional constraints on the processes defined for groups of these entities, and time-series of observations of properties of the behavior of the system being modeled, related to the variable properties of the entities. At output it provides a complete quantitative process-based model that adequately explains the observations.

A complete model is constructed by performing a constrained search through the space of process components, provided by the library of domain knowledge, and by modular composition by instantiation, taking into account the specific entities that comprise the observed system. The merit of a candidate construct (model) is established by estimation of the values of the constant parameters included in the processes of each candidate by performing nonlinear least-squares optimization. The objective of the parameter estimation is the minimization of the sum of squared errors between the model trajectories (obtained by simulating the model with the candidate parameters) and the provided observation data.

Different implementations address the task of PBM by considering different choices for the specifics of the formal representation, the imposing of constraints, search and optimization. It is worth noting that each successive implementation or improvement of a PBM approach tends to strengthen the representational bias as a mechanism to improve the performance and the explanatory power of the process-based models. The method of evaluation and the form of the evaluation bias, however, have remained unchanged.

The work presented in this thesis extends ProBMoT, the most recent implementation of PBM. Therefore, we focus on its representation of domain-knowledge, constraints and workflow in more detail.

2.3.1 IPM

Inductive Process Modeling (IPM) (Bridewell et al., 2008) formalizes the domain knowledge as a set of *generic processes*. The processes do not relate to specific entities, but to *generic entity* types. The entity types are defined by a hierarchy of types with a common root

– number. Each generic process contains a set of fragments of algebraic or differential equations that capture the changes of the quantity of the related variables (entities) that are the result of the process, the corresponding constant parameters and their value ranges.

The set of processes is unstructured, i.e., flat. All processes are at the only existing system level, leading to underconstrained model search space. Given entity instances with assigned types, IPM exhaustively generates the set of all valid instantiations of the generic processes. The search through the model space is performed by composing candidate model structures (generic models) using the powerset of the generated set of processes. During the search, IPM takes a naïve approach and uses constraints in the form of a list of processes that must be instantiated and an upper limit on the number of processes involved in the model in order to address the combinatorial explosion. However, this manner of composition is suboptimal in terms that it considers all process combinations as plausible, which might result in consideration of faulty models during the search, e.g., a candidate model that contains multiple alternatives of the same process.

In order to perform optimization of the values of the constant parameters defined for each process, the complete set of algebraic and differential equations for each candidate model structure is derived by additive aggregation of the equation fragments at the level of entities. The parameters are estimated by minimization of the sum of squared errors between the model simulation and the observations using a local optimization method based on gradient descent.

2.3.2 HIPM

Hierarchical Inductive Process Modeling (HIPM) (Todorovski et al., 2005) extends the IPM formalism for encoding the library of domain-knowledge by extending the definition of an *entity* and by imposing a structure on the *generic processes*.

Entities in the HIPM formalism are not represented simply as a type but are associated with properties represented by a set of variables and parameters. HIPM additionally relaxes the assumption of additive aggregation of entity variables and allows for definition of an aggregation function at the level of entity variables.

In the HIPM formalism generic processes can be organized as a hierarchy of process alternatives, each with a hierarchy of subprocesses, some of which may be marked as optional. The process alternatives are considered to be mutually exclusive at the same level of the hierarchy, while the subprocesses define valid structure patterns. In this way, HIPM replaces the assumption that any set of processes can be combined to compose a valid candidate model structure with a more realistic one. The model search space is thus constrained by both disjunctive and conjunctive rules and limited to more feasible models.

HIPM does not generate all valid instances of the generic processes but performs a heuristic beam search guided by refinement operators following the exclusivity and optionality of processes defined in the hierarchy. The parameters are estimated by using the same non-linear least-squares optimization method based on gradient descent as IPM. However, the parameter estimation for each candidate model is performed multiple times with random initialization in order to avoid local minima.

A drawback of HIPM is that its formalism is not independent from the implementation of the induction algorithm. HIPM allows for encoding of domain knowledge only within the internal structure of the algorithm, i.e., directly in its implementation using the Python programming language, which might be disconcerting for some domain experts that are not familiar with programming languages.

2.3.3 SC-IPM

Satisfying Constraints to Induce Process Models (SC-IPM) (Bridewell & Langley, 2010) is another implementation of PBM. It is an improvement of IPM and HIPM that extends the formalism towards the representation of *structural constraints*, overcoming the problem of considering faulty models.

The constraints are imposed on subsets of *generic processes* from the library of domain knowledge and can be one of the following types: *necessary*, *always-together*, *at-most-one* and *exactly-one*. The *necessary* constraint defines processes that must be instantiated in every candidate model structure. The *always-together* constraint defines a subset of generic processes that must be either instantiated together or not instantiated at all. The *at-most-one* and *exactly-one* constraints define mutually exclusive processes, where the *at-most-one* is a less-strict constraint which allows for the possibility that no process is instantiated.

The search in the space of candidate models is performed by a modified beam search following a two part process: first, using constraint satisfaction methods, a minimal candidate structure that satisfies the constraints is generated; second, the structure is expanded by including unconstrained processes. The parameter estimation procedure is the same as in HIPM.

2.3.4 ProBMoT

The Process-Based Modeling Tool (ProBMoT) (Čerepnalkoski et al., 2012; Čerepnalkoski, 2013) is the most recent implementation of PBM. It represents a complete solution in the form of integrated workflow, which includes an improved formalism for representation of domain-knowledge and global meta-heuristic optimization methods for parameter estimation.

The domain knowledge in ProBMoT is organized in a library defined in terms of *templates* which correspond roughly to the notion of *generics* used by the other formalism. The term template captures the character and the treatment of these constructs, offering a slightly different but important point of view of the nature of organization of domain specific knowledge and its use as a basis for the composition of models. A template is a representation of a concept, i.e., a sample of knowledge that is general enough to be reused in different scenarios. Information within the templates is represented in relation to other templates. An instantiation of a template is defined as a copy with replacement by specification.

A *library* is a named collection of template compartments, entities and processes.

Template compartments are containers used to organize named collections of template compartments, entitites and processes. Template compartments can be considered as a smaller scale library within a library, used for hierarchical organization of larger knowledge bases and enabling multi-compartmental modeling.

Template entities are named collections of variable and constant templates. They are used to represent common properties of the entity. The variable templates are represented by a range and an aggregation function as in the HIPM formalism. The constant templates are represented only by their range. The constant templates are commonly used to represent the constant parameters of an interaction, which the template entity takes part in.

Template processes are used to represent causal relations between template entities. Each template process is associated by arguments represented by entity types that are involved in the processes which are used to constrain its instantiation. The template process is defined by optional named collections of constant, equation and nested process templates. The constant templates, as in the template entities, are defined by their range

and usually represent constant parameters of the process that may be subject to estimation. The equation templates quantify the relation between the entities in the form of fragments of algebraic or ordinary differential equations, i.e., a function of the constants defined in the template process they belong to and the constants and variables defined in the entity templates associated as arguments to that process. The nested process templates allow for decomposition of a larger process into a collection of smaller ones. All nested processes are references to existing process templates. The nested processes must accept a subset of the argument types of the process in which they are nested. The nested processes themselves may contain nested processes, but cyclic nesting is not allowed.

Entity and process templates can be taxonomically organized in inheritance trees. Namely, templates lower in the hierarchy inherit the properties of the ancestors. Entities inherit the variables and constants, while processes inherit the arguments, constants, equations and nested processes. Additionally, templates at the same level in the tree are considered to be mutually exclusive.

For example, let us consider a simple library of domain-knowledge for transcriptional regulation and the use of this library to encode a model of a relaxation oscillator (Novak & Tyson, 2008; Zhou, Zhang, Yuan, & Chen, 2008). The knowledge in the library is simplified to include up to second order linear reaction kinetics.

A relaxation oscillator is based on a combination of a fast positive and delayed negative feedback mechanism. The basic structure of the oscillator is shown in Figure 2.1A. Each of the two nodes in the basic structure corresponds to a regulated gene that expresses a mRNA coding for a protein (transcription factor) as shown in Figure 2.1B.

The library contains five template entities shown in Table 2.1: `Gene`, `Product`, `mRNA`, `Protein` and `Complex`. Each template entity is represented by a template variable property named `quantity`. Note that both `mRNA` and `Protein` are child template entities in a simple hierarchy, where `Product` is the parent template, and inherit the property `quantity`. The template entity `Complex` represents the complex formed by binding a protein to the regulatory region of a gene. Two of the template entities contain additional constants. The template entity `Gene` contains a template constant `ktx` which corresponds to the unregulated transcription rate of the gene. Similarly, `mRNA` contains a template constant `ktl` which corresponds to the unregulated translation rate of the mRNA. Each property

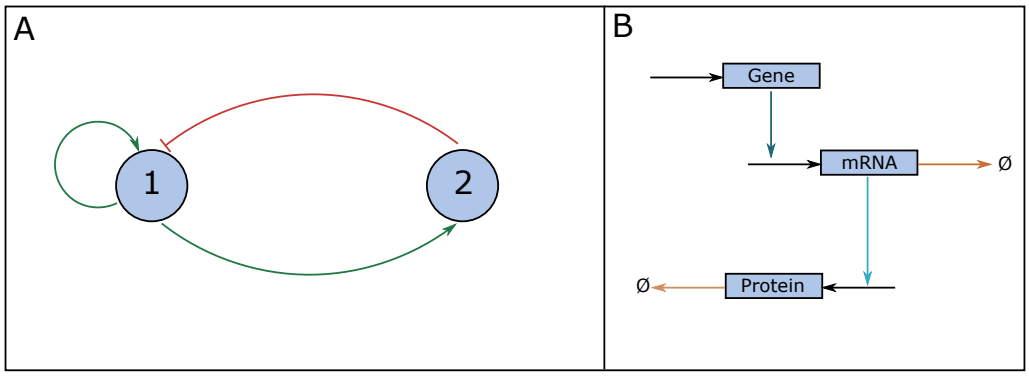

Figure 2.1: A) Topology of the genetic relaxation oscillator. Each node is an abstraction of a structure consisting of a gene, mRNA and a protein. The green arrows (→) represent positive regulation while the red arrow (⊣) represents negative regulation (inhibition) B) The detailed structure of the nodes. The dark blue arrow represents the transcription of mRNA from a gene. The light blue arrow represents translation of mRNA to a protein. The orange arrows represent degradation of products.

is represented by a corresponding range. The aggregation function is deliberately omitted and left to the default - addition.

In the graphical representation of the model (representation of a model as a drawing is a common practice in the biological community), the nodes in Figure 2.1A represent abstraction of a more complex common structure. Each arc in both Figure 2.1A and B is also an abstraction of a process guided by fundamental quantifiable interactions which can and should be mathematically encoded (Tyson, 2007). Each type of color or an arrow represents a specific type of process which can be encoded in the library of domain knowledge, which has a common underlying mathematical representation, and as a template can be reused each time a specific type of arrow appears in this drawing or another drawing of a model from the same domain. This mapping is reflected in the library as a collection of template processes shown in Table 2.2.

Table 2.1: Template entities defined in the library of domain-knowledge for transcriptional regulation.

```
library TranscriptionalRegulation;

template entity Gene {
  vars: quantity{range: <1, 100>};
  consts: ktx{range: <0.001, 10>};
}

template entity Product {
  vars: quantity{range: <0, 100>};
}

template entity mRNA : Product{
  consts: ktl{range: <0.001, 10>};
}

template entity Protein : Product {}

template entity Complex {
  vars: quantity{range: <0, 100>};
}
```

The template process `basic` is a compound process putting together all the processes that occur in a single node. It therefore represents a formal representation of each node, as depicted in Figure 2.1B. The nested processes reference three other template processes. The process of `transcription` of `mRNA` from a `Gene`, `translation` of a `Protein` from a `mRNA` and `degradation` of both mRNA and protein. Note that the template process `degradation` has assigned as argument an entity of the type `Product`, since `mRNA` and `Protein` are both products in terms of the defined hierarchy of template entities, they satisfy the type constraint.

The processes of regulation, representing interactions between two nodes, are encoded in a hierarchy of template processes. At the top level the template process `regulation` defines three constant properties and template equations that represent the change in the quantity of a `Gene`, `Protein` and a gene-protein `Complex` that result from the reversible binding of the protein to the regulatory region of the gene, thus regulating the expression of `mRNA`. The template processes `inhibition` and `activation` represent mutually exclusive mechanisms of positive and negative regulation correspondingly. Each of these two processes inherits

Table 2.2: Template processes defined in the library of domain-knowledge for transcriptional regulation. In the template equations, `td` is a keyword, denoting a time derivative, recognized by ProBMoT and the dot separator is used to access the properties of an entity.

```
template process basic(g: Gene, m: mRNA, p: Protein){
  processes:
    transcription(g, m),
    translation(m, p),
    degradation(m),
    degradation(p);
}

template process transcription(g: Gene, m: mRNA){
  equations: td(m.quantity) = g.ktx*g.quantity;
}

template process translation(m: mRNA, p: Protein){
  equations: td(p.quantity) = m.ktl*m.quantity;
}

template process degradation(d: Product){
  consts: kd{range: <0.001, 10>};
  equations: td(d.quantity) = -kd*d.quantity;
}

template process regulation(p: Protein, g: Gene, pg: Complex, m: mRNA){
  consts:
    kf{range: <0.001, 10>}, kr{range: <0.001, 10>}, km{range: <2, 1000>};
  equations:
    td(p.quantity) = -kf*p.quantity*g.quantity + kr*pg.quantity,
    td(g.quantity) = -kf*p.quantity*g.quantity + kr*pg.quantity,
    td(pg.quantity) = kf*p.quantity*g.quantity - kr*pg.quantity;
}

template process inhibition : regulation{}

template process activation : regulation{
  equations: td(m.quantity) = km*g.ktx*pg.quantity;
}

template process dimerization(p: Protein, dp: Protein){
  consts:
    kf{range: <0.001, 10>}, kr{range: <0.001, 10>};
  equations:
    td(p.quantity) = -2*kf*p.quantity*p.quantity + 2*kr*dp.quantity,
    td(dp.quantity) = kf*p.quantity*p.quantity - kr*dp.quantity;
}
```

the template equations and constants from the parent template. Therefore, the lack of additional template equations in the `inhibition` process represents that the mechanism of inhibition is based only on binding of a transcription factor that blocks the recruitment of RNA Polymerase needed for transcription, while the `activation` mechanism is based

on binding to a regulatory region that enhances the production of `mRNA` by `km` times, with regards to the unregulated transcription rate.

Table 2.3: Entities instantiated for the process-based model of a genetic relaxation oscillator. Note that the first line in the model definition defines the model name and links the library where the entity and process templates are defined.

```
model RelaxationOscillator : TranscriptionalRegulation;

entity g1 : Gene{
  vars: quantity {role: endogenous; initial: 1;};
  consts: ktx = 0.05;
}
entity m1 : mRNA{
  vars: quantity {role: endogenous; initial: 0;};
  consts: ktl = 0.1;
}
entity p1 : Protein{
  vars: quantity {role: endogenous; initial: 0;};
}

entity g2 : Gene{
  vars: quantity {role: endogenous; initial: 1;};
  consts: ktx = 0.05;
}
entity m2 : mRNA{
  vars: quantity {role: endogenous; initial: 0;};
  consts: ktl = 0.1;
}
entity p2 : Protein{
  vars: quantity {role: endogenous; initial: 0;};
}
entity p2dimer : Protein{
  vars: quantity {role: endogenous; initial: 0;};
}

entity p1g1 : Complex {
  vars: quantity {role: endogenous; initial: 0;};
}
entity p1g2 : Complex {
  vars: quantity {role: endogenous; initial: 0;};
}
entity p2g1 : Complex {
  vars: quantity {role: endogenous; initial: 0;};
}
```

Some forms of regulation require cooperative binding of transcription factors to the regulating regions of the gene. As a candidate template process for the library we consider the possibility that a dimer form of a transcription factor might be required for regulation. Therefore we introduce the template process representing reversible `dimerization`.

The template entities and processes as shown in Table 2.1 and 2.2 completely define the library of domain knowledge, given as input to ProBMoT.

Table 2.4: The process instances representing a process-based model of a genetic relaxation oscillator.

```
process basic1(g1, m1, p1) : basic{
  processes: txg1, tlm1, dm1, dp1;
}
process basic2(g2, m2, p2) : basic{
  processes: txg2, tlm2, dm2, dp2;
}

process txg1(g1, m1) : transcription{}
process txg2(g2, m2) : transcription{}
process tlm1(m1, p1) : translation{}
process tlm2(m2, p2) : translation{}
process dm1(m1) : degradation{
  consts: kd=0.2;
}
process dm2(m2) : degradation{
  consts: kd = 0.2;
}
process dp1(p1) : degradation{
  consts: kd = 0.4;
}
process dp2(p2) : degradation{
  consts: kd = 0.4;
}

process dimer(p2, p2dimer) : dimerization{
  consts: kf = 0.1, kr = 0.1;
}

process g1autoactivation(p1, g1, p1g1, m1) : activation {
  consts: kf = 1, kr = 1, km = 300;
}
process g2activation(p1, g2, p1g2, m2) : activation{
  consts: kf = 1, kr = 1, km = 100;
}
process g1inhibition(p2dimer, g1, p2g1, m1) : inhibition{
  consts: kf = 1, kr = 0.1, km = 0;
}
```

Given a library of domain knowledge, a *process-based model* is a collection of *instantiated* compartments, entities and processes. In order to model the structure depicted in Figure 2.1 we instantiate 10 specific entities as shown in Table 2.3. For each node we instantiate an entity representing a gene (`g1` and `g2`), mRNA (`m1` and `m2`) and a protein (`p1` and `p2`). We further instantiate three entity complexes `p1g1`, `p1g2` and `p2g1` representing the bound states of a combination of proteins and genes. Finally we instantiate a protein entity `p2dimer` which represents the dimerized form of the protein `p2`. The entity instances contain initial values for their state variable and constant properties and the role of each variable property.

The role is an additional information required by ProBMoT to be defined for each variable. The role of a variable for a specific model can be exogenous or endogenous. The variables marked as exogenous are treated as inputs to the system that are not modeled, but their behavior is mapped to external observations. The endogenous variables are

modeled as a part of the system and must be assigned an equation that is a result of process influences. We assume that the genetic relaxation oscillator is a closed system, therefore we define every entity to contain only endogenous variables.

The entity relationships for the model of a genetic relaxation oscillator are represented by 14 processes as shown in Table 2.4. For each node we instantiate a basic process that incorporates four nested processes that define the basic processes of transcription, translation and degradation. The regulation is defined by four process instances: a `g1autoactivation` process instance that corresponds to the positive regulation of the transcription of `m1` by the binding of `p1` to `g1` and formation of a `p1g1` complex; a `g2activation` process that corresponds to the positive regulation of the transcription of `m2` by the binding of `p1` to `g2` and formation of a `p1g2` complex; a process of `dimerization` of protein `p2` to a `p2dimer` protein; and a `g1inhibition` process that corresponds to the negative regulation of the transcription of `m1` by the binding of `p2dimer` to `g1` and formation of a `p2g1` complex.

The entity and process instances as shown in Table 2.3 and 2.4 completely define a process-based model that can be given as input to ProBMoT together with the corresponding library of domain knowledge which defines the templates from which they are instantiated.

For the genetic relaxation oscillator example shown in Table 2.3 and 2.4, ProBMoT compiles the following set of ODEs from the complete model, where $[x]$ denotes the quantity of x:

$$
\begin{aligned}
\frac{d[g1]}{dt} &= -1 \cdot [g1] \cdot [p2dimer] + 0.1 \cdot [p2g1] - 1 \cdot [g1] \cdot [p1] + 1 \cdot [p1g1] \\
\frac{d[m1]}{dt} &= 0.05 \cdot [g1] - 0.2 \cdot [m1] + 300 \cdot 0.05 \cdot [p1g1] \\
\frac{d[p1]}{dt} &= -1 \cdot [g2] \cdot [p1] + 1 \cdot [p1g2] + 0.1 \cdot [m1] - 0.4 \cdot [p1] - 1 \cdot [g1] \cdot [p1] + 1 \cdot [p1g1] \\
\frac{d[g2]}{dt} &= -1 \cdot [g2] \cdot [p1] + 0.1 \cdot [p1g2] \\
\frac{d[m2]}{dt} &= 0.05 \cdot [g2] - 0.2 \cdot [m2] + 100 \cdot 0.05 \cdot [p1g2] \\
\frac{d[p2]}{dt} &= -2 \cdot 0.1 \cdot [p2] \cdot [p2] + 2 \cdot 0.1 \cdot [p2dimer] + 0.1 \cdot [m2] - 0.4 \cdot [p2] \\
\frac{d[p2dimer]}{dt} &= -1 \cdot [g1] \cdot [p2dimer] + 0.1 \cdot [p2g1] + 0.1 \cdot [p2] \cdot [p2] - 0.1 \cdot [p2dimer] \\
\frac{d[p1g1]}{dt} &= 1 \cdot [g1] \cdot [p1] - 1 \cdot [p1g1] \\
\frac{d[p1g2]}{dt} &= 1 \cdot [g2] \cdot [p1] - 0.1 \cdot [p1g2] \\
\frac{d[p2g1]}{dt} &= 1 \cdot [g1] \cdot [p2dimer] - 0.1 \cdot [p2g1]
\end{aligned}
\tag{2.4}
$$

If we define a simulation task for the process-based model and define as model output as the time evolution of the quantity of unbound proteins `p1` and `p2`, we obtain the time-series shown in Figure 2.2 as output of ProBMoT.

A process-based model is *complete* if all initial values for the variable properties of each instantiated entity and the values for all entity and process constants are assigned, and if each process in the model is instantiated from a leaf in a template process tree.

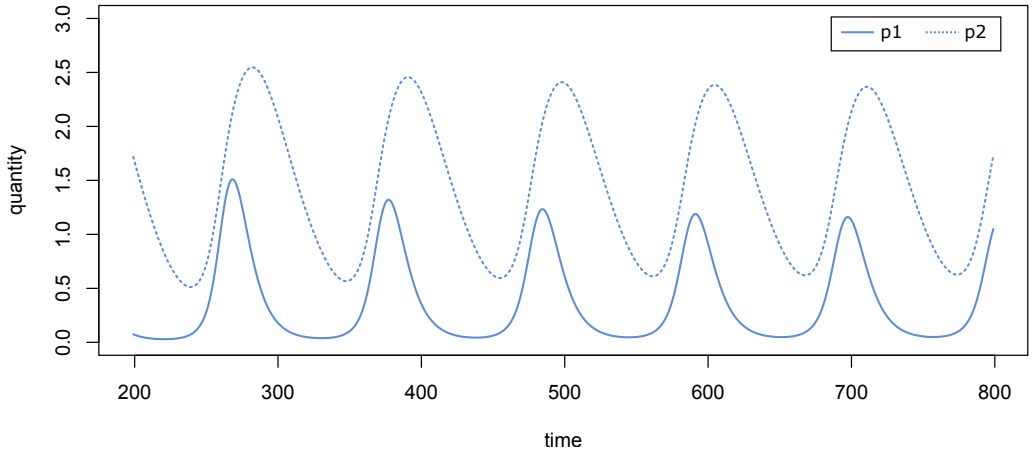

Figure 2.2: Simulation of the model of a genetic relaxation oscillator with structure and parameter values as shown in Table 2.3, 2.4, Equation 2.4 and output defined by the quantity of unbound proteins.

The model specification in ProBMoT introduces a requirement for *explicit constraints* based on instantiation of template processes. The model (complete or incomplete) cannot contain processes that are not explicitly instantiated from a template.

An *incomplete model* for process-based modeling with ProBMoT is used to explicitly define uncertainties in the model structure and parameter values. The structural uncertainty is reflected by the formal representation of a constrained model search space, using instances of non-terminal template processes. The uncertainty in the values of the parameters of a model is reflected by not assigning values to the corresponding instance parameters.

For example, when modeling a two node genetic oscillator with a partial knowledge of the principles of design of genetic oscillators, we might not be completely certain of the type of regulation mechanism between the two nodes. This uncertainty can be formally represented by encoding an incomplete model where the `g1activation` and `g2inhibition` instances are replaced by instances of the template processes `regulation`. The incomplete model containing these process instances defines a space of four possible model structures (the template process regulation has two mutually exclusive template process children), each with a different combination of regulatory processes. The values of the parameters of the `regulation` processes can be completely assigned – the incomplete model is used to define only structural uncertainty, or they can be partially or completely unassigned – the incomplete model defines uncertainty in both structure and parameter values.

Figure 2.3 depicts the most general ProBMoT workflow. At input it requires a library of domain knowledge, an incomplete model, a set of observations and a task specification. The task specification is composed of the type of task that needs to be performed, mappings of observations to process-based model variables, objective function specification and detailed settings for the model simulator and parameter estimator.

The only possible *task* for an input containing a complete model is the task of *model simulation* as shown previously. For an incomplete model two types of tasks can be defined. If the incomplete model contains at least one process instantiated from a non terminal node in a template process tree, the task is *model induction*. Model induction is performed by generating candidate model structures and estimating the parameter values for all candidate model structures, followed by model ranking and selection. If the incomplete model contains processes instantiated only from leaf nodes of a template process tree, but

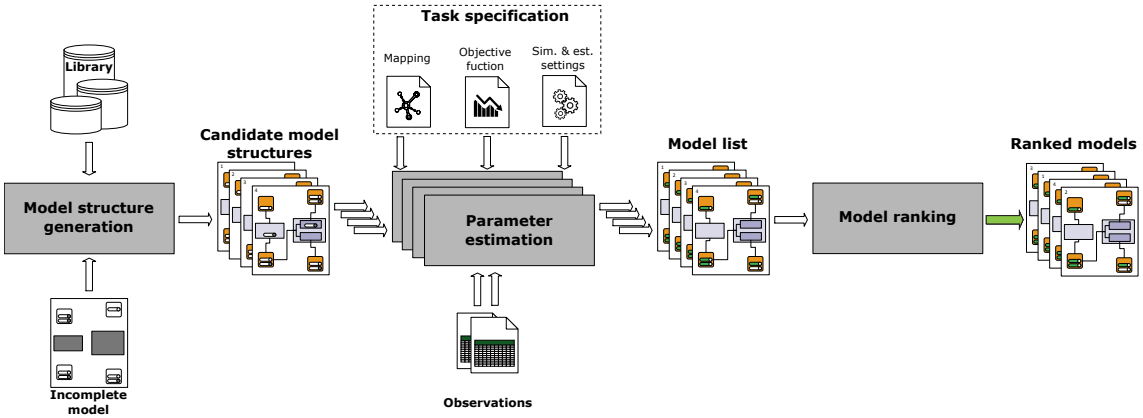

Figure 2.3: The ProBMoT workflow.

values for some of the variables or parameters are missing and need to be optimized, the task is *parameter estimation*.

Since the domain-knowledge is heavily exploited during the formalization of the search space, ProBMoT can take an "uninformed", naive approach towards a solution, i.e., search by exhaustive enumeration using process refinement. Processes are refined by their replacement with each leaf node from the subtree in which the template of the instantiated process belongs.

An important information for parameter estimation and simulation of a process-based model is contained within the observation-variable mappings. Apart from the mapping of a subset of the observations to the variables defined as exogenous, ProBMoT also requires explicit definition of the model output that relates system variables and constants to observations by arbitrary algebraic equations. The model output is considered for comparison with observations during the parameter estimation step. For parameter estimation ProBMoT uses the global meta-heuristic evolutionary optimization method Differential Evolution (Price, Storn, & Lampinen, 2005). The objective function, subject to minimization, used for parameter estimation is the sum of the root mean squared error between the model output and the observations.

In order to obtain an output by simulation of a complete process-based model or a candidate model structure with a candidate parameter set, the process-based model is first converted to a representation of a system of algebraic and ordinary differential equations by aggregation of specific process influences for each system variable, using the defined aggregation function and the fragments defined by each process. The system of algebraic and differential equations is then numerically solved using the CVODE implementation of the SUNDIALS suite (Hindmarsh et al., 2005). In particular, ProBMoT uses the Backwards Differentiation Formula and the generalized minimal residual method for a linear solver within the Newton iterations.

Chapter 3

Stochastic Process-Based Models of Dynamical Systems

3.1 Problem Description

A key feature of process-based models is their understandability/communicability. Process-based models provide a clear representation of the structure of a dynamical model in terms of its constituent entities and processes. The representation of the dynamical behavior of the processes is inherited from the library of domain-knowledge, where it is represented in terms of fragments of differential equations, which confines the resulting models to a coarse deterministic interpretation. The formalism used for encoding the library of domain knowledge and incomplete models is aimed at domain experts and is designed to facilitate the transfer of knowledge. However, encoding the knowledge by deriving specific fragments of differential equations for each process requires substantial effort and is not optimal in terms of understandability. Therefore, the existing formalism might not be the most adequate for application to modeling problems coming from the domain of life sciences, more specifically from systems and synthetic biology.

We address the issue of understandability and generality of interpretation by proposing a formalism for process-based modeling of stochastic dynamical systems based on the representation of processes by using reaction equations. The formalism retains the modularity of knowledge representation from previous process-based formalisms, but improves their understandability. It brings together the experts' finer grained view of the causal effects of interactions on one hand and the generality of process-based model on the other hand, allowing for a more realistic interpretation of the models that includes endogenous and exogenous stochastic effects.

Within the new formalism, the process templates are captured by a set of reaction equations in the form $R_s \rightarrow P_s$ $[rate]$. Here R_s and P_s are sets of reactant and product variables, i.e., variable properties of entities that are defined as arguments of the template process. The reactant/product variables are delimited by the + operator, and *rate* is the rate of the reaction that transforms the reactants into products.

Let us consider, for example, the relaxation oscillator from the previous chapter as depicted in Figure 2.1. The new formalism requires changes only in the form of the template equations within the library of domain knowledge. The template entities in the library of domain knowledge encoded by using the new formalism retain the form shown in Table 2.1. while the encoding of template processes by using reaction equations is shown in Table 3.1.

The representation of the model of the relaxation oscillator remains the same as shown in Table 2.3 and Table 2.4. There is, however, a significant change in the semantics of the representation.

The variable properties of an entity that take part as a reactant or a product in a reaction equation are state (reaction) variables. The properties that are defined by fragments of algebraic equations are auxiliary variables. An entity property cannot be a reaction and an auxiliary variable at the same time.

Table 3.1: Template processes defined in the library of domain-knowledge for transcriptional regulation using the formalism for representation of processes by reaction equations.

```
template process basic(g: Gene, m: mRNA, p: Protein){
  processes:
    transcription(g, m),
    translation(m, p),
    degradation(m),
    degradation(p);
}

template process transcription(g: Gene, m: mRNA){
  equations: g.quantity -> g.quantity + m.quantity [g.ktx];
}

template process translation(m: mRNA, p: Protein){
  equations: m.quantity -> m.quantity + p.quantity [m.ktl];
}

template process degradation(d: Product){
  consts: kd{range: <0.001, 10>};
  equations: d.quantity -> [kd];
}

template process regulation(p: Protein, g: Gene, pg: Complex, m: mRNA){
  consts:
    kf{range: <0.001, 10>}, kr{range: <0.001, 10>}, km{range: <2, 1000>};
  equations:
    p.quantity + g.quantity -> pg.quantity [kf],
    pg.quantity -> p.quantity + g.quantity [kr];
}

template process inhibition : regulation{}

template process activation : regulation{
  equations: pg.quantity -> pg.quantity + m.quantity [km*g.ktx];
}

template process dimerization(p: Protein, dp: Protein){
  consts:
    kf{range: <0.001, 10>}, kr{range: <0.001, 10>};
  equations:
    p.quantity + p.quantity -> dp.quantity [kf],
    dp.quantity -> p.quantity + p.quantity [kr];
}
```

The quantities of reaction variables are treated as discrete when considering the task of modeling a stochastic dynamical system, while the auxiliary variables are treated as continuous. The aggregation of influences for the reaction variables is not calculated for

each variable/property separately, but is performed at the level of all interacting properties, by composing a list of all instantiated reaction equations for a given process-based model. The influences for the auxiliary variables are aggregated according to the defined aggregation function, which by default is the summation function.

The representation of a process-based model is composed of an optional set of algebraic equations, one for each auxiliary variable, and a set of reaction equations. The process-based model of the relaxation oscillator contains no algebraic equations and the set of 18 reaction equations shown in Table 3.2. The reactions are grouped by the process that they represent.

Table 3.2: The set of reaction equations for the process-based model of a relaxation oscillator as shown in Table 2.3 and Table 2.4 derived by taking into account the library of domain knowledge shown in Table 2.1 and Table 3.1.

	Equation	Rate	Process
R_1	$g1.quantity \rightarrow g1.quantity + m1.quantity$	0.05	basic1.txg1
R_2	$m1.quantity \rightarrow m1.quantity + p1.quantity$	0.1	basic1.tlm1
R_3	$m1.quantity \rightarrow \emptyset$	0.2	basic1.dm1
R_4	$p1.quantity \rightarrow \emptyset$	0.4	basic1.dp1
R_5	$g2.quantity \rightarrow g2.quantity + m2.quantity$	0.05	basic2.txg2
R_6	$m2.quantity \rightarrow m2.quantity + p2.quantity$	0.1	basic2.tlm2
R_7	$m2.quantity \rightarrow \emptyset$	0.2	basic2.dm2
R_8	$p2.quantity \rightarrow \emptyset$	0.4	basic2.dp2
R_9	$p2.quantity + p2.quantity \rightarrow p2dimer.quantity$	0.1	dimer
R_{10}	$p2dimer.quantity \rightarrow p2.quantity + p2.quantity$	0.1	dimer
R_{11}	$p1.quantity + g1.quantity \rightarrow p1g1.quantity$	1	g1autoactivation
R_{12}	$p1g1.quantity \rightarrow p1.quantity + g1.quantity$	1	g1autoactivation
R_{13}	$p1g1.quantity \rightarrow p1g1.quantity + m1.quantity$	15	g1autoactivation
R_{14}	$p1.quantity + g2.quantity \rightarrow p1g2.quantity$	1	g2activation
R_{15}	$p1g2.quantity \rightarrow p1.quantity + g2.quantity$	1	g2activation
R_{16}	$p1g2.quantity \rightarrow p1g2.quantity + m2.quantity$	5	g2activation
R_{17}	$p2dimer.quantity + g1.quantity \rightarrow p2g1.quantity$	1	g1inhibition
R_{18}	$p2g1.quantity \rightarrow p2dimer.quantity + g1.quantity$	0.1	g1inhibition

The set of reaction equations can be decomposed into a stoichiometric matrix N, where each row corresponds to the total change in quantity of the reaction variables as a result of each reaction, and an array consisting of propensities of the reactions. The propensity of a reaction can be calculated directly from the form of the reaction equation. Given the state of the reaction variables x, the propensity $a_j(x)$ of each reaction R_j defines the probability that the reaction will be active in the infinitesimal time interval $[t; t + dt)$ as $a_j(x) = c_j \cdot h_j(x) \cdot dt$, where c_j denotes the reaction rate of the reaction R_j and $h_j(x)$ denotes the number of distinct combinations of the reactant quantities in the state x. In order not to restrict the expressiveness required for more complex or abstract models, in the formalism, a bang (!) operator may be added before the squared bracket in the reaction equation. The rate expression will then be treated as the propensity of the reaction, rather than its reaction rate.

Gillespie (1992) explicitly derived the Chemical Master Equation (CME) of a system from this representation. The CME represents a continuous time Markov chain that defines the evolution of the probability $P(x, t|x_0, t_0)$ that the system is in a state x at time t, given

the initial state x_0 at time t_0. It has the following form:

$$\frac{d}{dt}P(x,t|x_0,t_0) = \sum_{j=1}^{M}[a_j(x-\nu_j)\cdot P(x-\nu_j,t|x_0,t_0) - a_j(x)\cdot P(x,t|x_0,t_0)], \tag{3.1}$$

where ν_j is a row from the stoichiometric matrix N corresponding to reaction R_j.

Under the assumption that the change of the state in a sufficiently small interval does not significantly affect the propensities of the reactions while, at the same time, the number of reactions within this interval is much larger than 1, the CME of a dynamical system can be approximated by the Langevin equation of the system (Gillespie, 2000). The Langevin equation represents a system as a set of coupled Itō stochastic differential equations of the form:

$$\frac{dx}{dt} \approx \sum_{j=1}^{M}\nu_j\cdot a_j(x) + \sum_{j=1}^{M}\nu_j\cdot\sqrt{a_j(x)}\cdot\Gamma_j(t), \tag{3.2}$$

where $\Gamma_j(t)$ are temporally uncorrelated, statistically independent Gaussian white noises.

Under the assumption that the stochastic fluctuations do not significantly affect the behavior of the system, the CME can further be reduced to a system of ordinary differential equations representing the average behavior of the system:

$$\frac{dx}{dt} \approx \sum_{j=1}^{M}\nu_j\cdot a_j(x). \tag{3.3}$$

Although this is the most common representation of a dynamical system, the assumption that a negligible amount of noise is present within a system containing a large number of reactant molecules can still be considered as optimistic. For the model of a relaxation oscillator, the representation shown in Equation 2.4, under this assumption, can be derived from the encoding of the domain knowledge using the new formalism.

In addition to the improvement of the understandability and communicability of process-based models, the improved generality of process-based modeling with the reaction equation based formalism is evident from the possibility to use each of the aforementioned representations for learning and simulation of process-based models.

Within the implementation of the process-based modeling tool that includes the new formalism, we make available to the domain expert the choice of each level of representation allowing for both stochastic and deterministic interpretation. The simulation of a stochastic process-based model can be performed by drawing realizations of the CME using a Monte Carlo method such as the direct or the first reaction stochastic simulation algorithms proposed by Gillespie (1976) or by the numerous improvements and adaptations of these algorithms (Turner et al., 2004; Gillespie, 2007). For the simulation of process-based models we make available the direct and the first reaction methods, the next reaction method (Gibson & Bruck, 2000) and the τ-leaping algorithm (Gillespie & Petzold, 2003). The simulators are based on the Dizzy stochastic simulation software package (Ramsey, Orrell, & Bolouri, 2005) and are improved to be able to simulate models with external (exogenous) influences. We have also implemented the Euler-Maruyama method for the simulation of process-based models interpreted as a set of coupled stochastic differential equations.

In the previous approaches, an assumption was made that the observations used for model inference are obtained by a significantly large number of repeated experiments. As a result, the deterministic model simulation (which corresponds to the average behavior of the system) can be directly compared and fitted to observations. Although in a laboratory setting, the repeating of an experiment multiple times is an established practice,

the assumptions of the deterministic approximation of the behavior of the system do not hold, due to the small number of repeats or due to the nature and the properties of the dynamical system.

When inferring a stochastic model of a dynamical system, in addition to the better approximation of the nature and the properties of the dynamical system (available by different interpretations), the information about the experiment can be taken into account within the objective function used for parameter estimation. For example, if the data was obtained by averaging the observations from multiple repeats of an experiment, the same number of realizations of a stochastic model can be averaged in order to obtain a simulation that can be fairly compared to the observation. If the observations were obtained by a single experiment, the objective function might take this into account by calculating the fit of the model as the average of the fit of multiple single realizations of the model to the observations. If information about the distribution of observed values from a population of experiments is available at each point, the goodness of fit of the model can be then calculated by comparing these observations to the distribution of simulated values from a population of model realizations. In order to compensate for the stochastic nature of the system and stabilize the estimate, the parameter estimation can be restarted multiple times.

The evaluation of the method for learning stochastic process-based models of dynamical systems from knowledge and data is performed on four stochastic modeling tasks coming from the domain of genetic regulatory networks (GRNs) and epidemiology (Tanevski et al., 2016a). These include the reconstruction of GRNs with global and local kinetic rates and the learning of compartmental epidemiological models for the Eyam plague outbreak and the Trista da Cunha influenza outbreak.

This work was published in a journal article which constitutes the remainder of this chapter. The full bibliographic reference to the article is:

Tanevski, J., Todorovski, L., & Džeroski, S. (2016a). Learning stochastic process-based models of dynamical systems from knowledge and data. *BMC Systems Biology, 10*(1), 1-30.

3.2 Related Publication

Tanevski *et al. BMC Systems Biology* (2016) 10:30
DOI 10.1186/s12918-016-0273-4

BMC Systems Biology

METHODOLOGY ARTICLE Open Access

Learning stochastic process-based models of dynamical systems from knowledge and data

Jovan Tanevski[1,2*], Ljupčo Todorovski[3] and Sašo Džeroski[1,2]

Abstract

Background: Identifying a proper model structure, using methods that address both structural and parameter uncertainty, is a crucial problem within the systems approach to biology. And yet, it has a marginal presence in the recent literature. While many existing approaches integrate methods for simulation and parameter estimation of a single model to address parameter uncertainty, only few of them address structural uncertainty at the same time. The methods for handling structure uncertainty often oversimplify the problem by allowing the human modeler to explicitly enumerate a relatively small number of alternative model structures. On the other hand, process-based modeling methods provide flexible modular formalisms for specifying large classes of plausible model structures, but their scope is limited to deterministic models. Here, we aim at extending the scope of process-based modeling methods to inductively learn stochastic models from knowledge and data.

Results: We combine the flexibility of process-based modeling in terms of addressing structural uncertainty with the benefits of stochastic modeling. The proposed method combines search trough the space of plausible model structures, the parsimony principle and parameter estimation to identify a model with optimal structure and parameters. We illustrate the utility of the proposed method on four stochastic modeling tasks in two domains: gene regulatory networks and epidemiology. Within the first domain, using synthetically generated data, the method successfully recovers the structure and parameters of known regulatory networks from simulations. In the epidemiology domain, the method successfully reconstructs previously established models of epidemic outbreaks from real, sparse and noisy measurement data.

Conclusions: The method represents a unified approach to modeling dynamical systems that allows for flexible formalization of the space of candidate model structures, deterministic and stochastic interpretation of model dynamics, and automated induction of model structure and parameters from data. The method is able to reconstruct models of dynamical systems from synthetic and real data.

Keywords: Process-based modeling, Structural uncertainty, Dynamical systems, Stochastic models, Genetic regulatory networks, Compartmental epidemiological models

*Correspondence: jovan.tanevski@ijs.si
[1] Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
[2] Jožef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia
Full list of author information is available at the end of the article

Background

Most systems in biology exhibit dynamical behavior. Their properties change as a function of time and space in a complex manner. Considering a dynamical biological system to be a well-stirred mixture of its constituents, the most commonly used mathematical model of its dynamics takes the form of a system of coupled ordinary differential equations, treating the entity properties as continuous and assuming they evolve deterministically through time. However, the deterministic nature of ordinary differential equations renders them inadequate for systems with a small number of copies (only few orders of magnitude above one) of its constituents. Furthermore, ordinary differential equations fail to account for the underlying stochasticity of natural systems [1, 2]. In molecular systems, stochastic fluctuations are responsible for the divergence in phenotype and genetic activities [3–5]. In such cases, models based on stochastic kinetics are more suitable, as they allow for treating of the modeled systems as either discrete or continuous in terms of the properties of the observed entities and stochastic in terms of the reactions between them.

Establishing a deterministic or a stochastic model of an observed biological system is an omnipresent and often complex, tedious task. This task comprises the two subtasks of structure identification, i.e., selecting an appropriate model structure, and parameter estimation, i.e., determining values of the model parameters that, together with the selected structure, lead to accurate reconstruction of the observed system behavior. While many existing approaches integrate methods for simulation and parameter estimation of a single model, only few of them provide support for the task of structure identification [6, 7]. In this paper, we design and implement a computational tool that can deal with uncertainty in both model structure and the values of model parameters for both deterministic and stochastic models. The central component of our tool is the process-based modeling formalism that allows for modular, compositional specification of the space of candidate model structures.

Figure 1 puts the process-based modeling formalism in the context of existing formalisms used for modeling biological systems. The figure sorts (along the vertical axis) different formalisms according to their abilities to specify uncertainty with regard to the model parameter values and uncertainty with regard to the model structure. The vertical axis also refers to model specifications at different abstraction levels, from low-level model implementation to high-level model specification [8]. The horizontal axis refers to the possibilities of model interpretation: some of the formalisms are focused on deterministic, some on stochastic, while the third group of formalisms allows for both deterministic and stochastic model interpretation.

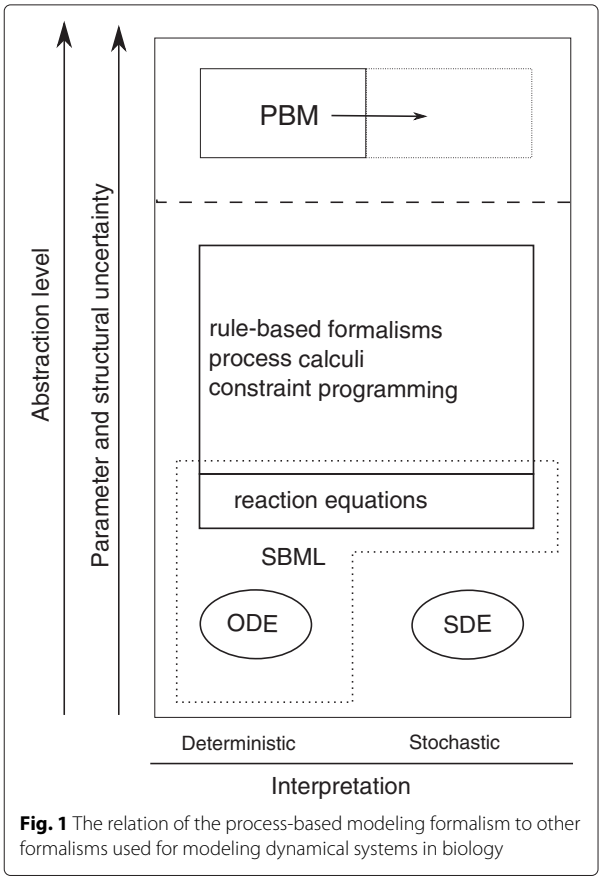

Fig. 1 The relation of the process-based modeling formalism to other formalisms used for modeling dynamical systems in biology

The formalisms of differential equations allow for encoding models with all the details needed for their execution, i.e., simulation of the behavior of the corresponding dynamical systems. Ordinary differential equations are limited to deterministic model interpretation, while stochastic differential equations are used for stochastic modeling. Differential equations are models at the lowest abstraction level, where every detail has to be fully specified and are used to encode a single model; on their own, they allow neither for parameter nor structural uncertainty.

At a higher abstraction level, the models in the domain of biology are often casted in the formalism of reaction equations. Following this formalism, the biological system is described as a reaction network. When coupled with appropriate kinetic rates, the model defines a network of possible transitions between system states. Reaction equations allow for both deterministic and probabilistic interpretation stemming from the propensity of each reaction [9].

The systems biology markup language, SBML [10], is a standard modeling formalism in system biology. It allows for encoding and exchange of individual models based on ordinary differential equations or reaction equations. Like equation-based formalisms, SBML focuses on encoding a single model structure with parameter uncertainties and does not support the specification of structural uncertainties.

Furthermore, a number of formalisms have emerged that deal with the issue of combinatorial complexity, i.e., the exponential complexity of the space of the combinations of elementary interactions between the entities observed in a given biological system. These formalisms allow for specifying rules (constraints) that limit the space of potential interactions between entities based on their properties. Note that the encoded constraints do not address the issue of structural uncertainty: Their application in the context of a given observed system leads to a single model structure. There are several classes of such formalisms.

The first group of rule-based (also referred to as interaction-based) languages, most notably BioNetGen [11] and kappa [12], define the constituent entities of a system at the level of objects with different properties. The network of interactions between the system entities is implicitly described by a set of rules that transform properties or create new entities (by forming complexes of existing entities). By defining the rules directly on the properties, the rule-based modeling approach efficiently deals with the problem of combinatorial complexity, which may arise when modeling protein-protein networks within complex signalization pathways. The rules are encoded using formalisms based on reaction equations.

The second group of agent-based formalisms, which includes process algebras [13, 14], model individual entities as agents in a complex system that act according to a set of predefined rules for communication with other agents. The process algebras describe the behavior of each agent through processes describing the inter-agent communications via different channels. A biological system described using process calculi is treated as a constrained distributed system of communication. This formal description allows for more detailed representation of the basic principles of interaction. Examples of process algebra extensions that have been adapted to and are being used in the domain of biology are the stochastic pi-calculus [15], Bio-PEPA [16] and beta binders [17].

Related to the process algebras group, the formalisms in the third group are based on constraint programming [18]. In contrast to the process calculi, the constraint programming approaches allow for defining interactions not only through specific communication channels, but by concurrently posting global constraints on the properties of the agent entities.

The limitation that is common to all aforementioned formalisms is that they can not properly represent the structural uncertainty. Uncertainty in parameter values is typically addressed by various formalism extensions that are complementary to the computational tools that offer support for them. COPASI [19] is an example of a such a tool that allows for introducing uncertainties in model parameter values and performing parameter estimation for models based on equations. The MathWorks SimBiology toolbox [20] is a proprietary software for modeling and analysis of dynamical systems in biology providing features similar to the ones of COPASI. Both tools provide a range of methods for the analysis of models (e.g., sensitivity and identifiability of model parameters), but do not provide computational methods for addressing structural uncertainty; users can only perform manual comparative analysis of different model structures.

Network inference methods [21] explicitly address structural uncertainty: most often, given gene expression data, the methods seek for a network of interactions between the observed genes. Since these methods focus on the structure of the observed network of interactions, they seldom deal with the reconstruction of the dynamical behavior of the observed system. Several methods are exception to this general rule and cast the reconstructed networks into the formalism of ordinary differential equations [21–24]. In contrast to the process-based modeling approach presented in this paper, these methods are limited to deterministic models. Furthermore, these methods follow the assumption that the same interaction dynamics applies to all of the network interactions: the process-based modeling formalism can encode different classes of model structures (interactions/processes) with different assumptions about the interaction dynamics. Finally, when it comes to constraining the space of possible model structures, some methods employ data-driven heuristics [22, 25], while some of them additionally limit the search for plausible structures based on the interactions already documented in the literature [22, 24]. The method by Wahl et al. [22] also allows for user-defined Boolean constraints specifying implausible network interactions.

Finally, ABC-SysBio [6] is most closely related to the process-based modeling approach presented here. It builds on SBML and addresses structural uncertainty by allowing the user to explicitly enumerate the alternative model structures. The process-based modeling formalism the we propose addresses exactly this limitation of the existing formalisms, i.e., the ability to properly address structural uncertainty. It allows for modular and flexible specification of the space of candidate model structures to be considered in the modeling process. Instead of

specifying a fixed list of candidates, like ABC-SysBio [6], our formalism allows users to specify model components, which are then used in a compositional modeling setting, where combinations of components correspond to candidate model structures. Thus, we approach the structure identification task as a search problem [26], where we search for the most appropriate combination of model components.

This paper builds upon our previous work on inductive process-based modeling that combines knowledge and data to automatically build explanatory models of dynamical systems [27–30]. While inductive process-based modeling has been successfully applied to modeling tasks in the domain of systems biology [7, 31, 32], its scope has been limited to building deterministic models of dynamical systems cast as ordinary differential equations.

Here, we extend the scope of the process-based modeling formalism to models cast as reaction equations, hence the arrow in the top-right corner of Fig. 1. In this way, we combine the benefits of process-based modeling (in terms of addressing structural uncertainty) with the benefits of different model interpretations (including the stochastic interpretation). Finally, the formalism is implemented within a computational tool ProBMoTs for automated induction of models that combines domain knowledge represented in our formalism with measurements of the observed system behavior.

In the remainder of the paper, we first introduce the process-based modeling formalism, its extensions towards handling stochastic models of biochemical systems and the computational tool for process-based modeling, ProBMoTs. We present then two examples of use of the proposed computational tool, i.e. modeling gene regulatory networks and modeling the spread of pathogens, illustrating the use of the proposed tool and evaluating its utility. Finally, we discuss the results of the evaluation, put them in the context of existing work and outline directions for further research.

Methods

In this section, we introduce the notion of process-based models and a formalism for their representation. We illustrate the formalism use on an example of encoding knowledge for modeling gene regulatory networks and a process-based model of a specific network, the repressilator [33]. We then introduce methods for inducing process-based models from knowledge and data by selecting appropriate model structures and parameters.

Process-based modeling

Scientists often describe dynamical systems in terms of processes that govern the system dynamics and the entities involved in the processes[1]. Following this high-level model description, modelers assign lower-level detailed equation-based specifications of the dynamics to individual processes and combine them into a system of coupled differential equations. The differential equations can be in turn used to simulate the behavior of the observed system or to extrapolate the simulation and predict future system behavior. However, by transforming the high-level model description into equations, its explanatory power is lost, since the equations fail to reveal (in an accessible manner) the structure of the observed system in terms of the interacting entities and processes.

Process-based modeling (PBM) clearly relates a high-level model description (entities and processes), that carries significant explanatory power, and a lower-level mathematical model (equations), that allows for simulation and prediction. To build process-based models, we first formalize the modeling knowledge by establishing templates of generic (template) entities that appear in the generic (template) processes that govern the dynamics of systems in the particular domain. Each process-based model then refers to these template components and instantiates them into specific components of the studied system.

Existing process-based formalisms rely on a coarse description of dynamics, based on fragments of differential equations. The formalism introduced in this paper relies on reaction equations, which are closer to the basic principles of system biology and are more comprehensible to biologists. A reaction equation $R_s \rightarrow P_s$ [*rate*] specifies a set of reactants R_s and a set of products P_s, as well as the reaction *rate*. Reaction equations are a powerful and flexible formalism for modeling the temporal evolution of dynamical systems.

Representation of modeling knowledge

Table 1 provides an example library of template components for modeling gene regulatory networks. It includes a template entity *gene*, whose instances represent nodes in gene regulatory networks. We assume gene entities to represent protein-coding genes and describe them using five numerical properties. The variable properties (`vars` section of the entity specification) denote two gene properties that change through time: `Pmol` is the number of encoded protein molecules and `mRNAmol` is the number of mRNA transcripts. The other three properties do not change over time; they denote the constant kinetic rates of the uncontrolled gene expression `alpha0`, the translation of mRNA into proteins and their degradation `beta`, as well as the mRNA molecules degradation `delta`.

Furthermore, the library specifies templates for modeling the processes of gene interaction, gene translation into proteins, and protein degradation. The `degradation` template specifies two reaction equations that correspond to the degradation of the encoded protein molecules with the kinetic rate of `g.beta` (i.e., the degradation kinetic

Table 1 Templates of entities and processes for modeling gene regulatory networks. The template entity *gene* typifies network nodes, while the process templates represent gene regulation, as well as translation and protein degradation processes. The empty set symbol Ø denotes the absence of reactants or products

```
template entity gene{
    vars: mRNAmol, Pmol;
    consts: alpha0, beta, delta;
}

template process regulation(sg:gene, dg:gene){
    consts: alpha, n;
}
template process inhibition: regulation{
    equations: Ø -> dg.mRNAmol [alpha / (1 + sg.Pmol^n)];
}
template process activation: regulation{
    equations:
        Ø -> dg.mRNAmol [alpha - (alpha / (1 + sg.Pmol^n))];
}

template process translation(g:gene){
    equations:
        Ø -> g.mRNAmol [g.alpha0],
        g.mRNAmol -> g.Pmol + g.mRNAmol [g.beta];
}
template process degradation(g:gene){
    equations: g.Pmol -> Ø [g.beta], g.mRNAmol ->
    Ø [g.delta];
}
```

rate for the particular gene `g`) and the degradation of the mRNA molecules with the rate of `g.delta`. Similarly, the `translation` process integrates the reaction equations of the gene transcription to mRNA and the mRNA translation to protein molecules.

Finally, the `regulation` process template represents gene interactions via their protein products. It has two mutually exclusive alternatives of `activation` and `inhibition`. The first corresponds to the case where one gene increases the transcription rate of the other, while the second alternative models repression, where one gene decreases the transcription rate of the other by binding the source gene protein to the promoter region of the repressed gene. In both cases, the reaction rate (specified between the brackets) is modeled using a Hill function, derived as a steady-state approximation of the biochemical kinetics [34].

The templates from Table 1 represent generic knowledge on modeling gene regulatory networks. They can be instantiated to entities and processes of an arbitrary network model. Note the hierarchical structure of the `regulation` template process: it constrains the space of instantiations by rendering the two subordinate templates of `activation` and `inhibition` mutually exclusive. This reflects the simple fact that only one regulation type applies to a given pair of genes. In the following, we will illustrate the use of this knowledge for modeling a simple regulatory network.

Process-based models

The repressilator [33] is a regulatory network of three genes interacting in a single feedback loop of inhibitions as depicted in Fig. 2. The repressilator is a synthetic network designed to exhibit a stable oscillatory behavior. Its in-vivo implementation in *E. coli* has been proven to exhibit the desired behavior. The three genes involved are TetR, often used for fine regulation in synthetic gene networks, and two repressor genes, cI and LacI.

Using the domain knowledge for modeling gene regulatory networks from Table 1, we can establish a process-based model of the repressilator, presented in Table 2. It provides a high-level representation akin to the graphical network layout depicted in Fig. 2, where entities correspond to network nodes, and processes are represented by arcs. The model does not give details about the particular modeling choices for degradation, translation and inhibition, since they are inherited from the corresponding process templates. Each entity specifies the boundary conditions for the variables (declarations of the `initial` value) and the parameter values, while each process specifies the involved entities and the parameter values. Note, for example, the value assignments for the parameters `alpha` and `n` in the inhibition processes.

The process-based model retains the understandability of the graphical model representation and provides a clear, high-level insight into the structure of the studied system. At the same time, by using the detailed knowledge of the reaction equations encoded in the templates, we

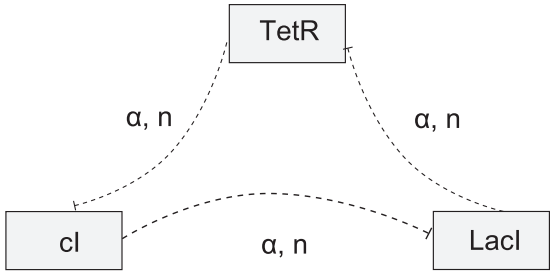

Fig. 2 Graphical representation of the repressilator gene regulatory network

Table 2 A process-based model of the repressilator built using the templates for modeling gene regulatory networks from Table 1

```
entity TetR:gene{
    vars: Pmol{initial: 5;}, mRNAmol{initial: 0;};
    consts: alpha0 = 0, beta = 9.75, delta = 1;
}
entity LacI:gene{
    vars: Pmol{initial: 0;}, mRNAmol{initial: 0;};
    consts: alpha0 = 0.27, beta = 10, delta = 1;
}
entity cI:gene{
    vars: Pmol{initial: 15;}, mRNAmol{initial: 0;};
    consts: alpha0 = 0.41, beta = 10, delta = 1;
}

process regualtion1(TetR, cI): inhibition{
    consts: alpha = 407, n = 3;
}
process regulation2(cI, LacI): inhibition{
    consts: alpha = 222, n = 4.7;
}
process regulation3(LacI, TetR): inhibition{
    consts: alpha = 237, n = 1.95;
}

process translation1(TetR): translation{}
process translation2(LacI): translation{}
process translation3(cI): translation{}

process degradation1(TetR): degradation{}
process degradation2(LacI): degradation{}
process degradation3(cI): degradation{}
```

Table 3 List of reaction equations stemming from the process-based model of the repressilator from Table 2

```
∅ -> TetR.mRNAmol[407 / (1 + cI.Pmol^3)]
∅ -> LacI.mRNAmol[222 / (1 + TetR.Pmol^4.7)]
∅ -> cI.mRNAmol[237 / (1 + LacI.Pmol^1.95)]

∅ -> TetR.mRNAmol[0]
TetR.mRNAmol -> TetR.Pmol + TetR.mRNAmol[9.75]
∅ -> LacI.mRNAmol[0.27]
LacI.mRNAmol -> LacI.Pmol + LacI.mRNAmol[10]
∅ -> cI.mRNAmol[0.41]
cI.mRNAmol -> cI.Pmol + cI.mRNAmol[10]

TetR.Pmol -> ∅[9.75]
TetR.mRNAmol -> ∅[1]
LacI.Pmol -> ∅[10]
LacI.mRNAmol -> ∅[1]
cI.Pmol -> ∅[10]
cI.mRNAmol -> ∅[1]
```

can automatically translate the high-level description into a mathematical model and use it for simulation and analysis. Consider the process `translation1` in Table 2: By combining it with the template `translation` from Table 1, we instantiate a set of two reaction equations modeling the uncontrolled transcription of *TetR* to mRNA (`∅ -> TetR.mRNAmol` with a kinetic rate of `TetR.alpha0 = 0`) and the translation of the mRNA to the *TetR* protein molecules (`TetR.mRNAmol -> TetR.Pmol + TetR.mRNAmol` with a kinetic rate of `TetR.beta = 9.75`).

Table 3 presents the mathematical model of the repressilator which includes the above two reaction equations, as well as all other reaction equations obtained by combining the processes in Table 2 with their corresponding templates from Table 1. The model is simulated by calculating the state of the system $x(t)$, a vector of the number of molecules of each reactant at time t. The repressilator state includes six variables: `TetR.Pmol`, `TetR.mRNAmol`, `LacI.Pmol`, `LacI.mRNAmol`, `cI.Pmol` and `cI.mRNAmol`. In any given state x, we can calculate the propensity, i.e., the probability that the reaction R_j will be active in the infinitesimal time interval $[t, t + dt)$, using the formula $a_j(x) = c_j h_j(x) dt$, where c_j denotes the reaction rate and $h_j(x)$ denotes the number of distinct combinations of reactant molecules in state x.

The evolution of the probability $P(x, t|x_0, t_0)$ that the system is in a state x at a given time t, given the initial state x_0 at time t_0, can be then defined using the following ordinary differential equation (also known as the Master Equation) [35]:

$$\frac{\partial}{\partial t}P(x,t|x_0,t_0) = \sum_{j=1}^{M}\left[a_j(x-\nu_j)P(x-\nu_j,t|x_0,t_0) - a_j(x)P(x,t|x_0,t_0)\right], \quad (1)$$

for $dt \to 0$, where ν_j is a vector specifying the changes of the number of reactant molecules after the reaction R_j. We can then model the system dynamics using coupled differential equations, where each equation models the probability that the system state equals a unique combination of values of the state variables x.

For real biological systems, the Master Equation is too complex to be solved analytically or numerically. To this end, alternative approaches to estimating the exact or approximate probabilities have been developed. One of the most popular exact approaches is based on Monte Carlo sampling and is known as the Stochastic Simulation Algorithm (SSA) proposed by Gillespie, where others include the Gibson-Bruck method of next reaction and the class of τ-leaping methods [9].

If we assume that the propensity does not significantly change in infinitesimal time intervals and that the expected number of firings of each reaction is significantly large (i.e., the number of reactant molecules is large compared to the probability rate constant), we can derive the Langevin Equation. It represents a mathematical model of the reaction equations cast in terms of coupled Itō stochastic differential equations [36]. These stochastic differential equations can further be reduced to ordinary differential equations, under the assumption that we observe a negligible amount of noise in a system with a large number of reactants.

Thus, from a process-based model, we can automatically infer the reaction equations and then simulate them using the Gillespie algorithm or its improvements [9]. Alternatively, we can transform them to a system of ordinary differential equations. Figure 3 shows the simulated trajectories of the number of *TetR* molecules obtained by simulating the reaction equations (left-hand side) and the system of ordinary differential equations (right-hand side) inferred from the process-based model of the repressilator.

To summarize, process-based models have four important properties that make them particularly suitable for modeling dynamical systems. First, they retain the *understandability* and explanatory power of graphical model representations by providing clear insight into the structure of the observed system. At the same time, they inherit the *utility* of mathematical models for simulation and analysis of system behavior. Third, process-based models provide *general* model descriptions that support both stochastic and deterministic approaches to modeling, simulation and analysis. The fourth property is the *modularity* provided by the knowledge representation formalism: the templates can be instantiated into a number of model components. This last property is particularly relevant for the algorithms that induce process-based models from data.

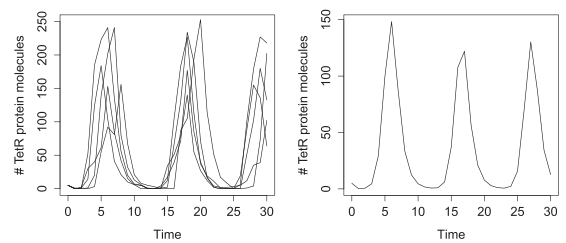

Fig. 3 Stochastic and deterministic simulation of the number of TetR protein molecules using the process-based model of the repressilator

Inducing process-based models

The formalized knowledge on modeling gene regulatory networks brings another benefit. It represents a source of constraints that limit the space of candidate model structures to be explored when modeling a particular gene regulatory network. Consider the repressilator model again and assume that we are only provided with information that it involves the three genes of *TetR*, *LacI* and *cI*. Now we can infer all the instances of the process templates from Table 1: the `degradation` process template that involves one gene, leads to three process instantiations, one for each gene. Similarly, the `translation` template leads to three processes. Finally, each pair of genes results in one instance of the *activation* and one instance of the *inhibition* template. Thus, for the three repressilator genes, we obtain six instances of the *activation* and six instances of the *inhibition* template. In sum, the three repressilator genes lead to 18 process instances.

Each of the process instances represents a valid model component. Following a naïve approach, one can consider any subset of components as a legitimate model structure, which yields $2^{18} = 262,144$ candidates. However, these include many implausible models, e.g., ones that do not include gene translation for some of the genes. To avoid implausible models, the inductive process modeling approach relies on the use of constraints that limit the ways model components are combined. For example, a constraint ruling out models that do not include translation and degradation processes for all the genes, reduces the search space to $2^{12} = 4096$ candidates. Furthermore, the constraint specifying the mutual exclusivity of the activation and inhibition processes for a given ordered pair of gene entities further reduces the number of candidates to $3^6 = 729$ (for each of the six possible pairs of repressilator genes, we consider three modeling alternatives: absence of regulatory influence; activation; and inhibition).

The constraints discussed above can be classified in two groups. First, the mutual exclusivity of the activation and inhibition processes is specified in the domain knowledge library shown in Table 1. Second, the constraint ruling out models that do not include translation and degradation of individual gene/protein are defined at the level of process instances. The constraints from the second group are specified in the incomplete model, which is one of the inputs to our software tool ProBMoTs. One

such incomplete model is depicted graphically in Fig. 5 and shown in Table 4. The lower part of the table specifies that the model must include both a translation and a degradation process for each of the three genes/proteins (note the mandatory qualifier in the process specifications). Figure 5 (and the upper part of Table 4) specifies the three modeling alternatives for each of the six possible pairs of genes.

Table 4 An incomplete process-based model of a gene regulatory network specifying the model structures as depicted in Fig. 5

```
entity TetR:gene{
    vars: Pmol{initial: 5;}, mRNAmol{initial: 0;};
    consts: alpha0, beta, delta = 1;
}
entity LacI:gene{
    vars: Pmol{initial: 0;}, mRNAmol{initial: 0;};
    consts: alpha0, beta, delta = 1;
}
entity cI:gene{
    vars: Pmol{initial: 15;}, mRNAmol{initial: 0;};
    consts: alpha0, beta, delta = 1;
}

process regualtion1(TetR, cI): regulation{
    consts: alpha, n;
}
process regulation2(cI, LacI): regulation{
    consts: alpha, n;
}
processregulation3(LacI, TetR): regulation{
    consts: alpha, n;
}
processregualtion4(TetR, LacI):regulation{
    consts: alpha,n;
}
processregulation5(LacI, cI):regulation{
    consts: alpha,n;
}
process regulation6(cI, TetR): regulation{
    consts: alpha,n;
}

mandatory process translation1(TetR):translation{}
mandatory process translation2(LacI): translation{}
mandatory process translation3(cI):translation{}

mandatory process degradation1(TetR):degradation{}
mandatory process degradation2(LacI):degradation{}
mandatory process degradation3(cI): degradation{}
```

Finally, the inductive process-based modeling approach validates each candidate model structure by matching its simulation against the observed system behavior. In order to simulate the model (and assess its quality), we first have to determine the values of its constant parameters. To this end, we employ parameter estimation and find parameter values that lead to a model reproducing the observed behavior as closely as possible. We formulate the parameter estimation task as an optimization problem: We aim at minimizing an objective function that measures the goodness of fit of the model simulation to the observed behavior using the maximum-likelihood estimator [37].

The algorithm for inducing process-based models, presented in Table 5, puts together the components outlined above. Its input is a *library* of template entities and processes, such as the one presented in Table 1, the specific *entity* instances observed in the system at hand, a set of *constraints* that limit the way we combine components into models, and time-series *data* comprising measurements of the system variables/outputs of observed system. The algorithm first instantiates the templates from the library using the entities of the observed system into a set of model *components*. Then, taking into account the constraints, the algorithm enumerates the plausible combinations of components as candidate model structures. For each model structure, the algorithm performs *parameter estimation* that fits the model simulation against observed data. At output, the algorithm returns a list of models ranked with respect to their fit against the measured data.

Different implementations of the induction algorithm make different design choices. In the following, we provide a brief overview of the different implementations: A detailed overview is given by Džeroski and Todorovski [7]. Lagramge 2.0 [38] transforms the library and constraints into a grammar that enumerates candidate model

Table 5 Top-level outline of the algorithm for inducing process-based models from knowledge and data

```
procedure IPM(library, entities, constraints, data):
    components = Instantiate(library, entities)
    for mstructure in Enumerate(components,
    constraints):
        (model, error) = ParameterEstimation(mstructure,
        data)
        append(model_list, (model, error))
    sort(model_list, key = error)
    return model_list
```

structures. IPM [28] takes a naïve approach and uses constraints on the number of components involved in the model to address combinatorial explosion. HIPM [29] encodes the constraints into a hierarchy of process templates and approaches enumeration as a combinatorial search problem. SCIPM [39] explicitly encodes the constraints and approaches the enumeration using constraint satisfaction methods. Finally, ProBMoT [40] extends HIPM with explicit constraints referring to the particular system at hand and meta-heuristic optimization methods for parameter estimation.

Note, however, that the above inductive process-based modeling approaches have limited their focus on inducing deterministic models cast as ordinary differential equations. ProBMoTs, our extension of ProBMoT presented in this paper that allows for inducing stochastic models of dynamical systems cast as reaction equations. The extension is based on the novel formalism for encoding a library of components that supports the specification of reaction equations as models of individual processes. ProBMoTs also integrates standard simulators for reaction equations [41].

Both ProBMoT and ProBMoTs are released as open-source software packages available for download at http://probmot.ijs.si [2].

Experimental setup and model selection

To evaluate the algorithm for inducing stochastic process-based models, we apply it to several problems of modeling dynamical behavior of biological systems at different scales. We consider two synthetic modeling problems from the domain of gene regulatory networks and two real modeling problems from the domain of epidemiology. In each domain, we first encode process-based knowledge for modeling dynamical systems. In this paper, our focus is limited to encoding domain knowledge in two domains, covering models on fundamentally different scales. Note, however, that the process-based modeling approach can be applied to other domains as well, given that modeling knowledge about the domain of interest is encoded as a library of entity and process templates. For example, when modeling metabolic networks, the central entity templates will represent enzymes and metabolites, while process templates would represent different metabolic reactions (with different kinetics), formulating different models of the dynamical interactions among them. For further examples of domain knowledge libraries for process-based modeling, we refer the reader to the ProBMoT web site. Second, for the synthetic modeling problems, we select a target model and simulate it to obtain a data set for inducing models. On the other hand, the real modeling problems come with data sets of measured system behavior. Third, for each modeling problem, we define an ordered list of plausible model structures P. For the synthetic problems, this list includes the target model only, while for the real modeling problems, it includes all the structures of the models that have been reported in the literature as plausible explanations of the measurements. Note that for all problems, the list of candidate models considered by the induction algorithm includes all the model structures from the list P.

To perform induction, for each modeling problem we run ProBMoTs using the corresponding modeling knowledge (including the constraints) and the data set as inputs. Recall that the modeling knowledge defines the space of candidate model structures. The values of the model parameters are estimated by using the Differential Evolution method [42] with the recommended parameter settings: crossover probability of 0.9, differential weight of 0.8, population size 50 and the *rand/1/bin* strategy. We set the number of evaluations of the objective function to 1000 times the number of constant model parameters. To assess the stability of the parameter estimator, we use 10 restarts of the Differential Evolution method. For simulating the reaction equations, ProBMoTs employs the Gillespie direct method [9] to obtain 20 realizations.

The parameter estimation method in ProBMoTs can use different objective functions for measuring the discrepancy between the realizations and the observed data. The first objective function we use in the experiments corresponds to a typical laboratory setting used in biology, where the measurements from multiple replicates of an experiment are averaged. Thus, the 20 realizations (K in the equation) are averaged just as the observed data:

$$RMSE_{AR}(m) = \sum_i \frac{1}{\sqrt{N}} \|x_i - \widehat{x}_i\|, \quad \widehat{x}_i = \frac{1}{K} \sum_k \widehat{x}_i^k, \quad (2)$$

where m denotes the model, i iterates over the observed variables x_i and k iterates over the realizations, where $\widehat{x}_i^k$ denotes the k-th realization of x_i, and N is the number of observed time points.

Alternatively, in situations where the data are measured within a single experiment, we use the second objective function. Instead of averaging the realizations, we average the error of each realization, i.e.:

$$RMSE_{SR}(m) = \frac{1}{K} \sum_k \sum_i \frac{1}{\sqrt{N}} \|x_i - \widehat{x}_i^k\|. \quad (3)$$

Recall that the result of ProBMoTs is a list of models ranked with respect to their descending fit against the measured data, in our case, ascending model error. The trivial model selection strategy would be to select the model with the optimal value of the objective function. Note, however, that error-based estimates of model performance tend to overfit observations, a problem especially relevant in the context of noisy experimental data. To address the problem of overfitting, we use an alternative model selection approach that introduces a penalty

for model complexity, measured as the number of reaction equations in the model. To additively combine the model complexity and the degree of fit into a single *score*, we normalize both to the [0, 1] scale. The normalization is based on the minimal and maximal values of the degree of fit and complexity, respectively, over all the candidate model structures considered by ProBMoTs.

We visualize the result of ProBMoTs (i.e., the ranked list of models) using an *error profile*, as depicted in Fig. 4. Each point of the error profile corresponds to a model induced by ProBMoTs and the y-axis of the profile corresponds to the respective value of the model selection criterion. In our experiments, we use the error profile to evaluate the ProBMoTs results in two ways. The first one selects the left-most model in the error profile, i.e., the model with the lowest model selection score, as the most appropriate model. We refer to this method as the *singular* method. This method is short-sighted since it only considers the best model. As an alternative to the *singular* method, we propose the *inclusive* method that considers models in the left-most plateau of the error profile. We employ a simple heuristic to identify plateaus: a relative change of error between two consecutive error-profile points that is above a threshold value of 0.1 indicates a plateau end. The first (leftmost) plateau of the error profile in Fig. 4 includes the cluster of ten points in the lower-left corner of the graph. Note that it includes ten top-ranked model structures that are indistinguishable in terms of the model error and therefore better represent the results of induction. The plateaus of the error profile lead to a partial ordering of the models.

Finally, to evaluate the results of induction, we compare the list of selected models to the list of plausible models P using a triple of metrics (*recall, hit, plateau_size*). The *recall* is the proportion of the plausible models in the first plateau of the error profile. The indicator *hit* tells us whether the first plateau contains the first model structure in P. The size of the plateau (*plateau_size*) indicates the discriminative power of the induction method: the smaller the plateau, the larger the discriminative power. The ideally performing induction method would lead to the triple (*100 %, true,* $|P|$).

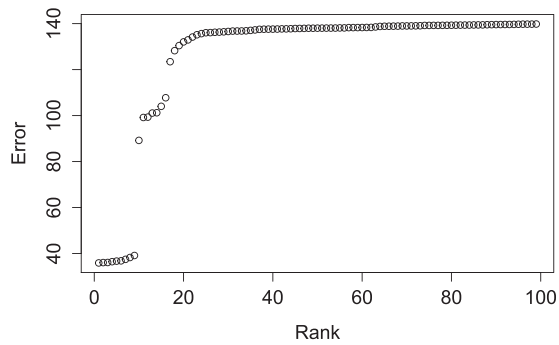

Fig. 4 An example error profile of a ProBMoTs output that includes 100 models ranked according to increasing model error

Results

In this section, we present the results of the evaluation of ProBMoTs on the four problems of inducing stochastic process-based models from knowledge and data. The first two are from the domain of gene regulatory networks, the other from the domain of epidemiology.

Gene regulatory networks

We first address the task of modeling the simple gene regulatory network of the repressilator, introduced in the previous section. We select the model from Table 3 as a target model and set the list of plausible model structures P to contain a single structure that corresponds to the target model. We then perform two experiments. In the first, we assume that the kinetic rates in processes belonging to a single class of regulatory processes (degradation, translation and regulation) have the same values. To this end, we restructure the library of templates to introduce an *global* template entity that declares the global kinetic rates, which are then used by the process templates. In the second experiment, we perform induction without the assumption of global kinetic rates and therefore use the library of templates as presented in the previous section.

Global kinetic rates

The model of the repressilator considered here has been already addressed in other studies [6, 43]. Note, however, that both studies address only the task of parameter estimation from synthetic data assuming a single model structure. In our experiment, we also aim at identifying the structure of the model. We select the single model structure used in previous studies as our target and use the following values of the global kinetic rates: $(alpha0, alpha, beta, delta, n) = (0.0, 250.0, 5.0, 1.0, 2.1)$. To obtain experimental data, we average 20 realizations of the target model in the time interval $t \in [0, 35]$. Accordingly, we use the $RMSE_{AR}$ objective function.

In order to define a structure identification problem, we describe the space of possible model structures as represented in Fig. 5. Each rectangle represents a gene entity, while the dashed lines represent a regulation interaction between the entities. The interactions in the incomplete model are instantiated from the `regulation` process template from Table 1. This results in $3^6 = 729$ possible model structures, one of which is the target model structure of the repressilator.

Figure 6 depicts the error profile for the list of models obtained with ProBMoTs. First, note that the small standard deviations across the restarts of the parameter

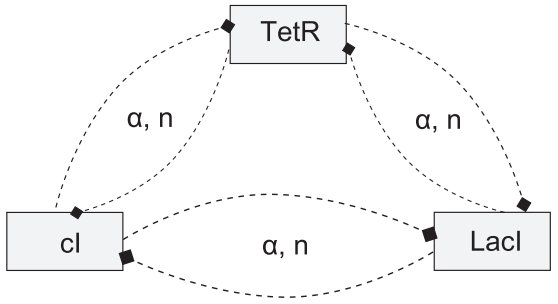

Fig. 5 Graphical representation of the space of model structures considered during the induction of the repressilator model. Note that we do not assume fixed forms of the regulation interactions between the genes

estimator show its stability. Furthermore, the first plateau of the error profile is easy to identify in the lower-left corner of the figure: it contains a single model. The structure of this model is a perfect match to the structure of the target model. Therefore, the recall is *100 %*, the hit is *true* and the plateau size is *1*, or in other words, the performance of ProBMoTs on this task is ideal. This result gives proof-of-principle evidence that confirms the ability of the developed process-based modeling method to induce both the structure and parameters of stochastic models from knowledge and data.

Local kinetic rates

To test the robustness of our method, we remove the assumption of global kinetic rates from the modeling scenario. Thus, we forget the changes we made to the library in the previous experiment and use the library as described in Table 1. Other than the different formalization of the domain knowledge, given the relaxed assumptions, the task remains the same: we use the same target model, the data set, the objective function ($RMSE_{AR}$) and the list of plausible models as in the first experiment. The relaxed assumptions lead to an explosion in the parameter space, while the structure space remains the same. We want to test whether (and how) the relaxed modeling assumption will influence (deteriorate) the results of ProBMoTs.

The obtained error profile for the described task is shown in Fig. 7; note again the small standard deviation of the error over the parameter estimator restarts. The first plateau of the error profile includes four models. The second model has the structure that exactly matches the structure of the target model leading to the performance triple of (*100 %*, *true*, *4*). The structures of the other three models in the plateau contain the repressilator motif and a number of additional gene regulation interactions, indicating an overfit of the experimental data. Indeed, Fig. 8 shows that if model complexity is taken into account when selecting models, the first plateau of the error profile includes only the target model, leading to the ideal performance triple of (*100 %*, *true*, *1*).

Compartmental epidemiological models

In the domain of epidemiology, we first formalize the knowledge to be used for establishing stochastic models, using the basic principles of compartmental modeling as presented by Brauer et al. [44]. There, the spread of disease is modeled by the flows of individuals between healthy and infected populations, referred to as compartments. Each flow is modeled using a reaction equation, where reactants and products correspond to compartments.

Figure 9 graphically illustrates the general structure of epidemiological compartmental models. We distinguish between six compartments corresponding to six

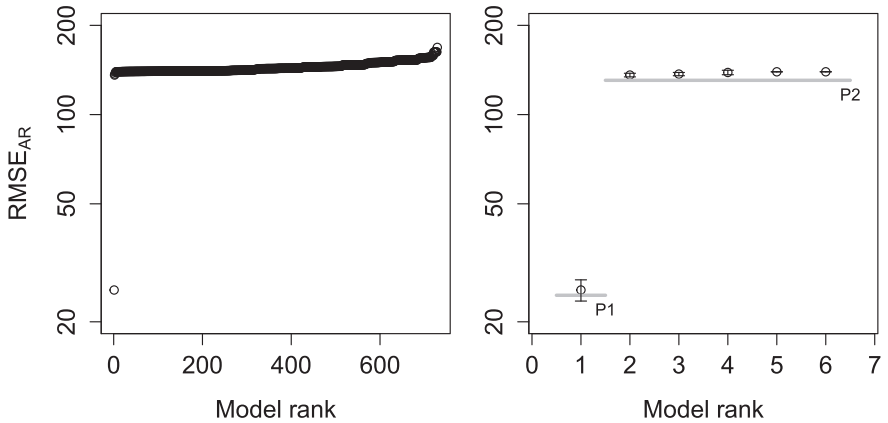

Fig. 6 Error profile for the task of inducing the repressilator model with global kinetic rates. Complete error profile (*left*). The top six models in the first two plateaus with error bars showing the standard deviation across restarts. The gray horizontal lines depict plateaus (*right*)

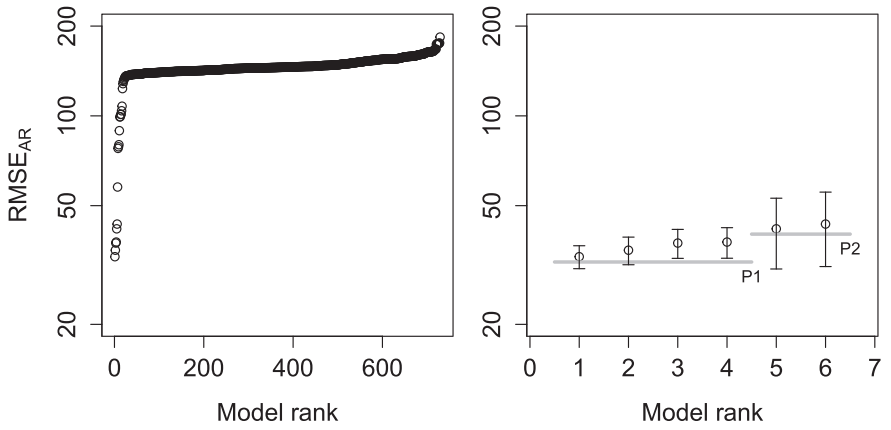

Fig. 7 Error profile for the task of inducing the repressilator model with local kinetic rates. Complete error profile (*left*). The top six models in the first two plateaus with error bars showing the standard deviation across restarts. The gray horizontal lines depict plateaus (*right*)

subpopulations of individuals that are susceptible (S) to the observed disease, latently infected (L), infected with (I) and without symptoms (A, i.e., asymptomatic), quarantined (Q) and recovered (or removed, in case of fatal diseases, R). In the library of modeling knowledge, all these compartments are represented with a single entity template *compartment* which has the variable property of *noi*, representing the number of individuals in the compartment at a given time point.

At the point of introduction of a pathogen in the population, the entire population can be considered to consist of susceptible individuals (in the compartment S), except for the individuals by whom the pathogen is introduced. From this point on, we can observe different processes of flow between compartments. One way to model the infection of individuals is to assume that all infected individuals manifest the disease symptoms. In this case, the A compartment is not populated. An alternative, more complex, model assumes that we can also have infected individuals that do not manifest the symptoms. In both cases, the infection might cause a direct flow from S to I (and/or A) or indirect flow through the L compartment of latently infected individuals.

The recovery of individuals from a disease can either cause flows from the A and I compartments to the population of recovered (or removed) individuals R or cause flows from the A and I to the population of susceptible individuals S. In any case, the recovery of the individuals from I can be controlled by moving the infected individuals to the quarantine compartment Q. Finally, the general model involves a flow of individuals from the recovery compartment to the population of susceptible individuals.

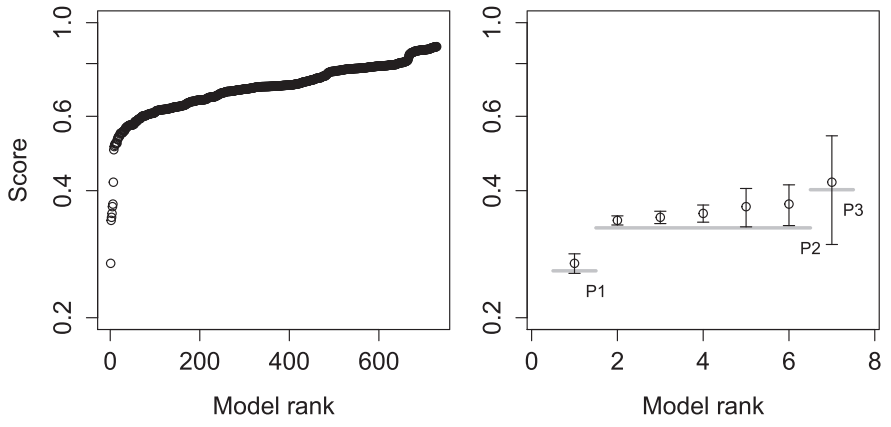

Fig. 8 Error profile for the task of inducing the repressilator model with local kinetic rates based on a model selection score that takes into account model complexity penalization. Complete error profile (*left*). The top seven models in the first three plateaus with error bars showing the standard deviation across restarts. The gray horizontal lines depict plateaus (*right*)

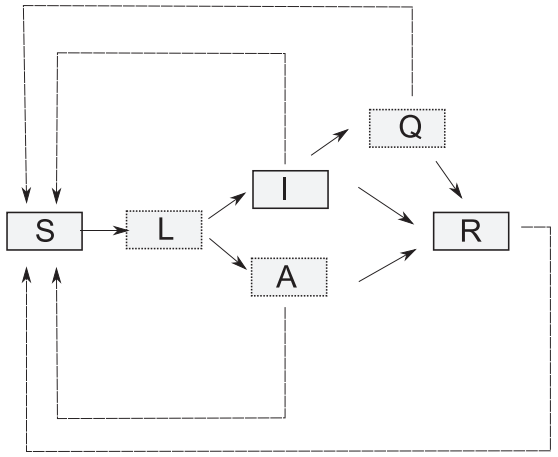

Fig. 9 Graphical representation of a general compartmental model in epidemiology. The boxes correspond to compartments, i.e., subpopulations, and arrows denote the flows of individuals between compartments

The general model can be instantiated to a number of variants, ranging from the simple SIR model that assumes only three compartments of susceptible, infected and recovered individuals, through the SLIR model that introduces the population of latently infected individuals, to the most complex SLIAQRS model that comprises all the compartments depicted in Fig. 9. For example, the SIR model includes two processes. The first instantiates the template process of *infection_symptomatic* that includes a single reaction equation: *S.noi + I.noi* $\rightarrow$ *I.noi + I.noi [i]*, where *i* represents the rate of infection. The other process represents the template of *recovery_symptomatic* that includes the reaction equation *I.noi* $\rightarrow$ *R.noi [r]*, where *r* denotes the recovery rate.

In contrast to the previous synthetic tasks, here we use two data sets of real measurements for induction. These come from two epidemic outbreaks, the outbreak of the Great Plague in Eyam in 1666 [45] and the outbreak of influenza type A subtype H3N2 in Tristan da Cunha in 1967 [46, 47]. The measurements for the case of the outbreak in Eyam are taken bimonthly at seven time points in the period from 3rd of July to 20th of October 1666. They include two variables: number of healthy individuals and the number of individuals that have complained of symptoms. The measurements from Tristan da Cunha are taken daily at 21 time points in October 1967. They also include two variables: number of individuals showing symptoms of infection and the number of recovered individuals.

To match the compartment variables to the variables in the data sets, we calculate the number of healthy (individuals not showing any symptoms of infection) as the sum of the number of individuals in the *S*, *L* and *A* compartments, the number of infected as the sum of the number of individuals in the *I* and *Q* compartments and the number of recovered as the number of individuals in the *R* compartment.

In accordance with the experimental setting for obtaining the measurements, we use the second objective function $RMSE_{SR}$. Since the experimental data comes from real and therefore noisy measurements, we take into account model complexity to obtain the model selection score.

Eyam plague outbreak

For this task, we consider all possible instances of the general model as previously described, by introducing a small set of constraints of mutual exclusivity of symptomatic and asymptomatic infection, thus instantiating only the corresponding recovery for each type of infection. The total number of model structures under these constraints is 24. The initial conditions at the first time point were set to 254 individuals in the *S*, 7 individuals in the *I* and 0 in the other compartments, which exactly matches the initial conditions from the original study by Ragget [45]. The same paper proposes two plausible model structures: SIR, the structure that has been analyzed in the paper, and SLIR, suggested as the most promising one for further study. Thus, our list of plausible models structures *P* is (SIR, SLIR).

The first plateau of the error profile, depicted in Fig. 10, contains a single model that has the SIR structure. Therefore the recall is *50 %*, the hit is *true* and the plateau size is *1*. The model with the SLIR structure is ranked as second and comprises the second error-profile plateau. Thus, when considering the two models in the two leftmost plateaus, ProBMoTs successfully reconstructs the two plausible model structures suggested before [45]. Note that the complexity-based model selection score bears high discriminative power, since each model forms its own plateau. The next four plateaus of the error profile include the SIRS, SLIRS, SIQR and SLIQR models, which render model structures that extend the basic SIR and SLIR with the assumptions of survivors (return to the susceptible compartment) or a quarantine compartment to provide plausible explanations of the observed data.

Tristan da Cunha influenza outbreak

For this task, we consider the same set of 24 model structures that instantiate the general model from Fig. 9. Based on the data available, we set the initial number of infected individuals to 1, other initial values to 0, except for the initial number of susceptible individuals that was fitted as a model parameter. We selected the two best performing model structures from Toni et al. [6] as plausible and set *P* to (SLIR, SIR). The other two model structures considered in the study are a modified SLIR structure, that includes time-delayed flow models, and a SIRS structure.

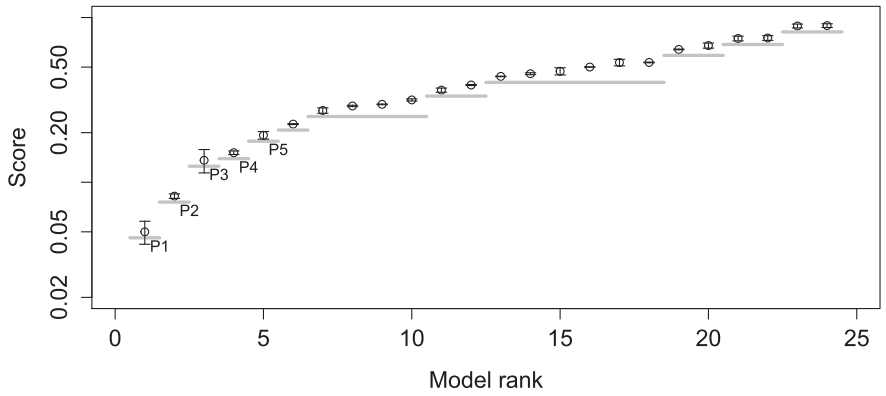

Fig. 10 Error profile for the Eyam plague modeling task. The error bars show the standard deviation of the model selection score across the runs. The gray horizontal lines depict plateaus

The first plateau of the error profile, depicted in Fig. 11, contains the SLIR model that is the first model in *P*, leading to a recall of *50 %*, the hit indicator is *true* and the plateau size is *1*. The second ranked model in the second plateau has the SIR structure of the second model in *P*. As in the case of the Eyam plague experiments, ProB-MoTs perfectly reconstructed the results of the previous modeling experiments reported by Toni et al. [6].

Discussion

The formalism for stochastic process-based modeling, that we introduce in this work, retains the modular and straight-forward specification of entire classes of model structures from its deterministic counterpart. In contrast to the formalisms commonly used in systems biology that employ different levels of abstraction but focus primarily on the efficient description of a single model structure [10–13], the process-based formalism allows for describing uncertainty in both the structure and parameter values of a model by representing classes of model structures. The introduction of reaction-equation based description of processes improves the understandability of process-based models and allows for their stochastic interpretation, improving the generality and utility of the process-based modeling approach and bringing it closer to the domain of biology.

The experimental evaluation shows that our approach can be successfully applied to a range of problems of learning stochastic models. These can come from different biological domains and represent phenomena at different scales. Our approach exhibits excellent performance on the considered tasks, producing accurate and understandable models and successfully reconstructing the results of previous modeling efforts. The proposed approach can be applied to an arbitrary domain of interest by encoding an appropriate library of template entities and processes encountered in the particular domain.

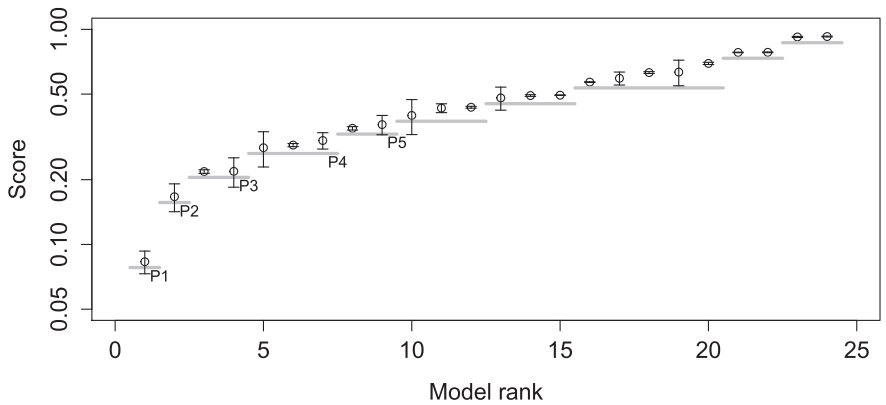

Fig. 11 Error profile for the Tristan da Cunha influenza outbreak modeling task. The error bars show the standard deviation of the model selection score across the runs. The gray horizontal lines depict plateaus

However, several limiting issues may arise during the application of the proposed approach.

First, solving the parameter estimation task for each model structure can lead to both identifiability and distinguishability problems. The identifiability of the model parameters is a problem often encountered when modeling biological systems [48]. Performing identifiability analysis for each candidate structure is in principle possible. However, this can be challenging in terms of computational complexity when considering a large number of candidate model structures. A more tractable problem is the one of distinguishability of the candidate model structures in terms of the applied model selection criteria. Within process-based modeling, this problem presents itself in the form of long plateaus in the error profile. This problem has been studied for the task of learning deterministic models of dynamical systems from data and domain knowledge [32]. The study shows that the problem of distinguishability can be successfully addressed by the introduction of problem specific, domain dependent criteria for parameter optimization and model selection. Although this study is limited to the case of deterministic models, further work can extend its scope to stochastic modeling.

Second, the encoding of very large and complex systems in the proposed formalisms may be cumbersome. The use of reaction equations to encode a system with many entities, that is comprised of a large number of simple and repeating interactions might lead to excessively lengthy descriptions. Following concepts from related work, the issue of combinatorial complexity of composing models from elementary interactions may be solved by rule- or agent-based formalisms by introducing further abstraction.

Finally, the third limitation of our approach is related to computational cost. The simulation of a candidate model is the computationally most expensive step of the process of model induction. Therefore, the computational cost is proportional to the number of evaluations (and the number of simulations per evaluation) needed for each model during the parameter estimation task. Our method requires exhaustive enumeration and optimization of a number of candidate model structures defined by entity/process templates organized in multiple-level hierarchies of alternatives within a library of domain knowledge. Subsequently, a combinatorial explosion is possible if the problem is not well constrained. This is exactly why the process-based modeling approach includes the facility for imposing constraints on the space of possible model structures by allowing for the definition of an incomplete model.

Conclusion

The area of computational biology lacks a unified methodology for modeling dynamical systems that would include a formalism for representing complex dynamics in a manner easily understandable to biologists and modeling experts. In this paper, we advocate the use of process-based modeling for this purpose. It allows for understandable description of a space of candidate model structures for a given modeling task. It allows for both deterministic and stochastic interpretation of process-based models. Also, it allows for automated induction of models from data and knowledge.

In order to bridge the gap between the existing and commonly used tools for modeling the dynamics of biological systems and the machine learning approaches to computational scientific discovery, we have extended the scope of process-based modeling approaches, specifically ProBMoT, to include stochastic models. As an intermediate representation, our ProBMoTs formalism includes the finer, more intuitive and easier to comprehend representation of reaction equations, which should increase the ease of use of process-based modeling in biology. This finer-grained representation of processes is a feature that broadens the possibilities of interpretation, mainly in the direction of capturing the inherent stochasticity of dynamical systems in biology.

Through the tasks considered in this work, we have shown that our approach can deal with complex parameter and structure search spaces, in lightly constrained settings, with synthetically generated tasks and in less constrained real world problems. We have thus demonstrated the potential of our approach for automated discovery of novel scientific knowledge in domains that require stochastic modeling of dynamical systems. Our results also point at an array of possibilities for further evaluation and improvement.

The presented extension of the process-based formalism integrates reaction equations as a proxy that allows for multiple interpretations of the process-based models. However, we can continue this initial step by integrating other higher-level formalisms. Combining rule-based modeling languages with the process templates from the process-based modeling formalism can be considered as a first direction for further work. The introduction of rule-based constraints would allow for automated modeling of more complex systems.

Another direction for further work stems naturally from the formulation of the modeling task as a combinatorial search problem. It concerns the implementation of incomplete, heuristics-based search strategies over the space of candidate models. Although a comparative evaluation with the method using exhaustive search is needed to establish its utility, this extension will scale-up our approach towards applications to large-scale modeling problems.

Other factors might also contribute to the overall success of our approach, e.g., the choice of a parameter

estimation method and a method for simulation. Existing literature offers comparisons of the performance of different parameter estimation methods on single model structures for modeling tasks from the domain of systems biology (both deterministic and stochastic) [49, 50]. A comparison of the performance of different parameter estimation methods has also been performed in the context of deterministic process-based modeling of aquatic ecosystems [40]. The conclusions from these studies are a good starting point to investigate their performance in the context of stochastic process-based modeling tasks.

Ethics approval

No aspect of this study required ethics approval.

Availability of data and materials

The libraries of domain knowledge, incomplete models and data supporting the conclusions of this article are available in the Zenodo repository https://zenodo.org/record/45503 (doi:10.5281/zenodo.45503).

Endnotes

[1]The notions of entities and processes are ontologically well-grounded and are present (as continuants and occurrents) in the Basic Formal Ontology.

[2]ProBMoT and ProBMoTs are released under the terms of the BSD license http://probmot.ijs.si/licence.html.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

LT and SD initiated the work. JT implemented the new formalism and integrated it within ProBMoT. JT encoded the domain knowledge in the new process-based formalism. JT designed and performed the experiments. JT and LT analyzed the results. JT and LT drafted the manuscript. SD gave critical advice on how to revise the manuscript. All authors read and approved the final manuscript.

Acknowledgments

We acknowledge the financial support of the Slovene Human Resources Development and Scholarship Fund, the Slovenian Research Agency (Grants P2-0103 and P5-0093 (B)), and the European Commission (Grants ICT-2013-612944 MAESTRA and ICT-2013-604102 HBP).

Author details

[1]Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia. [2]Jožef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia. [3]University of Ljubljana, Gosarjeva ulica 5, 1000 Ljubljana, Slovenia.

Received: 1 December 2015 Accepted: 6 March 2016
Published online: 22 March 2016

References

1. Wilkinson DJ. Stochastic Modelling for Systems Biology. Boca Raton: CRC Press; 2006.
2. Lecca P, Laurenzi I, Jordan F. Deterministic Versus Stochastic Modelling in Biochemistry and Systems Biology. Cambridge: Woodhead Publishing; 2013.
3. McAdams HH, Arkin A. Stochastic mechanisms in gene expression. Proc Natl Acad Sci. 1997;94(3):814–9.
4. Arkin A, Ross J, McAdams HH. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ - infected escherichia coli cells. Genetics. 1998;149(4):1633–48.
5. Samoilov M, Plyasunov S, Arkin AP. Stochastic amplification and signaling in enzymatic futile cycles through noise-induced bistability with oscillations. Proc Natl Acad Sci. 2005;102(7):2310–315.
6. Toni T, Welch D, Strelkowa N, Ipsen A, Stumpf MPH. Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. J R Soc Interface. 2009;6:187–202.
7. Džeroski S, Todorovski L. Equation discovery for systems biology: finding the structure and dynamics of biological networks from time course data. Curr Opin Biotechnol. 2008;19(4):360–8.
8. Regev A, Shapiro E. Cellular abstractions: cells as computation. Nature. 2002;419(6905):343.
9. Gillespie DT. Stochastic simulation of chemical kinetics. Annu Rev Phys Chem. 2007;58(1):35–55.
10. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, the rest of the SBML Forum, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr JH, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Le Novère N, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED, Nakayama Y, Nelson MR, Nielsen PF, Sakurada T, Schaff JC, Shapiro BE, Shimizu TS, Spence HD, Stelling J, Takahashi K, Tomita M, Wagner J, Wang J. The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. Bioinformatics. 2003;19(4):524–31.
11. Faeder JR, Blinov ML, Hlavacek WS. Rule-based modeling of biochemical systems with BioNetGen. In: Systems Biology. Methods in Molecular Biology. New York: Humana Press; 2009. p. 113–67.
12. Danos V, Feret J, Fontana W, Harmer R, Krivine J. Rule-Based Modelling, Symmetries, Refinements In: Fisher J, editor. Formal Methods in Systems Biology. Lecture Notes in Computer Science. Berlin Heidelberg: Springer; 2008. p. 103–22.
13. Priami C, Quaglia P. Modelling the dynamics of biosystems. Brief Bioinform. 2004;5(3):259–69.
14. Clark A, Gilmore S, Hillston J, Tribastone M. Stochastic process algebras. In: Formal Methods for Performance Evaluation. Lecture Notes in Computer Science. Berlin Heidelberg: Springer; 2007. p. 132–79.
15. Blossey R, Cardelli L, Phillips A. A compositional approach to the stochastic dynamics of gene networks. Trans Comput Syst Biol. 2006;3939(3939):99–122.
16. Ciocchetta F, Hillston J. Bio-pepa: A framework for the modelling and analysis of biological systems. Theor Comput Sci. 2009;410(33–34): 3065–084.
17. Priami C, Quaglia P. Beta binders for biological interactions. In: Computational Methods in Systems Biology. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer; 2005. p. 20–33.
18. Bortolussi L, Policriti A. Modeling biological systems in stochastic concurrent constraint programming. Constraints. 2008;13(1-2):66–90.
19. Hoops S, Sahle S, Gauges R, Lee C, Pahle J, Simus N, Singhal M, Xu L, Mendes P, Kummer U. Copasi - a complex pathway simulator. Bioinformatics. 2006;22(24):3067–074.
20. Mathworks SimBiology toolbox. http://www.mathworks.com/products/simbiology/. Accessed 01 Feb 2016.
21. Bansal M, Belcastro V, Ambesi-Impiombato A, di Bernardo D. How to infer gene networks from expression profiles. Mol Syst Biol. 2006;3:78–8.
22. Wahl SA, Haunschild MD, Oldiges M, Wiechert W. Unravelling the regulatory structure of biochemical networks using stimulus response experiments and large-scale model selection. IEE Proc Syst Biol. 2006;153: 275–8510.
23. Oates CJ, Dondelinger F, Bayani N, Korkola J, Gray JW, Mukherjee S. Causal network inference using biochemical kinetics. Bioinformatics. 2014;30(17):468–74.
24. Henriques D, Rocha M, Saez-Rodriguez J, Banga JR. Reverse engineering of logic-based differential equation models using a mixed-integer dynamic optimization approach. Bioinformatics. 2015;31(18):2999–3007.
25. Bonneau R, Reiss D, Shannon P, Facciotti M, Hood L, Baliga N, Thorsson V. The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. Genome Biol. 2006;7(5):36.

26. Langley P, Simon HA, Bradshaw GL, Zytkow JM. Scientific Discovery: Computational Explorations of the Creative Processes. Cambridge: MIT Press; 1987.
27. Džeroski S, Todorovski L. Encoding and using domain knowledge on population dynamics for equation discovery In: Magnani L, Nersessian NJ, editors. Logical and Computational Aspects of Model-Based Reasoning. Netherlands: Springer; 2002. p. 227–47.
28. Bridewell W, Langley P, Todorovski L, Džeroski S. Inductive process modelling. Mach Learn. 2008;71:109–30.
29. 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. Pittsburgh: AAAI Press; 2005. p. 892–7.
30. Čerepnalkoski D. Process-based models of dynamical systems: representation and induction. PhD thesis, Jožef Stefan International Postgraduate School, Ljubljana, Slovenia. 2013.
31. Džeroski S, Todorovski L. Modeling the dynamics of biological networks from time course data In: Choi S, editor. Systems Biology for Signaling Networks. Berlin Heidelberg: Springer; 2010. p. 275–94.
32. Tanevski J, Todorovski L, Kalaidzidis Y, Dzeroski S. Domain-specific model selection for structural identification of the rab5-rab7 dynamics in endocytosis. BMC Syst Biol. 2015;9(1):31.
33. Elowitz MB, Leibler S. A synthetic oscillatory network of transcriptional regulators. Nature. 2000;403:335–8.
34. Sanft KR, Gillespie DT, Petzold LR. Legitimacy of the stochastic michaelis-menten approximation. IET Syst Biol. 2011;5(1):58–69.
35. Gillespie DT. A rigorous derivation of the chemical master equation. Physica A Stat Mech Appl. 1992;188(1–3):404–25.
36. Gillespie DT. The chemical langevin equation. J Chem Phys. 2000;113(1): 297–306.
37. Rice JA. Mathematical Statistics and Data Analysis. Boston: Cengage Learning; 2006.
38. Todorovski L, Džeroski S. Integrating domain knowledge in equation discovery In: Džeroski S, Todorovski L, editors. Computational Discovery of Scientific Knowledge. Berlin Heidelberg: Springer; 2007. p. 69–97.
39. Bridewell W, Langley P. Two kinds of knowledge in scientific discovery. Top Cogn Sci. 2010;2(1):36–52.
40. Čerepnalkoski D, Taškova K, Todorovski L, Atanasova N, Džeroski S. The influence of parameter fitting methods on model structure selection in automated modeling of aquatic ecosystems. Ecol Model. 2012;245: 136–65.
41. Ramsey S, Orrell D, Bolouri H. Dizzy: Stochastic simulation of large-scale genetic regulatory networks. J Bioinforma Comput Biol. 2005;03(02): 415–36.
42. Price K, Storn RM, Lampinen JA. Differential Evolution: A Practical Approach to Global Optimization. Berlin Heidelberg: Springer; 2005.
43. Tomshine J, Kaznessis YN. Optimization of a stochastically simulated gene network model via simulated annealing. Biophys J. 2006;91(9):3196–205.
44. Brauer F, van den Driessche P, Wu J, Allen LJS. Mathematical Epidemiology. Berlin Heidelberg: Springer; 2008.
45. Ragget GF. Modelling the eyam plague. IMA J. 1982;18:221–6.
46. Hammond BJ, Tyrrell DAJ. A mathematical model of common-cold epidemics on tristan da cunha. J Hyg. 1971;69:423–33.
47. Shibli M, Gooch S, Lewis HE, Tyrrell DAJ. Common colds on tristan da cunha. J Hyg. 1971;69:255–62.
48. Gutenkunst RN, Waterfall JJ, Casey FP, Brown KS, Myers CR, Sethna JP. Universally sloppy parameter sensitivities in systems biology models. PLoS Comput Biol. 2007;3(10):1–8.
49. Moles CG, Mendes P, Banga JR. Parameter estimation in biochemical pathways: a comparison of global optimization methods. Genome Res. 2003;13:2467–474.
50. Sun J, Garibaldi JM, Hodgman C. Parameter estimation using metaheuristics in systems biology: A comprehensive review. IEEE/ACM Trans Compu Biol Bioinforma. 2012;9(1):185–202.

Chapter 4

Domain-Specific Selection Criteria for Process-Based Modeling

4.1 Problem Description

When considering multiple competing structural hypotheses for the task of inferring models of dynamical systems, selecting the best among them is a problem that must be addressed. The choice of the criteria and properties according to which a model can be considered "best" should be ultimately left to expert judgment. From a statistical modeling, or machine learning perspective, the model selection problem is considered to be the problem of selecting the model that has the lowest generalization error (Hastie et al., 2009). In a data-rich scenario, the generalization error of a model is calculated by using independent observations of the system that have not been used while training the model, i.e., by using a test set.

The assumption that the error of the model measured on the observations used to infer the model (training error) is a good estimate of generalization error is overly optimistic. This is due to the adaptation of the model to the available observations and may result in the selection of an overfitted model. A better estimate can be obtained by partitioning the observations into a training set and a validation set. The models can then be trained (optimized) using the training observations and be tested against the validation set to obtain an estimate of the generalization error. Many methods in machine learning focus on model selection based on variations of this approach.

However, the abundance of observational data for inferring a model of a dynamical biological system is also an overly optimistic assumption. Additionally, the observations used for modeling dynamical biological systems are in the form of time-series, which cannot be always randomly and correctly sampled to obtain independent datasets. An exception may be, for example, the case of collecting observations from a single dynamical system under the influence of different stimuli. Nevertheless, the sample size should be significantly large. Therefore, we turn to methods for estimating the generalization error in data-poor scenarios without partitioning the set of available observations.

The expected generalization error may come from three sources: error due to model bias, variance and irreducible error. The irreducible error cannot be estimated due to the existence of noise within the modeled system or the impossibility to be adequately captured observations. It represents a lower bound of the error. The estimation of the generalization error is therefore performed by estimation of model bias and variance, using the information captured by the structure of the model and the procedure used for its inference. The subsequent model selection is performed based on the most desirable trade-off between bias and variance.

The model bias comes from the model representation (representation bias) and from the function that guides the model inference and evaluates the candidate models (evaluation bias). The source of variance is the sensitivity of the inferred model to minor fluctuations in the set of observations used for its inference.

A diverse family of functions considered for modeling and high uncertainty are related to lower/weaker model bias, while a limited family of functions and low uncertainty in their structure/parameters are related to higher/stronger bias. Generally, the more complex the model, the lower the bias and the higher the variance error due to the strong adaptation of the model to the optimization objectives (overfitting). Less complex models have higher bias but lower variance error due to their inflexibility (underfitting). The choice of representational bias requires knowledge of the domain and the appropriate space of model hypotheses, and is orthogonal to the strength of the bias. A representational bias that defines a space that contains the exact family of functions that corresponds to the observed phenomena will lead to the lowest bias and variance error.

Another source of bias is the process of model inference and in particular the function that guides search and optimization. This bias comes from the bias property of a statistical estimator, specifying how well the likelihood function for a model is approximated, if they can be directly specified and used in the process of inference. Alternatively, the evaluation bias may come from the optimality and completeness of a specific heuristic function used for search or optimization that compensates for the lack of complete observations or observations with lower-grade quality and directly relates the modelers expectations to the realization of the model.

In both machine learning and biology, the most commonly adopted principle for selecting a model which offers the best bias-variance trade-off is the parsimony principle. This principle states that the best model is the one that has the simplest representation while adequately describing the observations. In general, the parsimony principle comes in the form of a complexity based regularization term that is added to a goodness-of-fit function $\mathcal{E}$ that estimates the fit of the model realization to the observations:

$$E_{reg} = \mathcal{E}(x, \hat{x}|\mathcal{M}, \theta) + \lambda \cdot \mathcal{R}(\theta), \tag{4.1}$$

where $\hat{x}$ is the realization obtained from the model $\mathcal{M}$ with parameter values θ, x is the observation of the system, and λ is a hyper-parameter that can be set by an expert or optimized at a meta-level.

Frequently, the parameters of a model are indicative of its complexity. Therefore, the regularization term is a function of the parameters of the model. Commonly used regularization terms are, for example, based on a norm of the parameters (such as the Tikhonov regularization where $\mathcal{R}(\theta) = ||\theta||_2^2$ and the Lasso regularization where $\mathcal{R}(\theta) = ||\theta||_1$) or the number of parameters (such as the AIC, where $\mathcal{R}(\theta) = |\theta|$ and BIC, where $\mathcal{R}(\theta) = \ln(N) \cdot |\theta|$, N is the number of data points, and the errors are regularized based on the logarithm of the likelihood function).

For model selection in systems and synthetic biology, regularization based approaches are commonly considered for model selection among manually enumerated candidate model structures (Cedersund & Roll, 2009; Kirk et al., 2013). Regularization has been shown to improve the performance of parameter estimation methods for modeling tasks in biology (Gábor & Banga, 2015) and may be sufficient for inferring models of dynamical systems by optimization based design methods.

We are concerned with model selection for process-based modeling, where parameter estimation is repeatedly performed in the context of selecting the model structure. The formalism of process-based modeling allows for capturing the most relevant domain knowledge needed for the composition of model structures and for a flexible representation of the space

of candidate models with various amounts of uncertainty in model structure/parameters. At the level of model structure, a process-based model is a linear composition of process instances, corresponding at a lower level to a set of instance equations. The regularized error E_{reg} for learning process-based models can be defined by the following equation:

$$E_{reg} = \mathcal{E}(x, \hat{x}|\mathcal{M}_{\mathcal{I}}, \theta) + \lambda \cdot \mathcal{R}(\theta'_{\mathcal{M}_{\mathcal{I}}}) \tag{4.2}$$

Here, as before, θ are the parameter values of the model $\mathcal{M}$, but θ' is the ordered set of structural parameters $(\forall\{t|t \in \theta'_{\mathcal{M}_{\mathcal{I}}}\}(t \in \{0,1\}))$ that indicate the presence of process or equation instances in the model, implicitly defined for a candidate model structure $\mathcal{M}_{\mathcal{I}}$ (with optimized parameter values) refined from an incomplete model $\mathcal{I}$, during the process of enumeration.

The parsimony principle may be underestimating the bias introduced by the minimization of the discrepancy between observed system behavior and simulated model behavior, when the observations are scarce or do not capture enough details of the true behavior of the system. In such a context, models of any complexity can overfit. In this type of scenario the parsimony principle may not offer an informative insight and depending on the choice of the hyper-parameter, model selection may result in choosing the least complex model which may not be correct or resort to choosing randomly from a large set of models that are indistinguishable according to their estimated generalization error.

We further conjecture that the strengthening of the evaluation bias by additional domain and knowledge based heuristics for process-based modeling can offer an alternative when regularization based on the complexity of the model does not alleviate the model selection problem. The evaluation bias strengthening for PBM is performed by encoding of additional domain knowledge in the form of domain-specific criteria. The domain specific criteria are included in the regularized error score from Equation 4.2 as follows:

$$E_{reg} = \mathcal{E}(x, \hat{x}|\mathcal{M}_{\mathcal{I}}, \theta) + \xi \cdot \mathcal{D}(x, \hat{x}|\mathcal{M}_{\mathcal{I}}, \theta) + \lambda \cdot \mathcal{R}(\theta'_{\mathcal{M}_{\mathcal{I}}}) \tag{4.3}$$

The function $\mathcal{D}$ is lower bounded by 0 and is subject to minimization. It relates knowledge about the desired behavior of the system to the properties of the model simulation and the observations. It serves as an additional heuristic that guides the parameter estimation of each model structure and ultimately the selection of the best model.

The parsimony principle is clearly insufficient to solve the model selection problem in the task of modeling the dynamics of the Rab5-Rab7 switch in endocytosis. As is the case in other tasks of modeling highly non-linear systems using real world data, due to the combination of limited observability and the high complexity of the considered structural hypotheses, the problem of model selection becomes particularly hard. Regularization based approaches such as the Bayesian information criterion fail to alleviate the problem (Tanevski et al., 2013).

Our work (Tanevski et al., 2015) extends the work on this modeling problem by Del Conte-Zerial et al. (2008) and the follow up work on parameter estimation focusing on a specific model by Tashkova, Korošec, Šilc, Todorovski, and Džeroski (2011). As an efficient alternative to the manual enumeration of candidate model structures considered by Del Conte-Zerial et al. (2008), we develop a library of domain knowledge and define a space of candidate model structures in the form of an incomplete process-based model. We further demonstrate that the bias strengthening approach using domain specific criteria outperforms the standard model selection approach based on the parsimony principle.

This work was published in a journal article which constitutes the remainder of this chapter. The full bibliographic reference to the article is:

Tanevski, J., Todorovski, L., Kalaidzidis, Y., Džeroski, S. (2015). Domain-specific model selection for structural identification of the Rab5-Rab7 dynamics in endocytosis. *BMC Systems Biology, 9*(1), 1-31.

4.2 Related Publication

Tanevski *et al. BMC Systems Biology*
DOI 10.1186/s12918-015-0175-x

BMC Systems Biology

RESEARCH ARTICLE Open Access

Domain-specific model selection for structural identification of the Rab5-Rab7 dynamics in endocytosis

Jovan Tanevski[1,2*], Ljupčo Todorovski[3], Yannis Kalaidzidis[4] and Sašo Džeroski[1]

Abstract

Background: Given its recent rapid development and the central role that modeling plays in the discipline, systems biology clearly needs methods for automated modeling of dynamical systems. Process-based modeling focuses on explanatory models of dynamical systems; it constructs such models from measured time-course data and formalized modeling knowledge. In this paper, we apply process-based modeling to the practically relevant task of modeling the Rab5-Rab7 conversion switch in endocytosis. The task is difficult due to the limited observability of the system variables and the noisy measurements, which pose serious challenges to the process of model selection. To address these issues, we propose a domain-specific model selection criteria that take into account knowledge about the necessary properties of the simulated model behavior.

Results: In a series of modeling experiments, we compare the results of process-based modeling obtained with different model selection criteria. The first is the standard maximum likelihood criterion based solely on least-squares model error. The second one is a parsimony-based criterion that also takes into account model complexity. We also introduce three domain-specific criteria based on domain expert expectations about the simulated behavior of an endocytosis model. According to the first criterion, 90 of the candidate models are indistinguishable. Furthermore, taking into account the complexity of the model does not lead to better model selection. However, the use of domain-specific criteria results in a remarkable improvement over the other two model selection criteria.

Conclusions: We demonstrate the applicability of process-based modeling to the task of modeling the Rab5-Rab7 dynamics in endocytosis. Our experiments show that the domain-specific criteria outperform the standard domain-independent criteria for model selection. We also find that some of the model structures discarded as implausible in previous studies lead to the expected Rab5-Rab7 switch behavior.

Keywords: Process-based modeling, Dynamical systems, Structural identification, Model selection, Endocytosis

Background

The area of computational systems biology aims at providing computational methods and tools that help in the processes of modeling biological systems, simulating the resulting models, and analyzing their behavior. The modeling process begins with formulating structural hypotheses, i.e., the knowledge-driven identification of the constituent system entities and the interactions between them. There are many modeling formalisms for systems biology that have been developed for the purpose of transformation of structural hypotheses into interpretable and executable models [1, 2]. Since systems biology focuses on dynamical behavior at the molecular level, where change of properties of molecular constituents is observed through time, ordinary differential equations are most commonly used to formulate mathematical models.

In order to refine a conjectured model structure into a complete model, one has to estimate the values of the model parameters. The parameter estimation task is often formulated as a nonlinear optimization problem [3, 4], where the aim is to minimize the discrepancy between

*Correspondence: jovan.tanevski@ijs.si
[1]Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
[2]Jožef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia
Full list of author information is available at the end of the article

the model simulation and the measured behavior of the observed system. Many of the commonly used systems biology tools, such as COPASI [5], CellDesigner [6] and others, focus on the parameter estimation task, considering a single model structure provided by the human modeler.

Recently, computational methods for automated modeling that address both structure identification and parameter estimation, have emerged. On the one hand, probabilistic methods [7] are intrinsically slow and inefficient when applied to large classes of complex model structures. On the other hand machine learning methods for equation discovery [8] are applicable in complex modeling scenarios [9]. A notable recent development is process-based modeling that allows for the integration of knowledge and measured data into the process of inducing mathematical models of dynamical systems [10–12]. These approaches have already been successfully applied in systems biology [13, 14] to the tasks of modeling the structure and dynamics of biological networks from time-course measurement data.

In this paper, we apply process-based modeling to endocytosis, an indispensable part of the cell immune response. Endocytosis is the target of many modeling efforts in systems biology. Del Conte-Zerial et al. [15] present such an effort focusing on the early phase of endocytosis, i.e., the conversion of Rab5 domain proteins to Rab7 domain proteins. They consider a number of alternative model structures, perform careful and extensive comparative analysis thereof and propose a particular cut-out switch structure as the most appropriate model of the Rab5-Rab7 conversion. In a follow-up paper, Tashkova et al. [16] address the task of estimating the parameters in this single cut-out switch model structure.

Process-based modeling

Process-based modeling is concerned with inducing explanatory models of dynamical systems from data (measurements of the behavior of the observed system) and knowledge (about modeling systems from the given domain). A process-based model describes a dynamical system at two levels of abstraction. At the higher abstraction level, the model is cast as a set of entities (that correspond to system variables) and processes (i.e. interactions between the entities). At the lower abstraction level, each process includes a set of differential and/or algebraic equations, which models the corresponding interaction between the entities involved in the process. While the higher level bears the explanatory power of a process-based model, revealing the structure of its interactions, the lower-level allows for automatic transformation of the model into a set of differential equations that can be used to simulate the dynamical behavior of the observed system.

ProBMoT [17] is a recent implementation of the process-based paradigm for automated modeling of dynamical systems from knowledge and data. It is implemented in Java.

It is still under active development, with the most recent version available for download at http://probmot.ijs.si.

A graphical description of the process of automated modeling using ProBMoT is presented in Fig. 1. ProBMoT takes as input time-series data, i.e., **measurements** of the dynamical behavior of the observed system. It also takes as input modeling knowledge about the studied domain, represented as a **library** of template model components, i.e., entities and processes. Finally, it takes as input a set of constraints, i.e., an **incomplete model**, that correspond to the particular modeling assumptions made for the specific task at hand.

The library of domain knowledge is a collection of template entities and processes that represent generic components for building models of dynamical systems in the domain of interest. For a particular modeling task, the user specifies an incomplete model that includes a set of entities in the observed system and constraints on the possible interactions between them. The specific entities in the modeling task are instances of the generic template entities in the library. Using them, ProBMoT can enumerate all possible instances of process templates in the library. Following the constraints from the incomplete model, ProBMoT combines these process instances into candidate model structures.

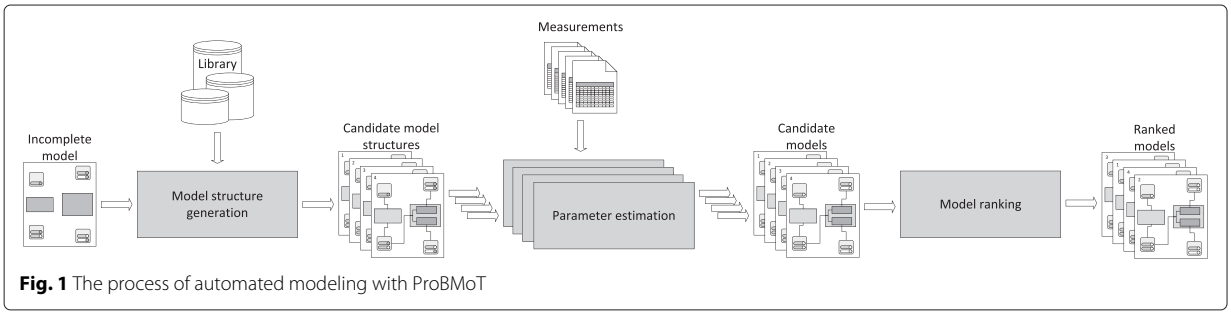

Fig. 1 The process of automated modeling with ProBMoT

For each candidate model structure, parameter estimation is performed to obtain a set of point estimates of the unknown model parameters that most adequately explains the observed system behavior. To achieve this, parameter estimation in ProBMoT minimizes an objective function that measures the difference between the observed and the simulated behavior. To this end, ProBMoT employs various meta-heuristic optimization methods from the jMetal framework [18] and the SUNDIALS suite for simulating ordinary differential equations [19]. The output of the parameter estimation task represents a candidate model. After the parameter values for all candidate model structures have been estimated, the resulting candidate models are ranked by the minimized value of the objective function. Finally, the ranked list of candidate models represents the output of ProBMoT.

Model selection

Given the output of ProBMoT, i.e. a ranked list of candidate models, we face the model selection problem of selecting the most appropriate candidate model. By default, the top-ranked model is selected that corresponds to the maximum likelihood criterion for model selection; it only takes into account the least-squares fit to the observed data. However, for the task at hand, the limited observability of protein concentrations and the noise in the measurements pose serious challenges to this default model selection method [16, 20], and the selected model often overfits the observed data.

To address this problem, various model selection criteria have been considered in systems biology [7]. Many of them follow the parsimony principle by combining the least-squares model error with the complexity of the model structure. In addition to this general criterion for model selection, we consider here domain-specific criteria that take into account the expected and necessary properties of model simulations in the particular context of endocytosis. We conjecture that these task-specific criteria for model selection will outperform the other two, general criteria.

Note that the model selection problem is especially important in the context of automated modeling, where large classes of candidate models are being considered. Few computational tools that address the structure identification task (e.g., ABC-SysBio) recast model selection into a parameter estimation task [21]. However, this reformulation requires the user to specify a list of candidate models, a demanding and tedious task for a human modeler. In contrast, process-based modeling offers a more flexible formalism for specifying complex spaces of candidate model structures. Additionally, ProBMoT can consider arbitrary objective functions that correspond to various model selection criteria.

Methods

First, we are going to cast the task of modeling the Rab5-Rab7 conversion in endocytosis as a process-based modeling task. Then, we are going to formally define the three model section criteria used in the study. Finally, we are going to introduce the experimental setup used for the empirical evaluation and the performance comparison of the models obtained using different model selection criteria.

Process-based modeling of endocytosis

Process-based modeling formalizes domain-specific knowledge describing entities, that correspond to the variables of the dynamic systems in the domain at hand, and processes, that correspond to interactions between entities. In the particular context of modeling endocytosis, entities correspond to protein domains and processes refer to biochemical interactions between them. The structure of the library is based on a modular formulation of the system of differential equations for modeling the conversion between the Rab5 and Rab7 protein domains [15] of the form:

$$\begin{aligned}
\frac{dr_5}{dt} &= K_1 - (k_1 + GEF_5(R_5, R_7)) \cdot r_5 + GAP_5(R_5, R_7) \cdot R_5 \\
\frac{dR_5}{dt} &= GEF_5(R_5, R_7) \cdot r_5 - GAP_5(R_5, R_7) \cdot R_5 \\
\frac{dr_7}{dt} &= K_2 - (k_2 + GEF_7(R_5, R_7)) \cdot r_7 + GAP_7(R_5, R_7) \cdot R_7 \\
\frac{dR_7}{dt} &= GEF_7(R_5, R_7) \cdot r_7 - GAP_7(R_5, R_7) \cdot R_7
\end{aligned} \tag{1}$$

where the variables r_5 and r_7 represent the concentrations of GDP-bound (passive state) Rab5 and Rab7 domain proteins, while R_5 and R_7 represent the concentrations of GTP-bound (active state) proteins. Furthermore, the parameters K_i and k_i represent GDP Dissociation Inhibitor (GDI) association rates and GDI dissociation fluxes respectively. The Rab5-Rab7 interactions labeled with GEF represent activating reactions which catalyze the GDP/GTP exchange by guanine nucleotide exchange factors, while the GAP interactions represent reactions which catalyse the GTP hydrolysis by means of GTPase-activating proteins. The rates of both (GEF and GAP) interactions depend on (are functions of) the GTP-bound state concentrations of Rab5 and Rab7.

Figure 2 provides a graphical representation of the model structure [15], where the dashed lines represent optional interactions between the Rab5 and Rab7 protein domains, while the solid lines represent non-optional (mandatory) interactions. The pointed arrows represent the catalisation (activation) of the corresponding exchange or hydrolysis, while the inhibition of the

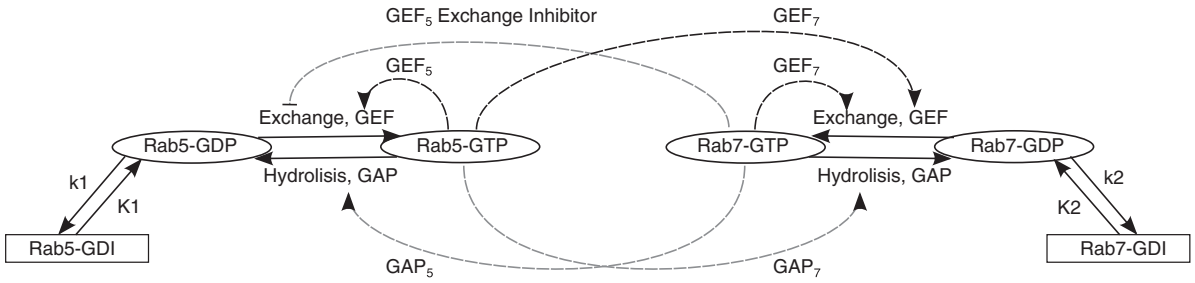

Fig. 2 A graphical representation of the Rab5-Rab7 interaction model structure as considered by del Conte-Zerial et al. [15]

GDP/GTP exchange by the GEF_5Exchange Inhibitor is represented by a truncated line. del Conte-Zerial et al. [15] consider a different set of functional forms for modeling each of the four (GEF and GAP) interactions; the combinations of the different functional forms they consider result in only 54 different model structures from the possible 126.

Based on the model structure, we start the development of the process-based library for modeling endocytosis by encoding a single template entity, presented in Table 1, that refers to a general protein domain. The first two variables in the template represent the concentrations of the active-state (*GTP_bound_state*) and passive-state (*GDP_bound_state*) proteins. The template includes declarations of two constant parameters that correspond to the dissociation flux and the association rate of the protein molecules in the domain. Note that the template entity from Table 1 represents an arbitrary protein domain. In the particular model of endocytosis from Eq. 1, the template entity instantiates into the two *specific* entities of *Rab5* and *Rab7*. In the process-based formalism, the variable *Rab5.GDP_bound_state_conc*, i.e., the *GDP_bound_state_conc* of the entity *Rab5*, corresponds to the model variable r_5. Similarly, *Rab7.GDI_dissociation_flux* corresponds to the model constant parameter k_2.

Table 1 Part of the developed library of domain knowledge. Definition of the template entity Protein

```
template entity Protein {
  vars:
    GDP_bound_state_conc {range:<0,2>},
    GTP_bound_state_conc {range:<0,2>},
    GEF, GAP, t;
  consts:
    GDI_dissociation_flux {range: <0.001, 4>},
    GDI_association_rate {range: <0.001, 4>};
}
// ...
```

When it comes to the process templates, the ProBMoT library, depicted in Fig. 3, closely follows the general structure of the endocytosis model from Fig. 2. Each node in Fig. 3 corresponds to a template process, where the top node label denotes the template name, while the following lines within the node correspond to subprocesses. The template process *Root* specifies the way that the models of individual subprocesses are being combined into the system of differential equations presented in Eq. 1.

The hierarchy of process templates specifies the mutually exclusive alternatives for modeling individual subprocesses in terms of the functional forms of the kinetic laws that govern the observed interaction. For example, the process template *GDI_GDP_membrane_interaction* refers to the interactions between the protein domains and GDI; it contains two subprocesses of *Association_with_GDI* and *Disassociation_with_GDI*. The two corresponding process templates specify the specific mass action kinetic laws used in the model. While each of these two process templates specifies a single kinetic law, the two process templates of *GAPProcess* and *GAPProcessPlus* specify **two** and **three** alternatives for modeling the hydrolysis of GAP_7 and GAP_5 respectively. These include an *Intrinsic_Hydrolysis* process which models a simple non-catalyzed hydrolysis from the active to the passive state of the protein domain and a *Michaelis_Menten* process in which the active state of the opposing protein catalyzes the hydrolysis, a process described by a Micaelis-Menten rate. The *GAPProcessPlus* defines an additional alternative in the form of a *Sigmoidal* process which describes the catalysis using a kinetic rate following a sigmoidal function.

Similarly, the *GEFProcess* template describes the **three** alternatives for modeling the GEF_5 interaction. Two of them describe the auto-catalysis of the exchange, while the *Exchange_Inhibition* process describes an alternative of the interaction where the second protein inhibits the exchange. Table 2 presents a snippet from a process-based library for modeling endocytosis that illustrates the

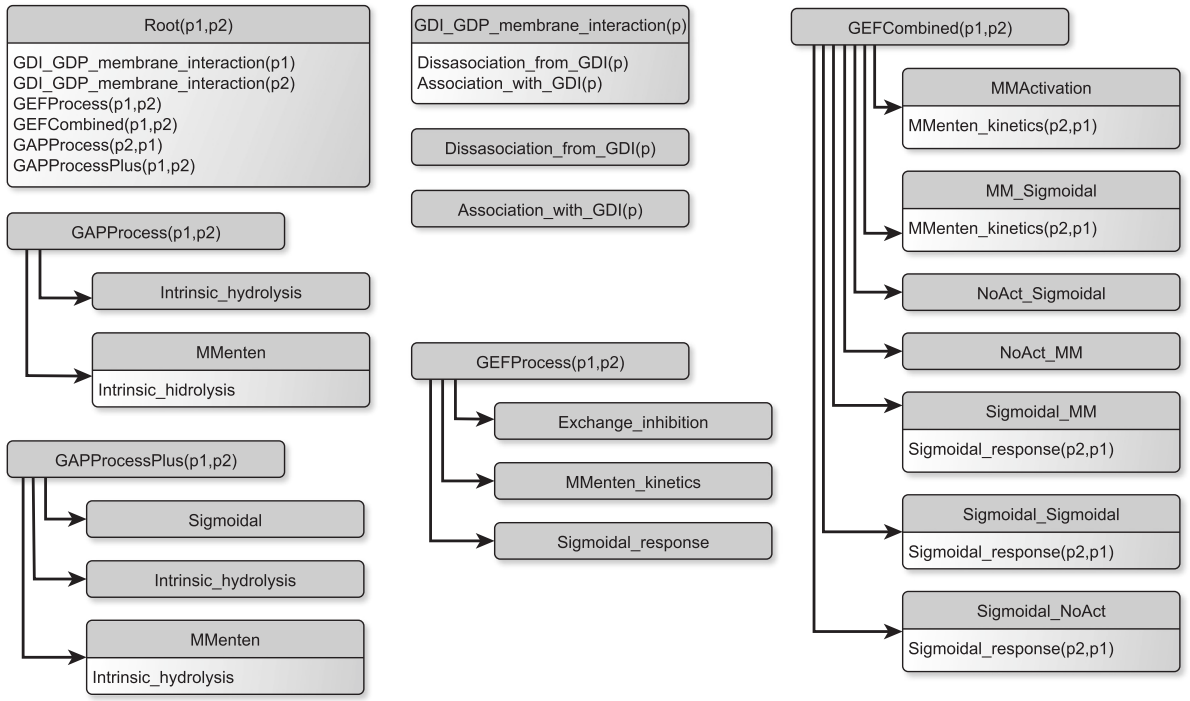

Fig. 3 A schematic representation of the process hierarchy in the library. The label of each node denotes the process name, each line in the content of the node denotes a subprocesses

formalization of individual process templates. It contains specifications of the three mutually exclusive modeling choices for the GEF_5 interaction (i.e., the `GEFProcess` process template).

Finally, the `GEFCombined` template describes the **seven** alternatives for modeling the GEF_7 interaction. Note that this process has two components (represented by two arrows in Fig. 2): auto-activation and activation by the second protein, both of which catalyze the exchange between the active and the passive state of the concerned protein. Therefore, some of the alternatives contain subprocesses which account for the auto-catalysis component and have the same functional forms as the auto-activating alternatives for the `GEFProcess` template.

The whole process-based library for modeling endocytosis, the incomplete model and the task description that have been used to perform all the modeling experiments elaborated later in the empirical part of the paper is

Table 2 Part of the developed library of domain knowledge. Definition of interaction processes with alternative forms

```
template process GEFProcess(p1: Protein, p2: Protein){
  consts: ke{range:<0.001,4>},kf{range:<0.001,4>},kg{range:<0.001,4>},
    km{range:<0.001,4>},ki{range:<0.001,4>};
}
template process MMKinetics : GEFProcess {
  equations:
    p1.GEF = ke*p1.GTP_bound_state_conc/(kg + p1.GTP_bound_state_conc);
}
template process Sigmoidal_response : GEFProcess {
  equations:
    p1.GEF = ke/(1 + exp(kg - p1.GTP_bound_state_conc)*kf);
}
template process Exchange_inhibition : GEFProcess {
  equations:
    p1.GEF = ke*p1.GTP_bound_state_conc/(km*(1+p2.GTP_bound_state_conc/ki)
                                         + p1.GTP_bound_state_conc);
}
// ...
```

available in Additional file 2. Given the library of domain knowledge, ProBMoT enumerates 126 candidate model structures for the particular endocytosis model of interaction between the two protein domains of Rab5 and Rab7. These 126 model structures correspond to the combinations of modeling alternatives specified in the library; the library specifies 2, 3, 3 and 7 alternatives for the four subprocesses of `GAPProcess`, `GAPProcessPlus`, `GEFProcess` and `GEFCombined` respectively, leading to $2 \cdot 3 \cdot 3 \cdot 7 = 126$ combinations. Note that the candidate model structures considered by ProBMoT include the 54 structures analyzed by del Conte-Zerial et al. [15]. In addition, ProBMoT considered 72 model structures that the authors of [15] dismissed in their manual modeling experiment as trivial and/or structurally flawed. In our automated modeling experiment, we decided to minimize the apriori modeling assumptions and consider all 126 model structures as valid alternatives. The distribution of structure components in all 126 models can be seen in Additional file 1: Figure S1.

Note finally, that the ranges specifying possible values of the model parameters in the library closely follow the ones used in previous studies: $[0, 2]$ for the initial values of the system variables ($R5$, $R7$, $r5$ and $r7$), $[5, 195]$ for the parameter `td` in the `root` process template, and $[10^{-3}, 4]$ for all the other model parameters. We used the same parameter estimation setting as the ones found to be most suited to the endocytosis modeling task by Tashkova et al. [16], i.e., the optimization method of Differential Evolution [22] with population size of 81, strategy *rand/1/bin*, differential weight (F) of 0.942 and crossover probability (Cr) of 0.915. The limit on the number of evaluations of the objective function is 20 thousand times the number of parameters, which amounts to about half a million of evaluations per model structure.

Model selection

The **standard approach** to parameter estimation is the one of least-squares, where we look for values of the constant parameters that minimize the sum of squared errors between the simulated model output and the observed system behavior. In other words, we minimize a function based on the sum of squared errors, which in the particular case of modeling endocytosis is calculated as the average relative root mean squared error over the two observed variables

$$E(m) = \frac{1}{2} \cdot \left(\sqrt{\frac{\sum_i (Rab_{5,i} - \widehat{Rab}_{5,i})^2}{\sum_i (Rab_{5,i} - \overline{Rab}_5)^2}} + \sqrt{\frac{\sum_i (Rab_{7,i} - \widehat{Rab}_{7,i})^2}{\sum_i (Rab_{7,i} - \overline{Rab}_7)^2}} \right) \quad (2)$$

where $Rab_{5,i}, Rab_{7,i}$ and $\widehat{Rab}_{5,i}, \widehat{Rab}_{7,i}$ denote the measured and simulated (using the model m) total concentrations of the corresponding Rab domain proteins at the i-th time point, while $\overline{Rab}_5, \overline{Rab}_7$ denote the mean measured values of the corresponding concentrations across all time points. The E measure normalizes the root mean squared error, so that the value of 1 corresponds to the error of a simple baseline model predicting the same mean measured value of the output at each time point.

Note, however, that a sum of squared errors based criterion might not be appropriate for use as a model selection criterion for two main reasons. One is the limited observability of the system variables, which does not provide enough information to discriminate among the different model structures in the space of model structures. The other reason is the risk of over-fitting the noisy data. To address these two issues, we employ three additional model selection criteria.

The following two are **domain-dependent criterion** that take into account the desired behavior of the two system variables that correspond to the concentrations of the active-state Rab domain proteins in the endocytosis model. Namely, when modeling endocytosis, the models of cargo transport through conversion from Rab5 to Rab7 [15, 23] show that the dynamics of the system is controlled by the active, GTP-bound state of the Rab domain proteins, while the concentration of their inactive GDP-bound state remains primarily constant throughout the conversion. Therefore, one would expect that the simulated concentration of the active-state Rab proteins should be highly correlated to the corresponding model output of total (active– and passive-state) protein concentration. Given this expectation about the simulated model behavior, one possible approach is to fit the concentrations of the active-state Rab proteins against the data on total concentrations. This approach was used as an analysis tool for the visual inspection of the model simulation against observed behavior by del Conte-Zerial et al. [15]. However, Tashkova et al. [16] show that this approach fails for parameter estimation, leads to over-fitting of the model to the measured data, and poorly explains the true behavior of the passive state Rab proteins. Here, we first propose an alternative criterion for model selection that discriminates models based on the correlation between the simulated values of the active-state Rab concentrations ($\widehat{R}_5, \widehat{R}_7$) with the observed total concentrations ($Rab5$ and $Rab7$). In particular, we measure

$$R(m) = \frac{1}{2} \cdot (\min(1 - r(\widehat{R}_5, Rab5), 1) + \min(1 - r(\widehat{R}_7, Rab7), 1)), \quad (3)$$

where $r(X, Y)$ denotes the Pearson's correlation coefficient between the time-series X and Y. The R measure takes values in the range $[0, 1]$. The value indicates the

degree of fit to the desired behavior of the hidden system variables, where lower values indicate better correlation between active-state and total protein concentrations.

Based on the same assumption, we introduce a second domain-dependent criterion based on the time when the switch between concentrations of *Rab*5 and *Rab*7 occurs:

$$X(m) = \frac{|t_s - \widehat{t_s}|}{t_{max} - t_0} \tag{4}$$

where t_s and $\widehat{t_s}$ are the switch time points observed in the measured data and the model simulation respectively, while t_0 and t_{max} correspond to the first and the last time point. Since we normalize the distance between the switching point in the simulation and in the measured data by the length of the entire observed time interval, the X measure takes values in the range $[0, 1]$.

We also consider a combination of the domain-dependent criteria $R(m)$ and $X(m)$:

$$RX(m) = \frac{1}{2} \cdot (R(m) + X(m)) \tag{5}$$

To explore the trade-off between the error $E(m)$ and the different domain-dependent criteria, we introduce the combined criteria,

$$ER(m) = \alpha \cdot E(m) + (1 - \alpha) \cdot R(m), \tag{6}$$

$$EX(m) = \alpha \cdot E(m) + (1 - \alpha) \cdot X(m), \tag{7}$$

$$ERX(m) = \alpha \cdot E(m) + (1 - \alpha) \cdot RX(m), \tag{8}$$

where α is a trade-off parameter in the range $[0, 1]$. The value of 0 leads to model selection based purely on the domain-dependent criteria, while the value of 1 leads to model selection based purely on the error $E(m)$.

Finally, we also consider a general, **domain independent criterion** commonly used to avoid overfitting, based on the parsimony principle. Following this principle, from a number of models with comparable error, we select the simplest one. Model selection approaches that follow the parsimony principle, such as the Akaike information criterion or minimal description length [24], deal with finding a trade-off between the model error and the model complexity. In the particular case of process-based models, we measure the complexity of a model as the number of processes in the model structure, i.e., $C(m) =$ #processes(m). In turn, we introduce the β parameter to trade-off between the model error (degree-of-fit) and complexity, as follows:

$$EC(m) = \beta \cdot E(m) + (1 - \beta) \cdot C(m), \tag{9}$$

where the value of the trade-off parameter β is in the range of $[0, 1]$.

A more complicated, domain-dependent, version of this criterion can be derived from equations 6-8 and equation 9, which trade-off between the domain-dependent criterion (instead of the error $E(m)$) and the model complexity $C(m)$

$$ERC(m) = \beta \cdot ER(m) + (1 - \beta) \cdot C(m), \tag{10}$$

$$EXC(m) = \beta \cdot EX(m) + (1 - \beta) \cdot C(m), \tag{11}$$

$$ERXC(m) = \beta \cdot ERX(m) + (1 - \beta) \cdot C(m). \tag{12}$$

For example, as $ER(m)$ combines $E(m)$ and $R(m)$, $ERC(m)$ combines $E(m)$, $R(m)$, and $C(m)$. When $\alpha = 1$, $R(m)$ is not taken into account and the $ERC(m)$ model selection criterion becomes the (domain-independent) trade-off between the model error and complexity, i.e., $EC(m)$.

Evaluation of modeling performance

Before we test our central hypothesis that the domain-specific model selection criteria are best suited for modeling endocytosis, we define the metrics that we use to measure the modeling performance of ProBMoT.

The first performance metric describes the ability of the model selection method to discriminate between the 126 model structures considered by ProBMoT. To measure the discriminative power of a particular model selection criterion, we run a ProBMoT experiment where the given criterion is used to rank the models. We then depict the error profile, i.e., plot the value of the given criterion for each model against the increasing model rank; see Fig. 4 for an example. Furthermore, we refer to the initial flat region of the error profile as the plateau; its length equals the number of models it contains. A simple heuristic for detecting the plateau is the test whether there is more than 10 % error difference between two consecutive points. The first such difference indicates the end of the plateau. For example, the plateau of the error profile in Fig. 4 contains 62 models. Note that the plateau represents the set of top-ranked model structures that are indistinguishable in terms of the model selection criterion used to rank them. The fewer models in the plateau, the better the performance of the model selection criterion, i.e., its ability to discriminate between the candidate model structures.

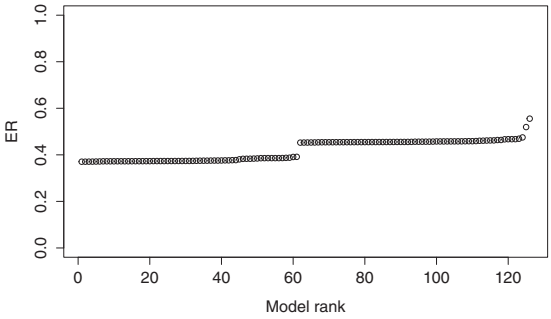

Fig. 4 Error profile. Sorted ranking of the 126 models according to the estimated values of the *ER* criterion. The trade-off parameter setting is $\alpha = 0.5$. Two long and two short plateaus can be identified in this error profile

Second, we compare the model structures in the first plateau with the three groups of the models that have been identified and grouped by their ability to produce bistable behavior [15]. The first group includes 26 models that can reproduce the bistable switch behavior from Rab5- to Rab7-dominated steady states, some of them follow a toggle switch, others a cut-out switch. We will refer to this first group of models as COT. The second group includes 18 models that follow an in-phase switch; we will refer to this second group of models as IP. The third groups includes 10 models that can not reproduce a bistable switch behavior; we will refer to this third group of models as NOBS. For the model structures in the first plateau, we are going to investigate the average rank of the models in each of these three groups as an indicator of the performance of our approach. We expect higher average rank and number of models in the first plateau that belong to the COT and IP groups, and relatively smaller average rank and number of models from the NOBS group. In order to make a fair evaluation of the performance of our approach and the chosen model selection criteria we completely exhaust all previously identified structural possibilities. Since stability analysis has not been performed on the remaining 72 model structures, we consider them to be in a separate fourth group.

Third, we analyze the structure and behavior of the best models. We aim at identifying the structure patterns for the models in the first plateau: to this end, we analyze the frequencies of the different modeling choices in the first-plateau models for the four functions of GEF_5, GAP_5, GEF_7 and GAP_7. We also report the structure and the simulated behavior of the top-ranked model. We repeat this analysis for the models in the first plateau that also belong to the COT and the IP groups.

Finally, we consider the problem of practical parameter identifiability, i.e. the uniqueness of the estimated parameters for a candidate model given the available measured data. A systematic study of a large number of systems biology models [25] and previous studies of the problem of identification of the model of the Rab5-Rab7 switch in endocytosis [16], indicate identifiability problems: Parameters in models from the area of systems biology are uncertain in general and the model proposed in the original study has specific practical parameter identifiability problems. Nevertheless, we investigate the possibility of further discrimination of the models based on this property, and the possible improvement of the identifiability given the best found combination of domain-dependent and independent criteria for optimization.

We follow the bootstrap method, proposed by Joshi et al. [26], to perform the parameter identifiability analysis, choosing it for several reasons. First, it provides more reliable estimates of the parameter confidence intervals compared to, for example, the Fisher-Information-Matrix based method. Second, it is better suited for highly nonlinear models with high parameter-value uncertainties. Third, the same method was used to perform parameter identifiability of an endocitosys model [16]. Note however, that the bootstrap method comes with a high computational cost since it requires a large number of parameter estimations on the same model structure using different data set with added random noise at a certain noise level. The obtained parameter estimates are then used to analyze the distribution of the values of individual parameters and the corresponding confidence intervals. We perform the parameter identifiability for the three selected models: the top-ranked model, the top-ranked COT model, and the top-ranked IP model.

Ethics approval

No aspect of this study required ethics approval.

Results

In the experiments, we vary the values of the trade-off parameters α and β in the range $[0, 1]$ with a step of 0.1. For each pair of values, we perform a single modeling experiment by running ProBMoT with the corresponding model selection criterion. We analyze the results of the experiments in terms of the performance metrics presented in the previous section.

Data

The data set used in the experiments of modeling endocytosis is derived from the measurements used by del Conte-Zerial et al. [15] and is available in Additional file 3. These include measurements collected by tracking early endosomes in three independent experiments that lead to 28 time courses of Rab5 and Rab7 intensity. The data from different experiments and time courses were then aggregated by carefully performed manual scaling and averaging into two time-series of length 10,571 time points along the time interval of [-5, 300] seconds, where the time point 0 corresponds to the Rab5-Rab7 conversion switch point [15]. Finally, to use the same alignment of the data against the model simulation as in previous studies [15, 16], we shifted the time axis using the transformation $t \leftarrow t + 828.56$.

Note that, due to the limitation of the measurement equipment, only the total (that is active– and passive-state) concentrations of the Rab5 and Rab7 domain proteins are observed: The observed values at each time point correspond to $Rab5 = R5 + r5$ and $Rab7 = R7 + r7$, respectively. Recall from Equation (1) that $R5, R7$ and $r5, r7$ correspond to the concentrations of the active (GTP-bound) and passive (GDP-bound) state of the Rab domain proteins, respectively. To deal with the limited observability of the system variables in the ProBMoT model, we define its outputs

as `(rab5.GDP_bound_state_conc + rab5.GTP_bound_state_conc) * K` and `(rab7.GDP_bound_state_conc + rab7.GTP_bound_state_conc) * K`, where K denotes a scaling factor that allows for proper matching of the measured data against the simulated model outputs. Note that the range of values of K considered by ProBMoT is $[10^3, 10^5]$ [16].

Domain-independent model selection

We start by analyzing the modeling results obtained with the default ProBMoT selection criterion of E, which corresponds to the setting of the trade-off parameters $\alpha = 1$, $\beta = 1$. As expected, all model errors are in a very narrow range, shown in the plateau and box-plot in Fig. 5. The plateau of size 113 shows that almost 90 % of all candidate models are indistinguishable in terms of the E criterion.

One of the approaches to distinguish between model structures is to perform model selection using both model error and complexity, i.e., using the EC model selection criterion. The distribution of the complexity of the models is shown in Additional file 1: Figure S2. Figure 6 shows the influence of the β trade-off parameter on the plateau size (black line-points) and the average ranks of the COT (green line-points), IP (yellow line-points) and NOBS models (red line-points). Note that small β values lead to short plateaus including only the simplest model structures, i.e., those including six processes, indicating a strong preference towards simple models. The simulated behavior of these models differs significantly from the expected bi-stable switch behavior. On the other hand, high β values lead to modeling performance comparable to or worse than the model selection criterion E.

Domain-dependent model selection

In a similar manner, we explore the performance of the ER, EX and ERX model selection criteria that trade-off between model error and model fit to the desired behavior of the hidden system variables. We find that the domain-dependent criteria lead to remarkable improvement in discriminative power over the domain-independent model selection criteria.

Figure 7 shows the influence of the change of the trade-off parameter α on the plateau size and the average ranks of the COT, IP and NOBS models in the list of models ranked using the ER criterion. Small and large values of α lead to large plateaus, with a significant drop of the plateau size for $\alpha = 0.4$ and a minimum at $\alpha = 0.5$. Note that this value also leads to the smallest average ranks of the plausible model structures. Additional file 1: Figure S3 provides further details on the results of the modeling experiment using the ER criterion with $\alpha = 0.5$. The size of the plateau is 62, i.e., less than 50 % of all the candidate models; a significant improvement in discriminative power over the 90 % obtained with E. Out of these 62 models, 13 have structures belonging to the COT group, 8 to the IP group, and 6 to the NOBS group. The range of errors is tight with a mean value of 0.42, a median of 0.45 and a standard deviation of 0.04. Note that the obtained behavior of some of the models in the first plateau can be considered as unsatisfactory, for example, the simulation of one of the active-state concentrations of the proteins can be uncorrelated to the corresponding measured density even though the correlation is taken into account within the ER criterion during optimization. We believe that this is due to the strong influence of the E component in the used criterion, combined with the imperfect optimization and the identifiability issues presented below.

Figure 8 shows the influence of the α value on the plateau size and the average ranks of the COT, IP and NOBS models in the list of models ranked using the EX criterion. The curve corresponding to the plateau size has a similar saddle-like shape as the one for the ER criterion from Fig. 7. The smallest plateau size is obtained for $\alpha = 0.9$. The size of the plateau is 42, reducing the percentage of candidate models in the plateau to 33 %, which represents a further improvement over the ER criterion.

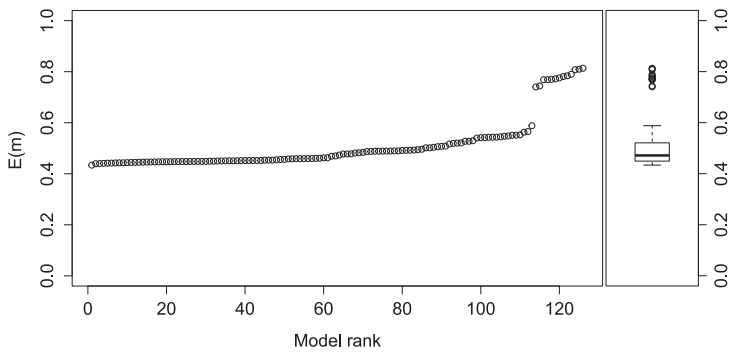

Fig. 5 Error profile and a box plot of the error obtained using the criterion E. Sorted ranking of the 126 models according to the estimated values of the E criterion

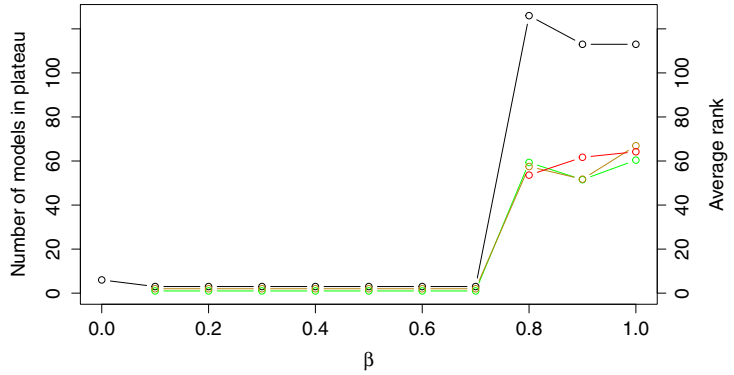

Fig. 6 The size of the error-profile plateau (black line) and the average ranks of the structures belonging to the COT (green line), IP (yellow line) and NOBS (red line) group obtained using the criterion *EC*. The plot is obtained by varying the values of the β trade-off parameter in the range [0, 1] with an increment of 0.1

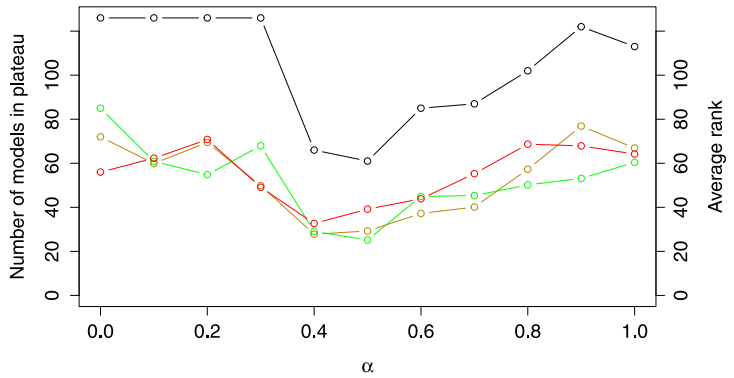

Fig. 7 The size of the error-profile plateau (black line) and the average ranks of the structures belonging to the COT (green line), IP (yellow line) and NOBS (red line) group obtained using the criterion *ER*. The plot is obtained by varying the values of the α trade-off parameter in the range [0, 1] with an increment of 0.1

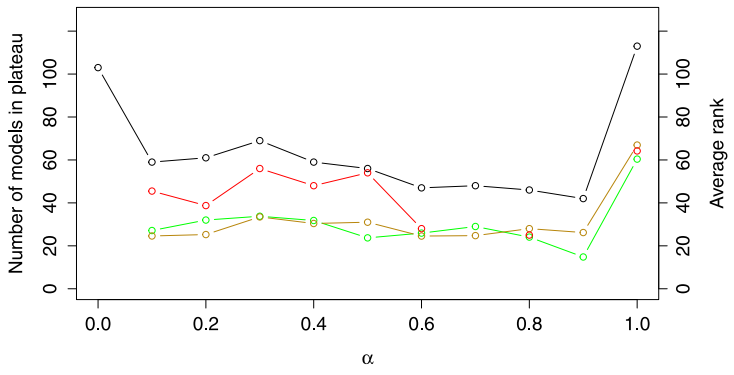

Fig. 8 The size of the error-profile plateau (black line) and the average ranks of the structures belonging to the COT (green line), IP (yellow line) and NOBS (red line) group obtained using the criterion *EX*. The plot is obtained by varying the values of the α trade-off parameter in the range [0, 1] with an increment of 0.1

Additional file 1: Figure S4 provides details on the results of the modeling experiment using the *EX* criterion with $\alpha = 0.9$. For values of $\alpha = 0.7$ and $\alpha = 0.9$, we find no structures belonging to the NOBS group in the plateau. In the smallest plateau, out of the 42 models, 15 have structures belonging to the COT group, 11 to the IP group and none to the NOBS group. The range of errors is significantly wider in comparison to the best case using the *ER* criterion with a mean equal to 0.65, a median of 0.73 and a standard deviation of 0.33, which leads to the overall conclusion of significantly improved discriminative power. In contrast to the experiments using the *ER* criterion, the behavior of the models in the plateau, regarding the optimized point of switch, is within the boundaries of the expected, i.e. there is no unsatisfactory behavior.

Combining the two domain-dependent criteria brings further improvements. Figure 9 shows the influence of α on the plateau size and the distribution of the ranks of the plausible model structures in the plateau using the *ERX* criterion. We observe a smooth saddle like shape of the plateau size as a function of α. The smallest plateau size is obtained for $\alpha = 0.5$. The size of this plateau is 33, reducing the percentage of candidate models in the plateau down to 26 %. Additional file 1: Figure S5 provides details on the results of the modeling experiment using the *ERX* criterion with $\alpha = 0.5$. There are no structures shown to not achieve bistable behavior in the plateau for values of alpha larger than 0.3 and smaller than 1.0. In comparison to using the *EX* criterion, the use of the combined *ERX* criterion leads to a slightly smaller number of models that have been shown to reproduce bistable behavior, slightly tighter range of error values and improved overall quality of the models regarding their fit to the data and the dynamic behavior of the components of the system. In the smallest plateau, out of the 33 models, 10 have structures belonging to the COT group, 7 to the IP group and none to the NOBS group. The range of errors has a mean equal to 0.47, a median of 0.44 and a standard deviation of 0.27. Using the combined criterion, no models in the plateau produce unsatisfactory behavior.

Going one step further, we combined the best performing domain-dependent criterion *ERX* (for $\alpha = 0.4$) with the normalized model complexity, to experiment with the combined *ERXC* criterion. Figure 10 shows the results of the experiments with varying values of the trade-off parameter β. They are similar to the case of using the *EC* criterion.

For the problem of modeling the Rab5-Rab7 switch in endocytosis, no further improvements of discriminative power can be achieved by considering the complexity of the model structure for model selection. The optimized values for each of the used criteria are uncorrelated to the complexity of the model structures. The distribution of errors for each criterion for each possible complexity of the model structures can be seen in Additional file 1: Figure S6.

Overall, the comparisons of the modeling results obtained using different values of α and β reveal that the *ERX* modeling criterion with $\alpha = 0.5$ has the best ability to discriminate between the candidate model structures.

For completeness of the results, Additional file 1: Table S1 presents the values of all the components of the combined *ERX* criterion, i.e., E, R and X for the 33 models in the first plateau of the *ERX* error profile from Additional file 1: Figure S5. Additional file 1: Table S2 presents the values of all the components of the *ERX* criterion for the least complex and the most complex models.

Analysis of the obtained models

We begin the analysis of the best obtained models, i.e., those in the first plateau of size 33, by analyzing the distribution of the components of their structures. The

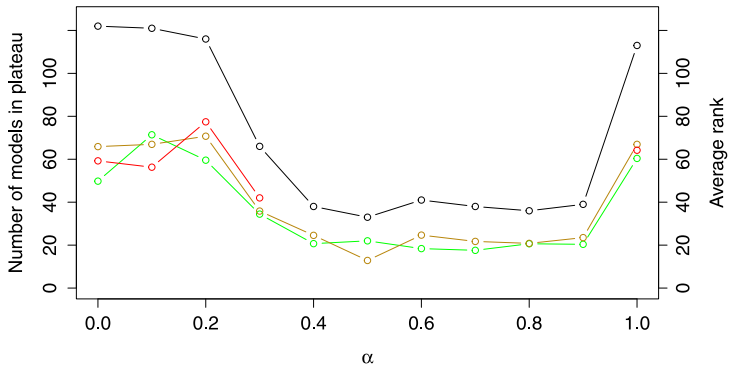

Fig. 9 The size of the error-profile plateau (black line) and the average ranks of the structures belonging to the COT (green line), IP (yellow line) and NOBS (red line) group obtained using the criterion *ERX*. The plot is obtained by varying the values of the α trade-off parameter in the range [0, 1] with an increment of 0.1

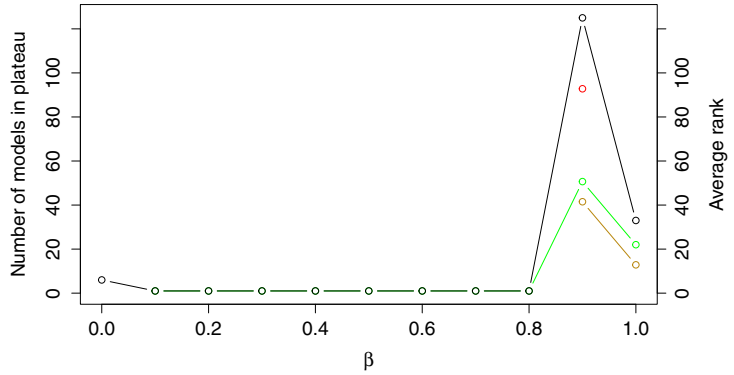

Fig. 10 The size of the error-profile plateau (black line) and the average ranks of the structures belonging to the COT (green line), IP (yellow line) and NOBS (red line) group obtained using the criterion *ERXC*. The plot is obtained using the value $\alpha = 0.5$ and by varying the values of the β trade-off parameter in the range [0, 1] with an increment of 0.1

distribution is shown in Fig. 11. For the entire plateau of models, it can be seen that there is a major shift in distribution for the GEF5 functional forms in favor of the Sigmoidal response, which is even more obvious when considering the distribution in the major classes of bistable models in the plateau. A minor shift in distribution is present in the GAP5 functional forms favoring the Sigmoidal response, which can be also observed in the distribution for the major classes of bistable models. In general, the evidence is in favor of positive regulation

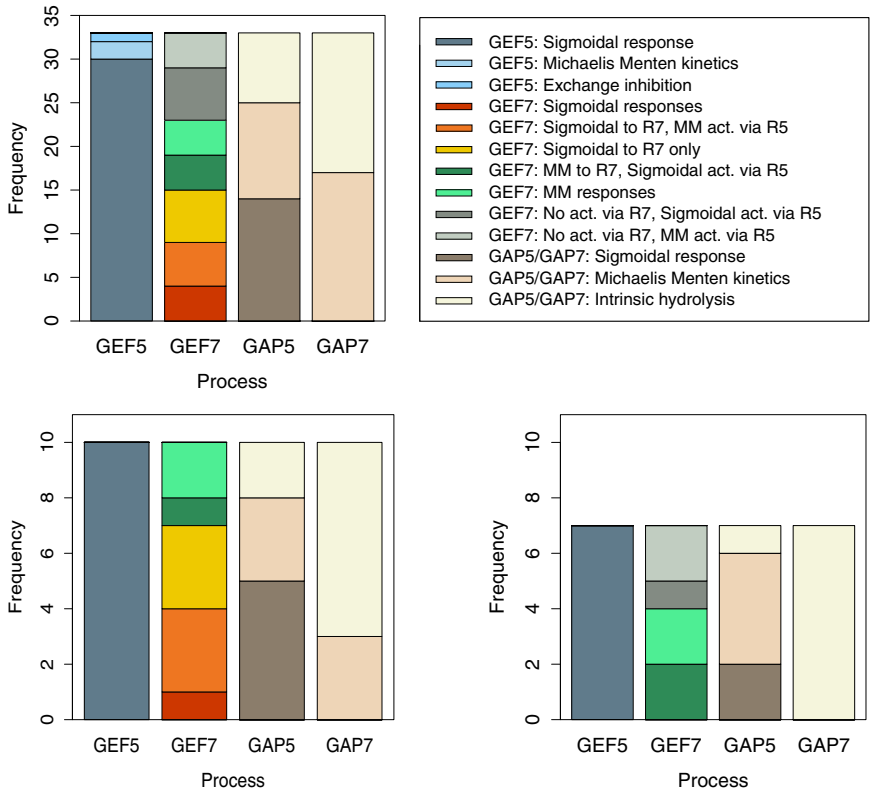

Fig. 11 Distribution of the structural components of the models in the plateau using the *ERX* criterion with $\alpha = 0.5$. The distribution of the components of all models (top left), the distribution of the components of the models in the plateau belonging to the COT group (bottom left) and the distribution of components belonging to the IP group (bottom right)

of the hydrolysis of active-state to passive-state Rab5 via Rab7 as opposed to no regulation (intrinsic hydrolysis).

While there is no significant shift in distribution for the GEF7 and GAP7 functional forms in the entire plateau, there are important differences in the specific groups of models. For the GEF7 process in models of the COT group an auto-catalytic process of exchange must be present. We observe a more frequent Sigmoidal than Michaelis-Menten response to the active-state Rab7. In models of the IP group, the requirement for an auto-catalytic process is not apparent. If it is present, however, it takes the form of a Michaelis-Menten response to the active-state Rab7. It can be also seen that the exchange of passive– to active-state Rab7 is positively regulated by active-state Rab5 in all cases.

For the GAP7 process, for both COT and IP group, the Intrinsic hydrolysis alternative for the GAP7 process is favored. This is indicative of an absence of regulation of the hydrolysis of active-state to passive-state Rab7 via Rab5, which is especially clear in the case of models from the IP group.

We next take a closer look at a sample of three endocytosis models from the plateau. We consider the top-ranked model overall and the best locally ranked models in the first plateau from each class of models.

Figure 12 depicts the structure of the top-ranked model overall, its simulated output behavior compared with the measurements, and the simulated behaviors of the hidden system variables (R_5, R_7, r_5 and r_7) representing the concentrations of the active and passive states of the protein domains. The simulation of the total densities of the protein domains has a reasonable fit to the measured data. The structure of the model leads to a switch behavior due to the strong influence of Rab5. However, there is no feedback mechanism which will allow for transition from one to another stable behavior.

Figure 13 depicts the structure of the top-ranked model having a structure belonging to the COT group. It is ranked as fourth overall. The simulation of the total densities has a good fit to the measured data, both qualitatively and quantitatively indistinguishable from the simulation of the top-ranked model. The simulation of the active and passive components of both protein domains achieve the expected behavior. The dynamics of the active states of the protein domains drives the dynamics of the system and their switching time corresponds to the switching time observed in the measurements. The passive state concentrations remain stable throughout the time of simulation.

The structure of the model allows for a cut-out switch behavior due to the strong positive influence of Rab5 on the exchange of passive to active-state Rab7 combined with auto-activation of the exchange, which overpowers the influence of Rab5 on the hydrolysis of Rab7 on one hand, and the negative feedback from Rab7 to Rab5, which leads to low concentrations of active-state Rab5 on the other.

Figure 14 depicts the structure of the top-ranked model having a structure belonging to the IP group. It is ranked

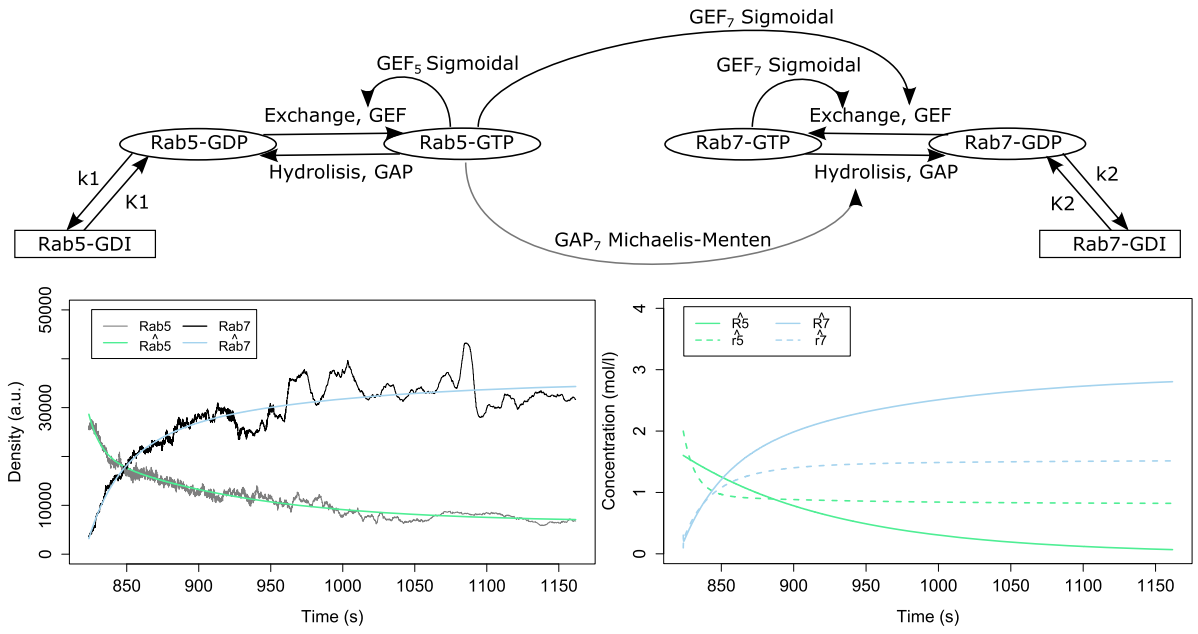

Fig. 12 The structure (top), the output behavior (bottom left) and the behavior of the active and passive state protein concentrations (bottom right) of the top-ranked model, obtained using the *ERX* criterion with $\alpha = 0.5$. The model has an error *ERX* = 0.126

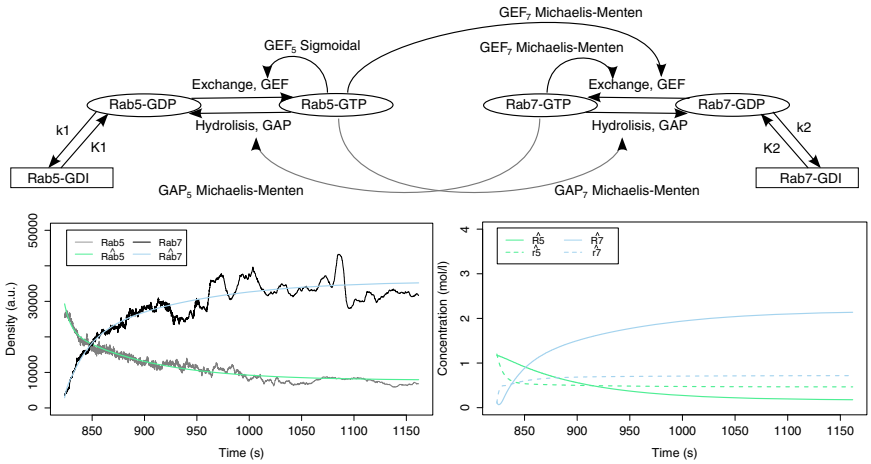

Fig. 13 The structure (top), the output behavior (bottom left) and the behavior of the active and passive state protein concentrations (bottom right) of the best ranked model from the COT group, obtained using the *ERX* criterion with $\alpha = 0.5$. The model is ranked fourth overall and has an error $ERX = 0.129$

fifth overall. As with the previous models, the simulation of the total densities has a good fit to the measured data. The simulation of the active and passive state component concentrations is qualitatively like the one of the previously discussed model. The structure reveals the reason for the similar behavior.

Compared to the best COT model, the best IP model is missing only a GAP_7 interaction. The other present interactions have the same functional forms. The dynamics and the bistable behavior arise from the same sources discussed above, lacking only the negative feedback from the active-state Rab5 via GAP_7.

The practical parameter identifiability analysis performed on the selected model structures shows, as expected, parameter identifiability problems. Although there is a slight improvement in the relative size of the confidence interval to the mean and the estimates for all models, overall the conclusions from our results correspond with the conclusions from previous experiments [16] on a related model.

The summarized statistics of the identifiability analysis for each model can be seen in Additional file 1: Tables S3–S5. The uncertainties (length of the 95 % confidence interval) are large for a significant number of parameters values for all functions, independent of the functional alternative in the selected models.

The shape of the distribution of the parameters differs significantly in most of the cases from the normal

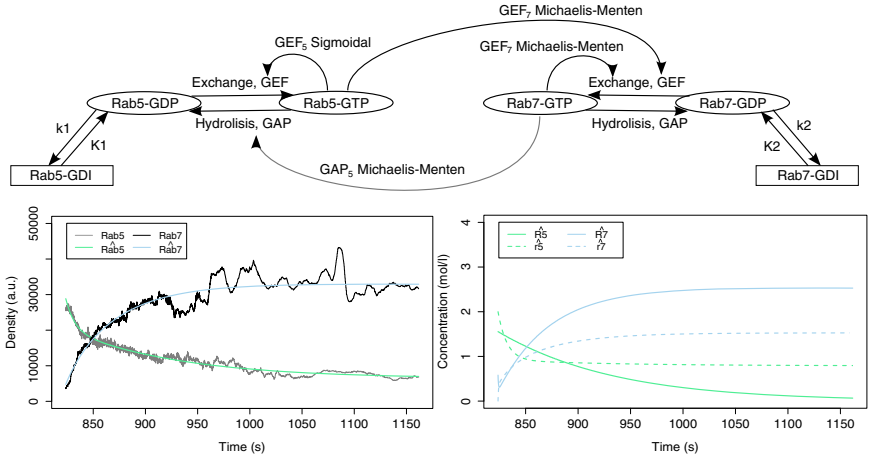

Fig. 14 The structure (top), the output behavior (bottom left) and the behavior of the active and passive state protein concentrations (bottom right) of the best ranked model from the IP group, obtained using the *ERX* criterion with $\alpha = 0.5$. The model is ranked fifth overall and has an error $ERX = 0.129$

distribution, indicating non-linearity of the systems with respect to their corresponding parameter values; see the histograms shown in Additional file 1: Figures S8, S10 and S12. This difference is most evident in the top-ranked model belonging to the IP group in contrast to the shapes of the distributions of parameter values of the top-ranked model structure. For the majority of the parameters, their values were most frequently estimated to be in close proximity to the bounds of the allowed range.

The correlation matrices for all model structures (Additional file 1: Figures S7, S9 and S11) show high absolute correlation values for certain sets of parameters. In all three models, we observe high correlation of the association rate and the dissociation flux of the proteins with GDI.

In the top-ranked model, there is high correlation between the estimated values of the parameters of the auto-catalysis component of GEF7 and between the estimated values of the GAP7 intrinsic hydrolysis rate and the maximum rate parameter in the Michaelis-Menten term. There is a high positive correlation between the estimated initial values of the active and passive states of Rab5 and a high negative correlation between the estimated initial values of the active and passive states of Rab7.

In the top ranked model from the COT group, we observe a high correlation between the intrinsic hydrolysis rate and the maximum rate parameter in the Michaelis-Menten term values in both the GAP5 and the GAP7 functions. As for the top-ranked model, there is a high positive correlation between the estimated initial values of the active and passive states of Rab5 and a high negative correlation between the estimated initial values of the active and passive states of Rab7.

In the top ranked model from the IP group, we observe high correlation between all of the parameters of the GEF5 and GAP5 function. There is a positive correlation between the estimated initial values of the active and passive states of Rab5.

Discussion

The combination of limited noisy observations, on one hand, and the expectations about the behavior of the unobserved system variables, on the other, poses a difficult model selection problem. We approach this problem by combining several criteria for model selection. Two are the standard model selection criteria of model error and simplicity and three are based on the expected behavior of hidden system variables.

The comparison of different criteria shows that the simplicity-based criterion leads to little or no improvement of discriminative power; the majority of the model structures remain indistinguishable. This is also evident from the low correlation of the optimized values for each of the used criteria and the complexity of the model structures. The plateaus are not a result of over-fitting and cannot be avoided by considering the principle of parsimony. On the other hand, a combination of a domain-independent least-squares based optimization criterion with a simple problem-specific criterion is better suited to the real-world problem at hand than the simplicity-based criterion. In our experiments, the combination of the domain-independent criterion with two different domain-dependent criteria leads to additional improvement. The introduction of domain-specific criteria leads to significantly improved selectivity of the process-based modeling algorithm. In the case of modeling endocytosis, this improvement is evident from the absence of those models which have been previously shown to have monostable behavior (NOBS group), whose average rank (or lack thereof) in the plateau we show in red color in the plots for each criterion.

The simulation of the dynamics of both the observed total density and the unobserved states of the protein domains provides a good fit to the measured data and expected dynamical behavior of the components of the system. This property is consistent in the best ranked models. Due to the existing parameter identifiability problems in all selected representative models, further discrimination (based on the identifiability) cannot be made.

A number of models in the first plateau (even in the experiment using the combination of criteria that has the highest selectivity) do not belong to any of the COT, IP or NOBS groups. Among these, there are some that might be considered as structurally flawed under some expectations for structural mechanisms as is the case with the missing feedback mechanism in the top-ranked model. The presence of these structures may be (in part) a result of overfitting due to the complex representation of processes, the number of free parameters, the limited observability, and the quality of the data. Nevertheless, some of these previously identified (but not considered) models, given their performance, might lead to the reconsideration of parts of or their complete structure in further studies.

We consider the introduction of domain-specific criteria and the performed comparison to be an important step towards improved automated modeling approaches and a solution of the model selection problem. The majority of model selection criteria employed in the domain of systems biology are based either on likelihood, on the Bayesian principle or a combination of the previous [7], due to their well-established reputation in other areas. Most of them have the principle of parsimony implicitly encoded. On the other hand, in biology, the principle of parsimony should be sometimes set aside in favor of selecting better (although more complex) explanatory models [27]. We argue that knowledge-based,

domain-specific criteria for model selection should be considered prior to or in conjunction with approaches based on the parsimony principle. These criteria can offer solid alternative solutions for the model selection problem in scenarios with limited observability and noisy data.

However, domain specific criteria for model selection should always be carefully chosen, based on solid background, and their influence on the final selection decision should be carefully weighted. Combined with global heuristic parameter estimation approaches, as used in this study, inattentively chosen criteria might shift the solution to an unwanted direction. Incorrect weighting, on the other hand, might aggravate the selection problem by under– or over-fitting of the candidate models.

Conclusion

We have demonstrated the applicability of the automated modeling tool ProBMoT to the real-world problem of modeling the Rab5-Rab7 conversion switch in the important cellular process of endocytosis. By using ProBMoT, we improve upon the classical modeling approach by using domain-specific knowledge, good practices, and automation. While the applicability of ProBMoT and other modeling approaches has been illustrated before [13], this is the first study focusing on the problem of model selection. In these terms, we go beyond the work of Čerepnalkoski et al. [17] and Tashkova et al. [16] and make a step further towards elucidating the problem of model selection in the context of automated modeling of dynamical systems.

Furthermore, we show that ProBMoT is able, in an automated fashion and using a combination of knowledge– and data-driven modeling, to solve a complex, relevant and challenging problem from the domain of systems biology. We analyze its utility by comparing the results of automated modeling with the ones obtained in a manual modeling experiment. In this way, we evaluate both the automated approach and the manual modeling process. The results show that ProBMoT is able to reconstruct the results of the manual experiment by using limited and noisy observations of the modeled system. The modeling experiments presented here confirm the finding that a group of model structures able to achieve a cut-out or toggle switch behavior explains the available data. We also show that another group of model structures (IP group), previously considered less plausible, and a number of previously not considered model structures, are still equally capable of reproducing the observations and expectations and should still be considered as relevant.

We identify several points for further work. Additional criteria, more complex than the considered one, which complement the information about the model fit to the measured data should be considered. Such criteria can be based on the properties of the model structure: Del Conte-Zerial et al. [15] perform e.g. bifurcation and phase plane analysis on each model structure, after which they dismiss the structures that lack certain properties.

A similar effect can be achieved by apriori filtering of candidate model structures based on their structural properties. The constraining of the domain knowledge based on valid assumptions and the introduction of specific knowledge related to the problem at hand, will result in a reduced number of candidate models to be fitted. This will reduce the computational time needed for the experiments and facilitate the model selection problem. However, as shown trough our experiments, with the use of a domain-specific criteria, the automated process-based modeling achieves high selectivity even in the presence of unfiltered model structures.

Finally, the automated modeling approach can be used to gain knowledge about other dynamical systems, i.e., other parts of the endocytic pathway. The gained knowledge can contribute to the development of a complete explanatory model of endocytosis. By performing experiments on other real-world problems, additional insight into the process of automated modeling can be obtained. This will further improve the used approaches, which can in turn be used to discover better explanatory models.

Additional files

Additional file 1: Supplementary material. This file contains supplemental figures and tables.

Additional file 2: ProBMoT Library, Incomplete Model and Task files. This file contains the complete library, the incomplete model and the task used for modeling the Rab5-Rab7 switch.

Additional file 3: Data. This file contains space delimited time points and measurements used for fitting the model structures.

Abbreviations

ProBMoT: Process Based Modeling Tool; GDP: Guanosine diphosphate; GTP: Guanosine triphosphate; GDI: Guanosise diphosphate Dissociation Inhibitor; GEF: Guanine nucleotide Exchange Factor; GAP: GTPase-Activating Protein; COT: Cut-Out/Toggle switch; IP: In-Phase switch; NOBS: NO Bistable Switch.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

LT and SD initiated the work. YK provided the data and domain knowledge. JT encoded the knowledge in the process-based formalism. JT designed and performed the experiments. JT and LT analyzed the results. JT and LT drafted the manuscript. YK and SD gave critical advice on how to revise the manuscript. All authors read and approved the final manuscript.

Acknowledgements

The real experimental data (total protein concentrations of Rab5 and Rab7) come from the group of Marino Zerial at the Max-Planck Institute for Cell Biology and Genetics in Dresden, Germany. We acknowledge the financial support of the Slovene Human Resources Development and Scholarship

Fund, the Slovenian Research Agency (Grants P2-0103 and P5-0093 (B)), and the European Commission (Grants ICT-2013-612944 MAESTRA and ICT-2013-604102 HBP).

Author details
[1] Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia. [2] Jožef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia. [3] University of Ljubljana, Gosarjeva ulica 5, 1000 Ljubljana, Slovenia. [4] Max Planck Institute of Molecular Cell Biology and Genetics, Pfotenhauer Straße 108, 01308 Dresden, Germany.

Received: 6 January 2015 Accepted: 2 June 2015
Published online: 26 June 2015

References

1. Machado D, Costa RS, Rocha M, Ferreira EC, Tidor B, Rocha I. Modeling formalisms in systems biology. AMB Express. 2011;1(1):1–14.
2. Fisher J, Henzinger TA. Executable cell biology,. Nat Biotechnol. 2007;25(11):1239–1249.
3. Moles CG, Mendes P, Banga JR. Parameter estimation in biochemical pathways: a comparison of global optimization methods. Genome Res. 2003;13:2467–474.
4. Sun J, Garibaldi JM, Hodgman C. Parameter estimation using metaheuristics in systems biology: a comprehensive review. IEEE/ACM Trans Comput Biol Bioinform. 2012;9(1):185–202.
5. Hoops S, Sahle S, Gauges R, Lee C, Pahle J, Simus N, Singhal M, Xu L, Mendes P, Kummer U. COPASI - COmplex PAthway SImulator. Bioinformatics. 2006;22(24):3067–074.
6. Funahashi A, Tanimura N, Morohashi M, Kitano H. CellDesigner: A process diagram editor for gene-regulatory and biochemical networks. BIOSILICO. 2003;1(5):159–62.
7. Kirk P, Thorne T, Stumpf MP. Model selection in systems and synthetic biology. Curr Opin Biotechnol. 2013;24(4):767–74.
8. Langley P, Simon HA, Bradshaw GL, Zytkow JM. Scientific discovery: computational explorations of the creative processes. Cambridge: MA: The MIT Press; 1987.
9. Todorovski L, Džeroski S. Declarative bias in equation discovery. In: Proceedings of the fourteenth international conference on machine learning. San Francisco, CA: Morgan Kaufmann; 1997. p. 376–84.
10. Džeroski S, Todorovski L. Logical and computational aspects of model-based reasoning In: Magnani L, Nersessian NJ, editors. Dordrecht, The Netherlands: Kluwer Academic Publishers; 2002. p. 227–47.
11. Bridewell W, Langley P, Todorovski L, Džeroski S. Inductive process modelling. Mach Learn. 2008;71:109–30.
12. 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. Palo Alto: AAAI Press; 2005. p. 892–7.
13. Džeroski S, Todorovski L. Equation discovery for systems biology: Finding the structure and dynamics of biological networks from time course data. Curr Opin Biotechnol. 2008;19(4):360–8.
14. Džeroski S, Todorovski L. Modeling the dynamics of biological networks from time course data In: Choi S, editor. Systems Biology for Signaling Networks. New York: Springer; 2010. p. 275–94.
15. Del Conte-Zerial P, Brusch L, Rink J, Collinet C, Kalaidzidis Y, Zerial M, Deutsch A. Membrane identity and GTPase cascades regulated by toggle and cut-out switches. Mol Syst Biol. 2008;4:. doi:10.1038/msb.2008.45.
16. Tashkova K, Korošec P, Šilc J, Todorovski L, Džeroski S. Parameter estimation with bio-inspired meta-heuristic optimization: Modeling the dynamics of endocytosis. BMC Syst Biol. 2011;5(1):159. doi:10.1186/1752-0509-5-159.
17. Čerepnalkoski D, Taškova K, Todorovski L, Atanasova N, Džeroski S. The influence of parameter fitting methods on model structure selection in automated modeling of aquatic ecosystems. Ecol Model. 2012;245: 136–65. doi:10.1016/j.ecolmodel.2012.06.001.
18. Durillo JJ, Nebro AJ. jmetal: A java framework for multi-objective optimization. Adv Eng Softw. 2011;42:760–71.
19. Hindmarsh AC, Brown PN, Grant KE, Lee SL, Serban R, Shumaker DE, Woodward CS. SUNDIALS: suite of nonlinear and differential/algebraic equation solvers. ACM Trans Math Softw. 2005;31(3):363–96. doi:10.1145/1089014.1089020.
20. Tanevski J, Todorovski L, Kalaidzidis Y, Džeroski S. Inductive process modeling of Rab5-Rab7 conversion in endocytosis. In: Proceeedings of the sixteenth international conference on discovery science. Berlin: Springer; 2013. p. 265–80.
21. Toni T, Welch D, Strelkowa N, Ipsen A, Stumpf MPH. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. J R Soc Interface. 2009;6:187–202.
22. Storn R, Price KV. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim. 1997;11(34):341–59.
23. Rink J, Ghigo E, Kalaidzidis Y, Zerial M. Rab conversion as a mechanism of progression from early to late endosomes. Cell. 2005;122:735–49.
24. Hastie T, Tibshirani R, Friedman JH. The Elements of Statistical Learning, 2nd edn. Berlin: Springer; 2009.
25. Gutenkunst RN, Waterfall JJ, Casey FP, Brown KS, Myers CR, Sethna JP. Universally sloppy parameter sensitivities in systems biology models. PLoS Comput Biol. 2007;3(10):. doi:10.1371/journal.pcbi.0030189.
26. Joshi M, Seidel-Morgenstern A, Kremling A. Exploiting the bootstrap method for quantifying parameter confidence intervals in dynamical systems. Metab Eng. 2006;8(5):447–55. doi:10.1016/j.ymben.2006.04.003.
27. Westerhoff HV, Winder C, Messiha H, Simeonidis E, Adamczyk M, Verma M, Bruggeman FJ, Dunn W. Systems biology: the elements and principles of life. FEBS Lett. 2009;583(24):3882–890.

4.3 Supplementary Material

Domain-specific model selection for structural identification of the Rab5-Rab7 dynamics in endocytosis –Supplementary material–

Jovan Tanevski, Ljupčo Todorovski, Yannis Kalaidzidis, Sašo Džeroski

Supplemental figures

Supplemental tables

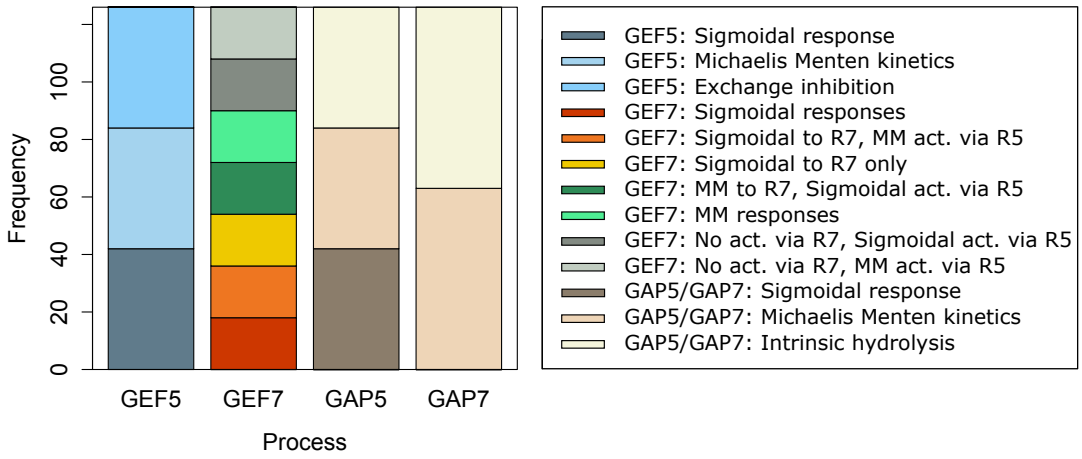

Figure S1: Distribution of the structural components of all 126 candidate models considered in the experiments, as defined in the library.

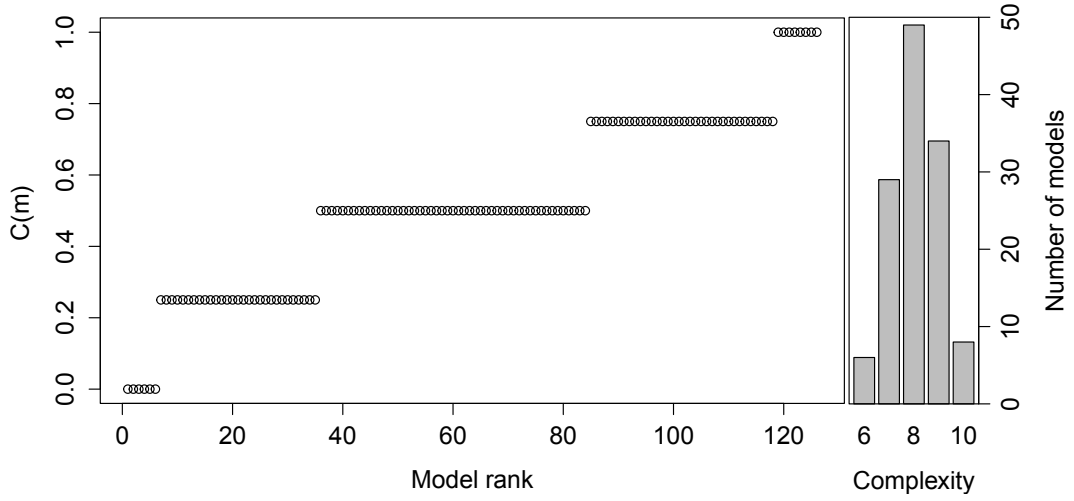

Figure S2: Plot of the sorted complexities of the 126 candidate models. $C(m)$ represents the normalized complexity of the model. The complexity in the histogram is expressed in terms of the number of interactions present in the model.

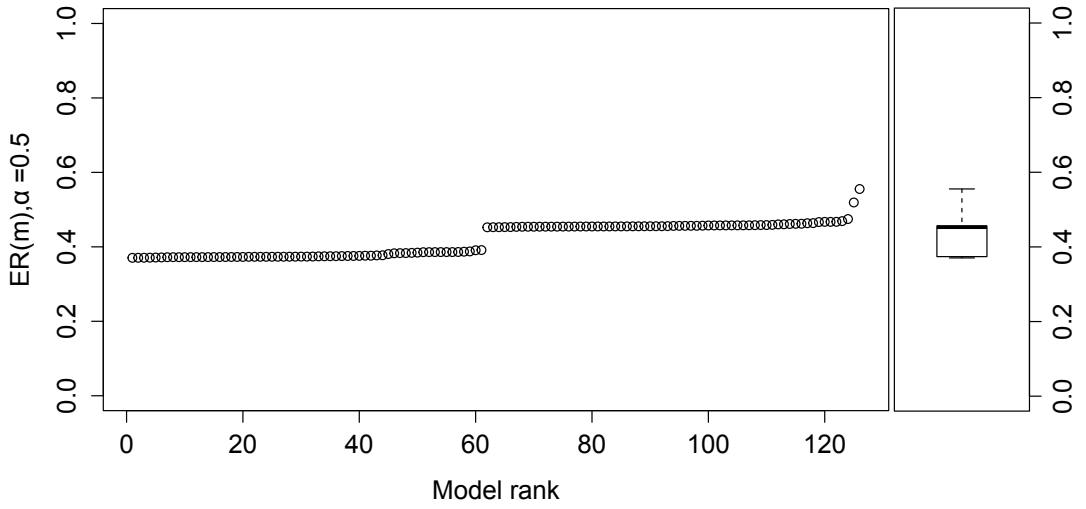

Figure S3: The error profile obtained using the ER criterion and a trade-off parameter setting $\alpha = 0.5$ (left). The box plot shows the distribution of error in the profile (right).

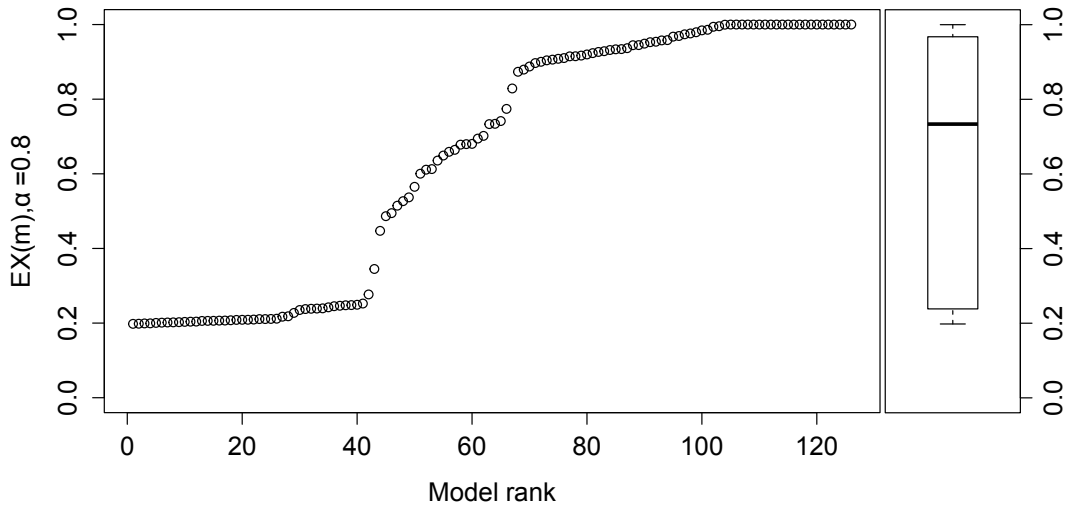

Figure S4: The error profile obtained using the EX criterion and a trade-off parameter setting $\alpha = 0.9$ (left). The box plot shows the distribution of error in the profile (right).

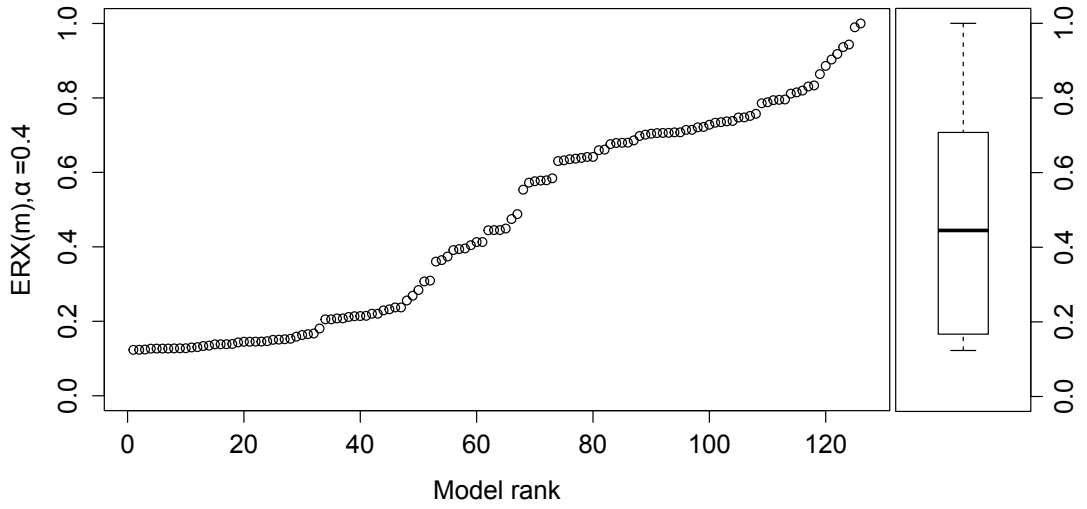

Figure S5: The error profile obtained using the ERX criterion and a trade-off parameter setting $\alpha = 0.5$ (left). The box plot shows the distribution of error in the profile (right).

Table S1: Detailed description of the structure and the performance of the models in the plateau obtained using the ERX criterion with $\alpha = 0.5$.

Rank	Model	Group	Structure					Performance				C
			GEF5	GAP5	GEF7 auto catalysis	GEF7 via Rab5	GAP7	E	R	X	ERX	
1	**74**	**OTHER**	**Sigmoidal_response**	**Intrinsic_hydrolysis**	**Sigmoidal_response**	**None**	**MMenten**	**0.224**	**0.045**	**0.010**	**0.126**	**0.50**
2	77	OTHER	Sigmoidal_response	Intrinsic_hydrolysis	Sigmoidal_response	None	MMenten	0.224	0.045	0.010	0.126	0.25
3	104	OTHER	Sigmoidal_response	MMenten	None	Sigmoidal_response	MMenten	0.226	0.045	0.010	0.127	0.50
4	**86**	**COT**	**Sigmoidal_response**	**MMenten**	**MMenten_kinetics**	**MMenten_kinetics**	**MMenten**	**0.233**	**0.040**	**0.010**	**0.129**	**0.75**
5	**23**	**IP**	**Sigmoidal_response**	**MMenten**	**MMenten_kinetics**	**MMenten_kinetics**	**Intrinsic_hydrolysis**	**0.230**	**0.047**	**0.010**	**0.129**	**0.50**
6	101	OTHER	Sigmoidal_response	MMenten	None	MMenten_kinetics	MMenten	0.228	0.050	0.010	0.129	0.50
7	41	IP	Sigmoidal_response	MMenten	None	Sigmoidal_response	Intrinsic_hydrolysis	0.231	0.045	0.010	0.129	0.25
8	71	OTHER	Sigmoidal_response	Intrinsic_hydrolysis	Sigmoidal_response	MMenten_kinetics	MMenten	0.232	0.041	0.010	0.129	0.50
9	38	IP	Sigmoidal_response	MMenten	None	MMenten_kinetics	Intrinsic_hydrolysis	0.233	0.044	0.010	0.130	0.25
10	60	OTHER	Exchange_inhibition	Sigmoidal	None	MMenten_kinetics	Intrinsic_hydrolysis	0.233	0.044	0.010	0.130	0.50
11	59	IP	Sigmoidal_response	Sigmoidal	None	MMenten_kinetics	Intrinsic_hydrolysis	0.234	0.051	0.010	0.132	0.25
12	125	OTHER	Sigmoidal_response	Sigmoidal	None	Sigmoidal_response	MMenten	0.234	0.051	0.010	0.132	0.50
13	29	COT	Sigmoidal_response	Sigmoidal	Sigmoidal_response	MMenten_kinetics	Intrinsic_hydrolysis	0.241	0.046	0.010	0.134	0.50
14	8	COT	Sigmoidal_response	Intrinsic_hydrolysis	Sigmoidal_response	MMenten_kinetics	Intrinsic_hydrolysis	0.247	0.042	0.010	0.137	0.25
15	44	IP	Sigmoidal_response	Sigmoidal	MMenten_kinetics	MMenten_kinetics	Intrinsic_hydrolysis	0.254	0.044	0.010	0.140	0.50
16	83	OTHER	Sigmoidal_response	Intrinsic_hydrolysis	None	Sigmoidal_response	MMenten	0.259	0.035	0.010	0.141	0.25
17	98	OTHER	Sigmoidal_response	MMenten	Sigmoidal_response	None	MMenten	0.259	0.035	0.010	0.141	0.50
18	95	OTHER	Sigmoidal_response	MMenten	Sigmoidal_response	Sigmoidal_response	MMenten	0.260	0.036	0.010	0.142	0.75
19	107	COT	Sigmoidal_response	Sigmoidal	MMenten_kinetics	MMenten_kinetics	MMenten	0.266	0.041	0.010	0.145	0.75
20	119	OTHER	Sigmoidal_response	Sigmoidal	Sigmoidal_response	None	MMenten	0.260	0.041	0.037	0.150	0.50
21	26	IP	Sigmoidal_response	MMenten	MMenten_kinetics	Sigmoidal_response	Intrinsic_hydrolysis	0.258	0.064	0.010	0.148	0.50
22	5	IP	Sigmoidal_response	Intrinsic_hydrolysis	MMenten_kinetics	Sigmoidal_response	Intrinsic_hydrolysis	0.259	0.063	0.010	0.148	0.25
23	122	OTHER	Sigmoidal_response	Sigmoidal	None	Sigmoidal_response	MMenten	0.270	0.039	0.010	0.148	0.50
24	116	OTHER	Sigmoidal_response	Sigmoidal	Sigmoidal_response	Sigmoidal_response	MMenten	0.273	0.04	0.010	0.149	0.75
25	50	COT	Sigmoidal_response	Sigmoidal	Sigmoidal_response	MMenten_kinetics	Intrinsic_hydrolysis	0.277	0.045	0.010	0.152	0.50
26	56	COT	Sigmoidal_response	Sigmoidal	Sigmoidal_response	None	Intrinsic_hydrolysis	0.281	0.041	0.010	0.153	0.25
27	113	OTHER	Sigmoidal_response	Sigmoidal	Sigmoidal_response	MMenten_kinetics	MMenten	0.282	0.040	0.010	0.154	0.75
28	110	COT	Sigmoidal_response	Sigmoidal	MMenten_kinetics	Sigmoidal_response	MMenten	0.282	0.041	0.010	0.154	0.75
29	53	COT	Sigmoidal_response	Sigmoidal	Sigmoidal_response	Sigmoidal_response	Intrinsic_hydrolysis	0.240	0.154	0.010	0.161	0.50
30	14	COT	Sigmoidal_response	Intrinsic_hydrolysis	Sigmoidal_response	None	Intrinsic_hydrolysis	0.243	0.166	0.010	0.166	0.00
31	4	OTHER	Mmenten_kinetics	Intrinsic_hydrolysis	MMenten_kinetics	Sigmoidal_response	Intrinsic_hydrolysis	0.306	0.050	0.010	0.168	0.25
32	35	COT	Sigmoidal_response	MMenten	Sigmoidal_response	None	Intrinsic_hydrolysis	0.247	0.172	0.010	0.169	0.25
33	124	OTHER	MMenten_kinetics	Sigmoidal	None	Sigmoidal_response	MMenten	0.325	0.062	0.037	0.187	0.50

Table S2: Detailed description of the structure and the performance of the most complex and least complex models in the plateau obtained using the ERX criterion with $\alpha = 0.5$.

Rank	Model	Group	Structure					Performance				C
			GEF5	GAP5	GEF7 auto catalysis	GEF7 via Rab5	GAP7	E	R	X	ERX	
30	14	COT	Sigmoidal_response	Intrinsic_hydrolysis	Sigmoidal_response	None	Intrinsic_hydrolysis	0.242	0.166	0.010	0.166	0.00
38	17	IP	Sigmoidal_response	Intrinsic_hydrolysis	None	MMenten_kinetics	Intrinsic_hydrolysis	0.401	0.044	0.010	0.214	0.00
43	20	IP	Sigmoidal_response	Intrinsic_hydrolysis	None	Sigmoidal_response	Intrinsic_hydrolysis	0.406	0.069	0.010	0.223	0.00
69	111	OTHER	Exchange_Inhibition	Sigmoidal	MMenten_kinetics	Sigmoidal_response	MMenten	1.000	0.200	0.080	0.573	1.00
82	114	OTHER	Exchange_Inhibition	Sigmoidal	Sigmoidal_response	MMenten_kinetics	MMenten	1.163	0.240	0.080	0.662	1.00
86	93	OTHER	Exchange_Inhibition	MMenten	Sigmoidal_response	MMenten_kinetics	MMenten	1.188	0.272	0.075	0.681	1.00
92	96	OTHER	Exchange_Inhibition	MMenten	Sigmoidal_response	Sigmoidal_response	MMenten	1.054	0.636	0.081	0.710	1.00
95	16	OTHER	MMenten_kinetics	Intrinsic_hydrolysis	None	MMenten_kinetics	Intrinsic_hydrolysis	0.982	0.787	0.080	0.710	0.00
96	117	OTHER	Exchange_Inhibition	Sigmoidal	Sigmoidal_response	Sigmoidal_response	MMenten	0.999	0.780	0.080	0.714	1.00
99	19	OTHER	MMenten_kinetics	Intrinsic_hydrolysis	None	Sigmoidal_response	Intrinsic_hydrolysis	1.183	0.440	0.081	0.722	0.00
114	87	OTHER	Exchange_Inhibition	MMenten	MMenten_kinetics	MMenten_kinetics	MMenten	1.197	0.772	0.080	0.812	1.00
116	13	OTHER	MMenten_kinetics	Intrinsic_hydrolysis	Sigmoidal_response	None	Intrinsic_hydrolysis	1.152	0.900	0.076	0.820	0.00
124	108	OTHER	Exchange_Inhibition	Sigmoidal	MMenten_kinetics	MMenten_kinetics	MMenten	1.130	1.000	1.000	1.070	1.00
125	90	OTHER	Exchange_Inhibition	MMenten	MMenten_kinetics	Sigmoidal_response	MMenten	1.172	1.000	1.000	1.086	1.00

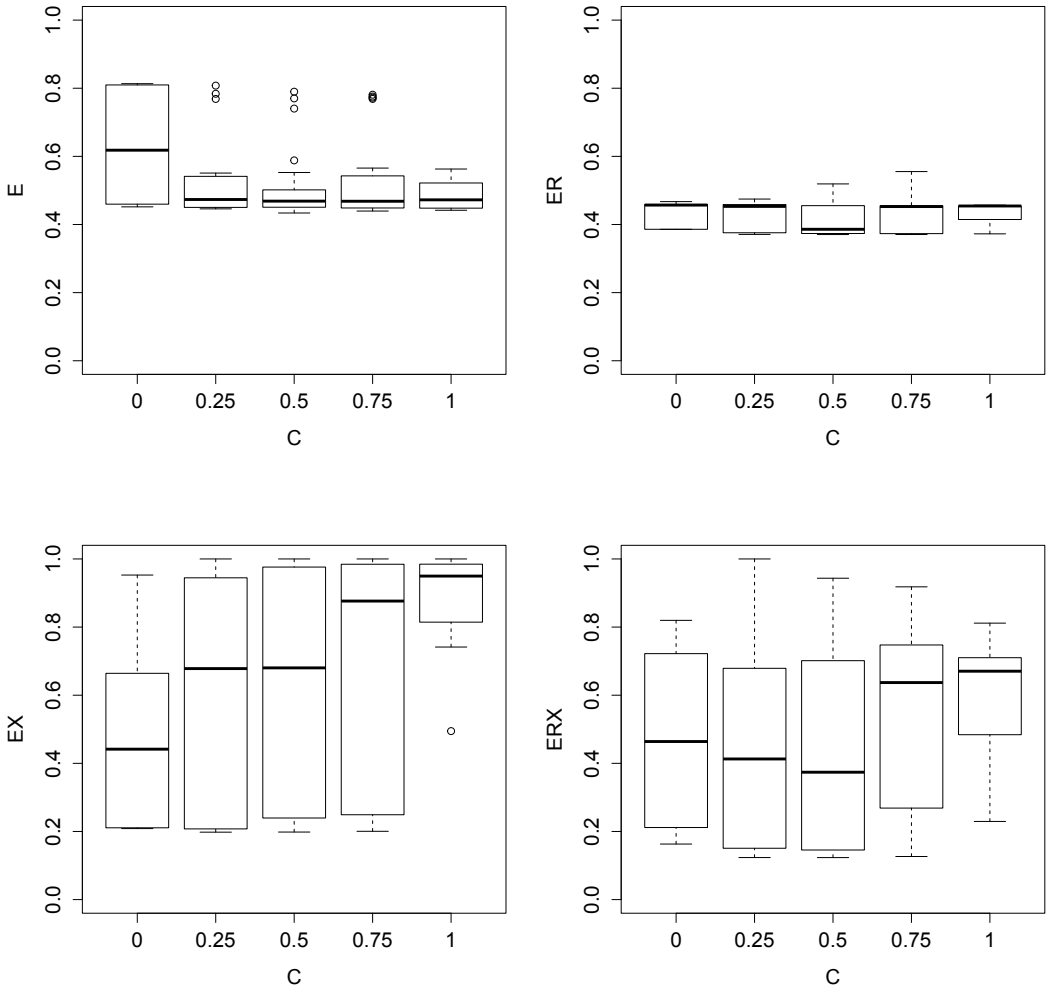

Figure S6: Box plots of the distribution of errors (top) E, ER, (bottom) EX and ERX relative to the complexity C of the model structures. The correlations between the errors and the complexity of the model structures are as follows: $\rho(E, C) = -0.110$, $\rho(ER, C) = -0.159$, $\rho(EX, C) = 0.184$, $\rho(ERX, C) = 0.168$, where ρ is the Spearman's rank correlation coefficient.

Top ranked model:

$$
\begin{aligned}
\frac{dr_5}{dt} &= K_1 - (k_1 + \frac{k_{e,5}}{1 + e^{(k_{g,5} - R_5) \cdot k_{f,5}}}) \cdot r_5 + k_{h,5} \cdot R_5 \\
\frac{dR_5}{dt} &= \frac{k_{e,5}}{1 + e^{(k_{g,5} - R_5) \cdot k_{f,5}}} \cdot r_5 - k_{h,5} \cdot R_5 \\
\frac{dr_7}{dt} &= K_2 - (k_2 + \frac{k_{e,7}}{1 + e^{(k_{g,7} - R_7) \cdot k_{f,7}}} + \frac{k_{E,7}}{1 + e^{(k_{G,7} - R_5) \cdot k_{F,7}}}) \cdot r_7 + k_{h,7} + \frac{k_{H,7} \cdot R_5}{k_{y,7} + R_5} \cdot R_7 \\
\frac{dR_7}{dt} &= \frac{k_{e,7}}{1 + e^{(k_{g,7} - R_7) \cdot k_{f,7}}} + \frac{k_{E,7}}{1 + e^{(k_{G,7} - R_5) \cdot k_{F,7}}} \cdot r_7 - k_{h,7} + \frac{k_{H,7} \cdot R_5}{k_{y,7} + R_5} \cdot R_7
\end{aligned}
\tag{1}
$$

Table S3: Summary of the identifiability analysis for the estimated parameter values in the overall top-ranked model using the bootstrap method. The column estimate contains the values in the model obtained using ProBMoT. The column Mean contains the mean of the outliers-filtered sample. The C^{lo} and C^{hi} columns contain the lower and upper bound of the 95% confidence interval. The C^L and C^{sh} columns contain the length and the shape of the confidence interval, where $C^L = C^{lo} - C^{hi}$ and $C^{sh} = \frac{C^{hi} - Mean}{Mean - C^{lo}}$. Assuming a confidence interval for a normal distribution, $C^{sh} = 1$ and the interval is symmetric about the mean, the column outliers contains the number of outliers from a sample of 1000.

Parameter	Estimate	Mean	C^{lo}	C^{hi}	C^L	C^{sh}	Outliers
r5(0)	2	1.892	1.555	2	0.445	0.318	143
R5(0)	1.603	1.511	1.308	1.747	0.438	1.164	191
K1	0.098	0.132	0.111	0.155	0.044	1.132	209
k1	0.119	0.139	0.122	0.158	0.036	1.094	163
r7(0)	0.097	0.187	0	0.485	0.485	1.590	27
R7(0)	0.305	0.107	0	0.310	0.310	1.888	5
K2	0.141	0.223	0.001	0.623	0.622	1.800	141
k2	0.093	1.132	0.001	3.772	3.771	2.333	0
td	31.744	104.918	5	195	190	0.901	0
ke,5	0.001	0.001	0.001	0.001	0.000	11.104	192
kf,5	0.744	2.750	0.001	4	3.999	0.454	0
kg,5	3.317	3.320	0.914	4	3.085	0.282	79
ke,7	2.488	1.888	0.001	4	3.999	1.118	0
kf,7	0.001	1.174	0.001	4	3.999	2.406	0
kg,7	0.675	1.868	0.001	4	3.999	1.141	0
kh,5	0.013	0.013	0.013	0.014	0.001	1.193	170
kH,7	3.144	0.710	0.001	2.071	2.070	1.919	19
ky,7	4	3.714	2.830	4	1.169	0.323	99
kh,7	2.946	0.497	0.001	1.218	1.217	1.450	3
kE,7	3.890	3.271	1.754	4	2.245	0.479	82
kF,7	3.821	2.007	0.001	4	3.999	0.992	0
kG,7	0.142	0.742	0.001	2.988	2.987	3.031	94
K	7946.916	8304.990	7287.316	9371.584	2084.267	1.048	107

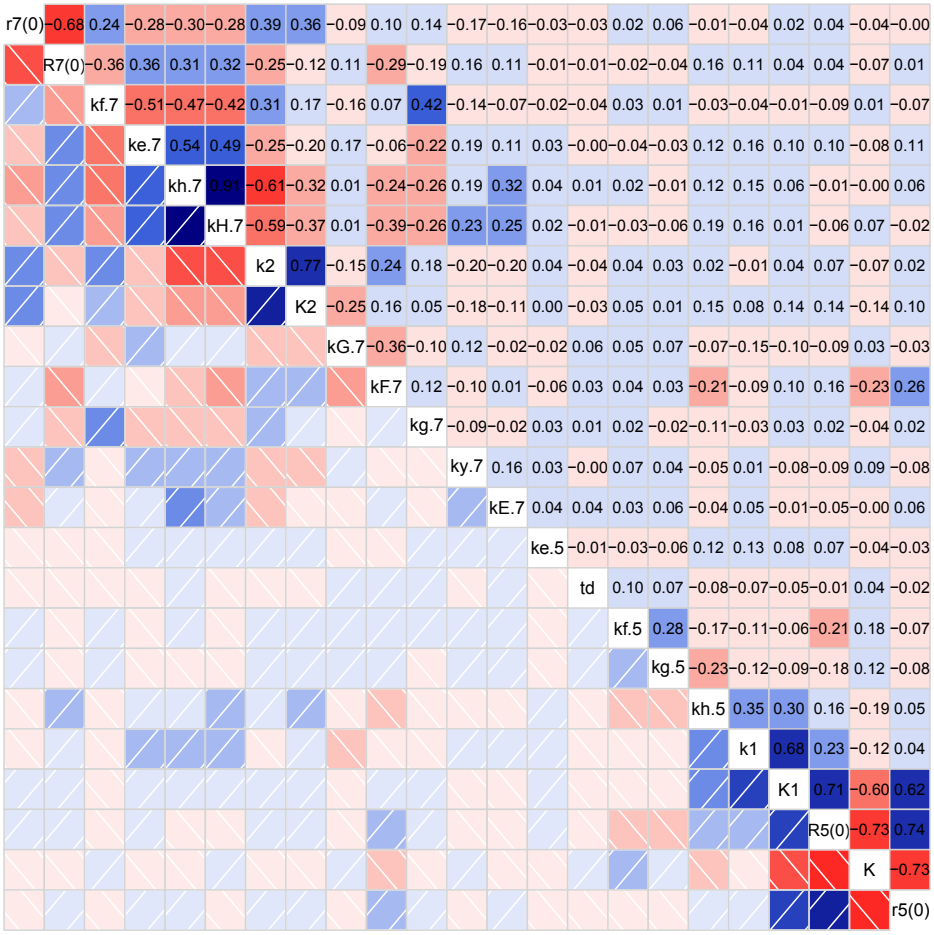

Figure S7: Correlation matrix for the parameters of the overall top-ranked model obtained using the bootstrap method. The names of the parameters for each row/column are shown in the main diagonal. The matrix is symmetric. Above the main diagonal, the values of the correlations correspond to the Pearson correlation coefficient.

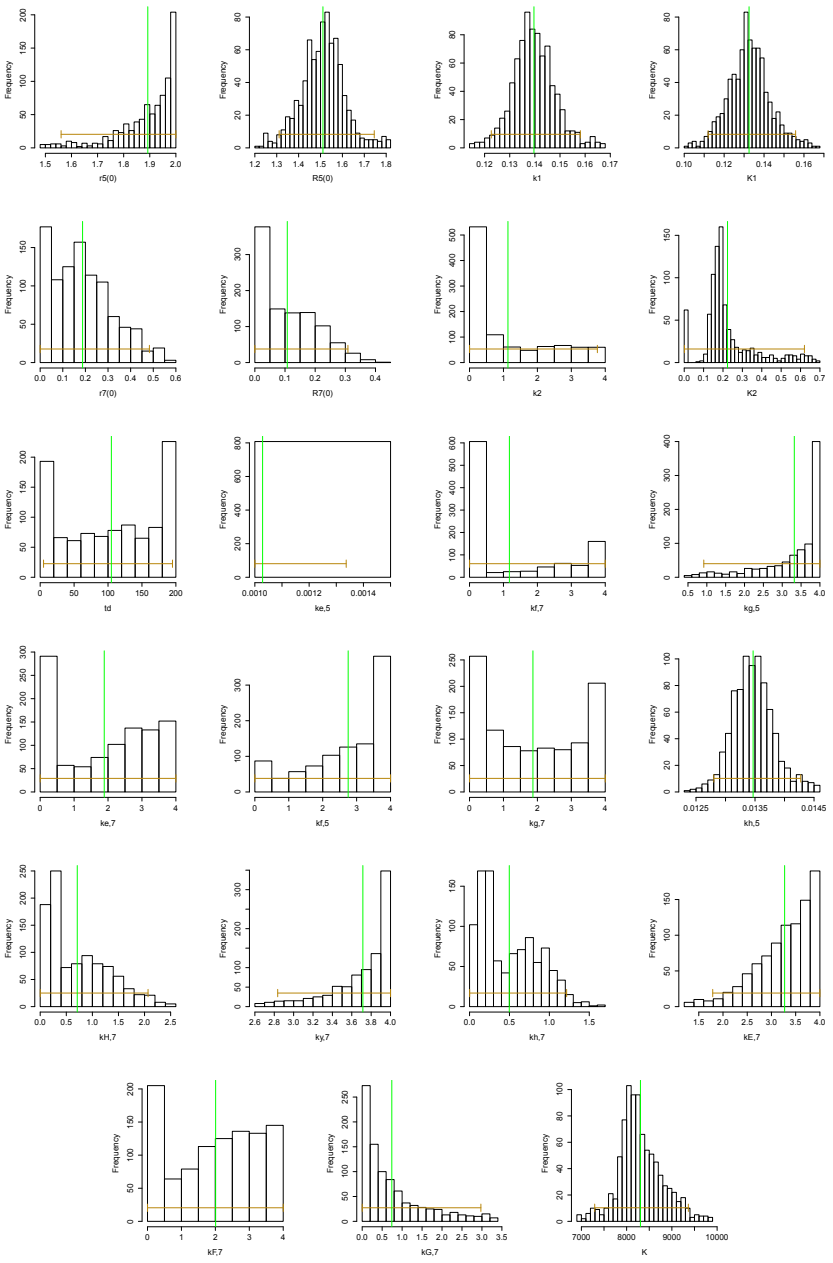

Figure S8: Histograms of the estimated parameter values of the overall top-ranked model obtained using the bootstrap method. In each histogram, the yellow horizontal line represents the 95% confidence interval and the green vertical line represents the mean of the outliers-filtered sample. The width of the bins is calculated according to the Freedman-Diaconis rule.

Top ranked COT model:

$$\begin{aligned}
\frac{dr_5}{dt} &= K_1 - (k_1 + \frac{k_{e,5}}{1 + e^{(k_{g,5} - R_5)\cdot k_{f,5}}}) \cdot r_5 + k_{h,5} + \frac{k_{H,5} \cdot R_7}{k_{y,5} + R_7} \cdot R_5 \\
\frac{dR_5}{dt} &= \frac{k_{e,5}}{1 + e^{(k_{g,5} - R_5)\cdot k_{f,5}}} \cdot r_5 - k_{h,5} + \frac{k_{H,5} \cdot R_7}{k_{y,5} + R_7} \cdot R_5 \\
\frac{dr_7}{dt} &= K_2 - (k_2 + \frac{k_{e,7} \cdot R_7}{k_{g,5} + R_7} + \frac{k_{E,7} \cdot R_5}{k_{G,7} + R_5}) \cdot r_7 + k_{h,7} + \frac{k_{H,7} \cdot R_5}{k_{y,7} + R_5} \cdot R_7 \\
\frac{dR_7}{dt} &= \frac{k_{e,7} \cdot R_7}{k_{g,5} + R_7} + \frac{k_{E,7} \cdot R_5}{k_{G,7} + R_5} \cdot r_7 - k_{h,7} + \frac{k_{H,7} \cdot R_5}{k_{y,7} + R_5} \cdot R_7
\end{aligned} \tag{2}$$

Table S4: Summary of the identifiability analysis for the estimated parameter values in the top-ranked model from the COT group using the bootstrap method. The column estimate contains the values in the model obtained using ProBMoT. The column Mean contains the mean of the outliers-filtered sample. The C^{lo} and C^{hi} columns contain the lower and upper bound of the 95% confidence interval. The C^L and C^{sh} columns contain the length and the shape of the confidence interval, where $C^L = C^{lo} - C^{hi}$ and $C^{sh} = \frac{C^{hi} - Mean}{Mean - C^{lo}}$. Assuming a confidence interval for a normal distribution, $C^{sh} = 1$ and the interval is symmetric about the mean, the column outliers contains the number of outliers from a sample of 1000.

Parameter	Estimate	Mean	C^{lo}	C^{hi}	C^L	C^{sh}	Outliers
r5(0)	1.201	1.491	0.085	2	1.914	0.361	10
R5(0)	1.173	1.217	0	2	2	0.643	0
K1	0.073	0.097	0.001	0.177	0.176	0.842	102
k1	0.156	0.138	0.099	0.180	0.081	1.088	322
r7(0)	0.100	0.236	0	0.800	0.800	2.391	72
R7(0)	0.132	0.089	0	0.345	0.345	2.884	51
K2	0.160	0.238	0.001	0.721	0.720	2.040	75
k2	0.222	1.199	0.001	3.985	3.984	2.324	0
td	20.169	87.954	5	195	190	1.290	0
ke,5	0.010	0.002	0.001	0.009	0.008	9.104	244
kf,5	0.714	2.481	0.001	4	3.999	0.612	0
kg,5	0.351	2.650	0.001	4	3.999	0.509	0
ke,7	0.096	1.233	0.001	4	3.999	2.245	0
kg,7	3.659	1.192	0.001	4	3.999	2.355	0
kH,5	0.002	0.005	0.001	0.015	0.014	2.128	180
ky,5	0.413	0.959	0.001	4	3.999	3.171	0
kH,7	1.695	0.579	0.001	2.212	2.211	2.825	78
ky,7	3.754	2.576	0.001	4	3.999	0.552	0
kh,7	0.739	0.215	0.001	0.840	0.839	2.916	36
kE,7	3.822	1.508	0.001	3.983	3.982	1.642	0
kG,7	1.229	2.021	0.001	4	3.999	0.979	0
kh,5	0.013	0.008	0.001	0.016	0.015	1.097	218
K	12330.653	9005.032	7121.954	15383.476	8261.522	3.387	200

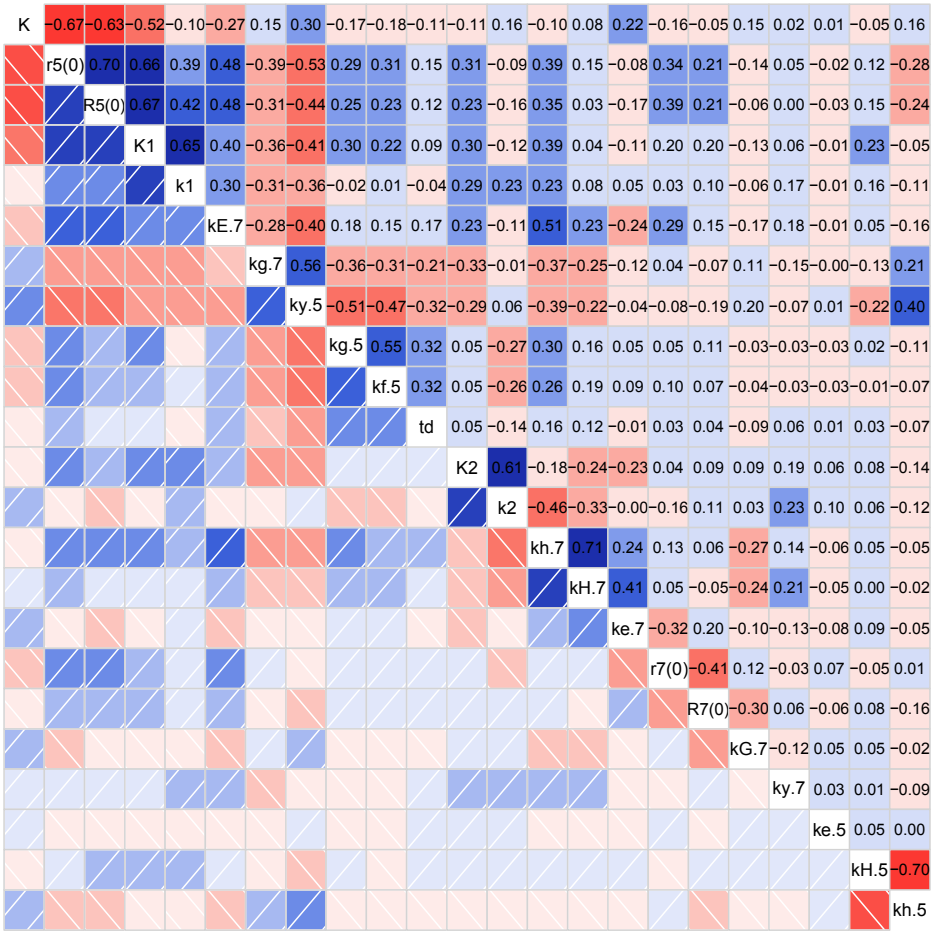

Figure S9: Correlation matrix for the parameters of the top-ranked model from the COT group obtained using the bootstrap method. The names of the parameters for each row/column are shown in the main diagonal. The matrix is symmetric. Above the main diagonal, the values of the correlations correspond to the Pearson correlation coefficient.

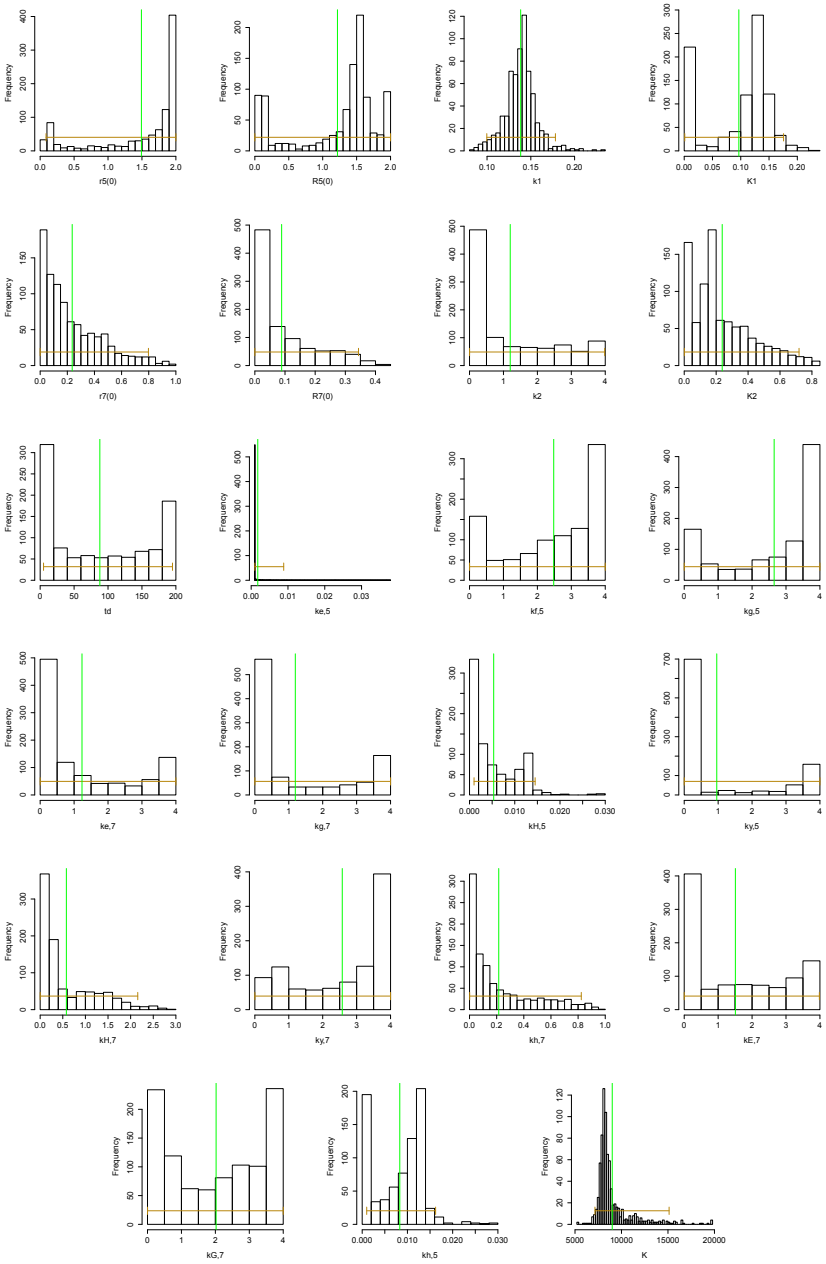

Figure S10: Histograms of the estimated parameter values of the top-ranked model from the COT group obtained using the bootstrap method. In each histogram, the yellow horizontal line represents the 95% confidence interval and the green vertical line represents the mean of the outliers-filtered sample. The width of the bins is calculated according to the Freedman-Diaconis rule.

Top ranked IP model:

$$\begin{aligned}
\frac{dr_5}{dt} &= K_1 - (k_1 + \frac{k_{e,5}}{1 + e^{(k_{g,5} - R_5) \cdot k_{f,5}}}) \cdot r_5 + k_{h,5} + \frac{k_{H,5} \cdot R_7}{k_{y,5} + R_7} \cdot R_5 \\
\frac{dR_5}{dt} &= \frac{k_{e,5}}{1 + e^{(k_{g,5} - R_5) \cdot k_{f,5}}} \cdot r_5 - k_{h,5} + \frac{k_{H,5} \cdot R_7}{k_{y,5} + R_7} \cdot R_5 \\
\frac{dr_7}{dt} &= K_2 - (k_2 + \frac{k_{e,7} \cdot R_7}{k_{g,5} + R_7} + \frac{k_{E,7} \cdot R_5}{k_{G,7} + R_5}) \cdot r_7 + k_{h,7} \cdot R_7 \\
\frac{dR_7}{dt} &= \frac{k_{e,7} \cdot R_7}{k_{g,5} + R_7} + \frac{k_{E,7} \cdot R_5}{k_{G,7} + R_5} \cdot r_7 - k_{h,7} \cdot R_7
\end{aligned} \tag{3}$$

Table S5: Summary of the identifiability analysis for the estimated parameter values in the top-ranked model from the IP group using the bootstrap method. The column estimate contains the values in the model obtained using ProBMoT. The column Mean contains the mean of the outliers-filtered sample. The C^{lo} and C^{hi} columns contain the lower and upper bound of the 95% confidence interval. The C^L and C^{sh} columns contain the length and the shape of the confidence interval, where $C^L = C^{lo} - C^{hi}$ and $C^{sh} = \frac{C^{hi} - Mean}{Mean - C^{lo}}$. Assuming a confidence interval for a normal distribution, $C^{sh} = 1$ and the interval is symmetric about the mean, the column outliers contains the number of outliers from a sample of 1000.

Parameter	Estimate	Mean	C^{lo}	C^{hi}	C^L	C^{sh}	Outliers
r5(0)	2	0.875	0	2	2	1.283	0
R5(0)	1.553	0.927	0	2	2	1.156	0
K1	0.092	0.088	0.001	0.345	0.344	2.950	192
k1	0.117	1.718	0.013	4	3.987	1.337	0
r7(0)	0.002	0.110	0	0.504	0.504	3.551	113
R7(0)	0.580	0.123	0	0.537	0.537	3.341	48
K2	0.114	0.349	0.001	1.520	1.519	3.356	53
k2	0.075	1.565	0.001	3.999	3.998	1.556	0
td	5	74.272	5	195	190	1.742	0
ke,5	0.001	1.019	0.001	4	3.999	2.924	0
kf,5	1.356	1.871	0.001	4	3.999	1.138	0
kg,5	3.399	2.125	0.001	4	3.999	0.882	0
ke,7	0.754	0.579	0.001	2.322	2.321	3.010	69
kg,7	0.001	2.937	0.001	4	3.999	0.362	0
kH,5	0.002	0.005	0.001	0.019	0.018	3.509	221
ky,5	4	1.501	0.001	4	3.999	1.664	0
kh,7	1.259	0.008	0.001	0.046	0.045	5.798	166
kE,7	3.422	0.025	0.001	0.278	0.277	10.573	222
kG,7	3.915	2.190	0.001	4	3.999	0.826	0
kh,5	0.009	0.015	0.001	0.048	0.047	2.478	186
K	8118.054	22189.124	7476.843	81297.686	73820.843	4.017	124

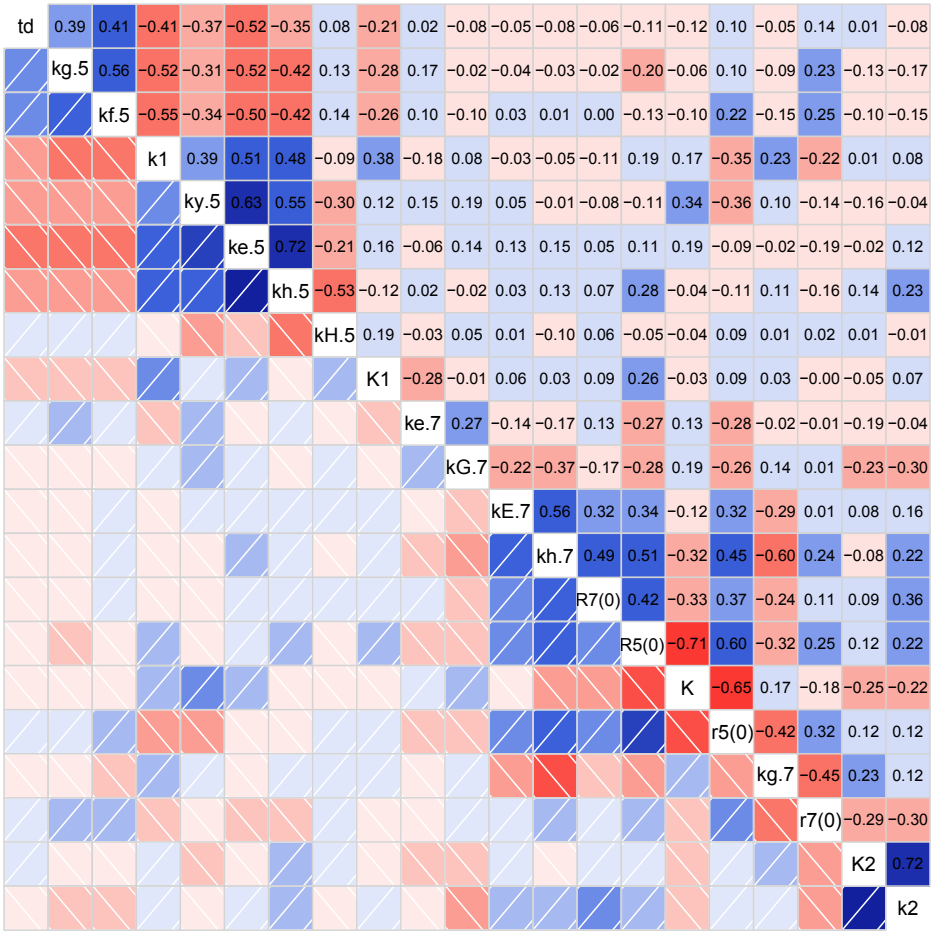

Figure S11: Correlation matrix for the parameters of the top-ranked model from the IP group obtained using the bootstrap method. The names of the parameters for each row/column are shown in the main diagonal. The matrix is symmetric. Above the main diagonal, the values of the correlations correspond to the Pearson correlation coefficient.

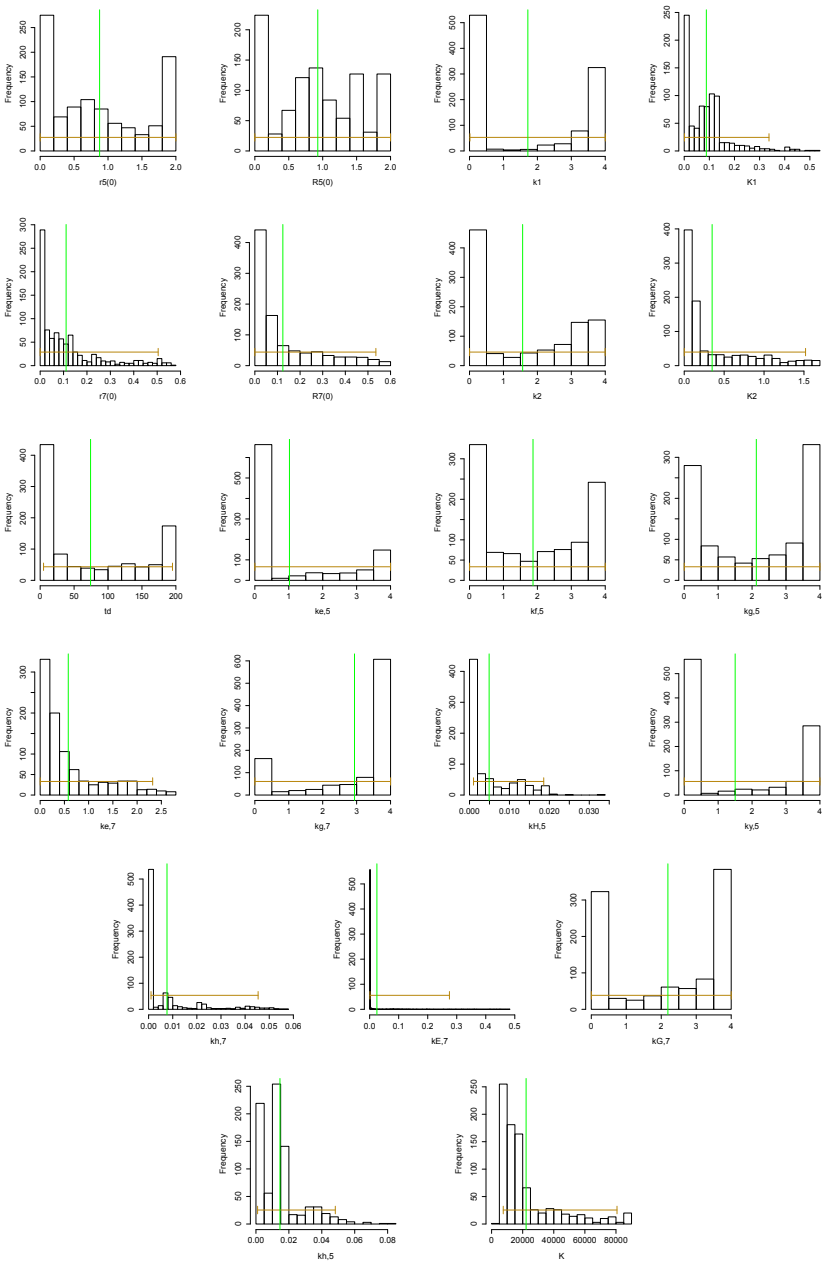

Figure S12: Histograms of the estimated parameter values of the top-ranked model from the IP group obtained using the bootstrap method. In each histogram, the yellow horizontal line represents the 95% confidence interval and the green vertical line represents the mean of the outliers-filtered sample. The width of the bins is calculated according to the Freedman-Diaconis rule.

Chapter 5

Process-Based Design of Dynamical Biological Systems

5.1 Problem Description

The newly developed ability to apply the PBM approach to problems of modeling both deterministic and stochastic dynamical biological systems, combined with the possibility of model selection (by using the bias strengthening approach) in scenarios with extremely limited availability of data, motivated the development of a process-based approach to the design of biological systems for synthetic biology applications.

In order to develop methods for process-based design of dynamical biological systems, we need to first consider the specifics of the task of design that need to be addressed. The issue of formally representing the components needed for the composition of novel systems is explicitly addressed by the process-based formalism. The knowledge of the different available components, their properties, and interactions can be flexibly encoded within the library of domain knowledge. The space of candidate design models can be defined by imposing constraints that describe feasible compositions of components with an incomplete process-based model.

The main issue preventing the direct use of the PBM approach to model inference for the task of design is that PBM is, for the most part, data-driven. Due to the nature of the task of modeling, the optimization of the parameter values of each candidate model structure is based on the minimization of the discrepancy between observed system behavior and simulated model behavior. For the task of design, apart from the possibility of considering different exogenous influences, such as stimuli and experimental or environmental conditions, no observations can be given at input.

We have already relaxed this strict form of optimization by adding to the objective function heuristic terms related to the expected behavior/properties of the systems. We have shown that these can guide parameter estimation towards a more discriminative and correct model selection. In the extreme case, due to unavailability of observed data for comparison, this approach can be applied to the task of process-based design by dropping the first term of Equation 4.3 and considering the following form of the heuristic score of a model:

$$E = \mathcal{D}(x, \hat{x}|\mathcal{M}_{\mathcal{I}}, \theta) + \lambda \cdot \mathcal{R}(\theta'_{\mathcal{M}_{\mathcal{I}}}) \tag{5.1}$$

When solving the task of design if the objective function is based on a single specific desired property of the candidate models, the parameter inference, as in the case of less informative observations and an objective function based on the discrepancy between ob-

servations and model simulation, may result in overfitting or in a hard model selection problem.

Therefore, the heuristic used for design is usually based on the optimization of multiple, possibly conflicting objectives. There are two general approaches to solving a multi-objective optimization problem. The first approach is the transformation of the multi-objective problem into a single objective problem by scalarization. The most common scalarization is based on a variation of the method of using a weighted sum of the individual objectives, i.e., $\mathcal{D}(x, \hat{x}|\mathcal{M}_{\mathcal{I}}, \theta) = \sum_{j=1}^{d} w_d \cdot O_d(x, \hat{x}|\mathcal{M}_{\mathcal{I}}, \theta)$, where the weights w_d correspond to the importance of the objective functions O_d. Another method of scalarization is the selection of a single objective for optimization, while defining acceptable upper bounds on the others to serve as constraints to the optimization problem. The scalarization of multiple objectives and the introduction of a subjective preference for one individual objective causes information loss, especially in cases where conflicting objectives guide the optimization and the Pareto front is non-convex.

The second approach is based on the simultaneous optimization of the objective functions and the approximation of the entire Pareto front. For the task of design by optimization, a single Pareto front for the entire space of candidate designs can be obtained as a final output (Higuera et al., 2012; Otero-Muras & Banga, 2014). The final choice of a single model structure with specific parameter values is left to the modeler.

However, while considering all candidate structures at the same time, the information about the potential range of behaviors of each candidate structure for different parameter values is discarded, which may result in a problem of selection among overfitted models. Namely, the multi-objective optimization of an individual candidate design results in a Pareto front of sets of parameter values for the specific design structure which reflects the ability of the candidate structure to optimally achieve the desired properties. The overall Pareto front obtained by design as optimization methods can be considered as an aggregation of Pareto fronts for individual model structures by using the dominance function.

Better model selection can be performed by making use of the complete information available from the individual structures' Pareto fronts to quantify the quality of each candidate design by using the quality indicator method. Zitzler, Knowles, and Thiele (2008) review different quality indicators for comparing Pareto fronts. For the process-based design of dynamical biological systems, we argue that a model selection score based on Equation 5.1, can be obtained by replacing $\mathcal{D}(x, \hat{x}|\mathcal{M}_{\mathcal{I}}, \theta)$ with a quality indicator based on the hyper-volume under the Pareto front (Lu & Anderson-Cook, 2013).

We demonstrate the adequacy of the process-based design method for both formulating design tasks (by specifying the candidate designs and design objectives), and solving them (by employing the hyper-volume measure as a design-selection strategy) by approaching two design tasks involving a stochastic toggle switch without cooperativity and a deterministic oscillator (Tanevski et al., 2016b).

This work, submitted for a review for publication as a journal article, constitutes the remainder of this chapter. The full bibliographic reference to the article is:

Tanevski, J., Todorovski, L., Džeroski, S. (2016b). Process-based design of dynamical biological systems. *Scientific Reports*, Under review.

5.2 Related Publication

Process-based design of dynamical biological systems

Jovan Tanevski[1,2,*], Ljupčo Todorovski[3], and Sašo Džeroski[1,2]

[1]Jožef Stefan Institute, Ljubljana, Slovenia
[2]Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
[3]University of Ljubljana, Slovenia
[*]jovan.tanevski@ijs.si

ABSTRACT

The computational design of dynamical systems is an important emerging task in synthetic biology. Given desired properties of the behaviour of a dynamical system, the task of design is to build an in-silico model of a system whose simulated behaviour meets these properties. We introduce a new, process-based, design methodology for addressing this task. The new methodology combines a flexible process-based formalism for specifying the space of candidate designs with multi-objective optimization approaches for selecting the most appropriate among these candidates. We demonstrate that the methodology is general enough to both formulate and solve tasks of designing deterministic and stochastic systems, successfully reproducing plausible designs reported in previous studies and proposing new designs that meet the design criteria, but have not been previously considered.

Introduction

Systems-based approaches to biology lead to better understanding of interactions in biological systems represented at different organizational levels. They rely on formalizing a model of a given system by specifying its constituents at a chosen organizational level, its structure, i.e. the interactions between the constituents, and the particular modelling assumptions for each interaction. The model is often used as a tool for analysis of the complex dynamical behaviour of the system over time and under changing internal and external conditions. Using models as analytical tools, we can obtain insights into the essential mechanisms that lead to emergence of complex dynamics in biological systems. In turn, these insights can be employed when solving the task of design, i.e. when constructing models of dynamical systems that exhibit a desired behaviour.

The most important input to the task of design is the knowledge about modelling dynamics in the domain of interest. This knowledge includes the systematic categorization of constituents of dynamical systems in the domain and the potential interactions between them. The second input to the design task are the design objectives, i.e. a set of expected properties of the desired dynamical behaviour of the system. The output of the design process is a candidate design (or a set thereof), i.e. a model with known structure and parameter values. To solve the design task specified above, we need to resolve two types of uncertainties. The first type of uncertainty is related to the model structure: The more different model structures we need to consider, the larger the uncertainty. To resolve the structural uncertainty, we need to select a model structure, i.e. to select a proper set of model constituents, the interactions among them, and make specific modelling assumptions on the kinetics of the interactions. The second type of uncertainty is related to the values of the constant parameters in the models, such as kinetic rates and initial conditions. We call this type of uncertainty parametric uncertainty. Resolving these two types of uncertainties leads to a candidate design (or a set thereof) that produces dynamical behaviour with expected properties.

In response to the increasing relevance of the task of designing biological systems for practical applications,[1,2] numerous computational approaches addressing the design task have been proposed. The approaches differ in the way the inputs and the uncertainties mentioned above are formalized and resolved. In particular, two classes of approaches are related to the work presented in this paper. The first class includes approaches that follow the "design by composition" paradigm, where valid compositions of standardized components with known types of interactions are sought for, given a design objective describing the relationship between designated inputs and outputs. The approaches in the second class follow the "design by optimization" paradigm, where the design objectives are transformed into objective functions that are then subject to optimization. This allows for considering a broader class of design objectives related to the qualitative and quantitative properties of the desired dynamical behaviour of the system. On the other hand, these approaches employ a rigid formalism for specifying the structural uncertainty that requires users to provide an explicit and complete equation-based specification of the structure of each candidate design.

Composition-oriented approaches are built upon the concepts of rule-based modelling and computer-aided design of

electronic circuits. Rule-based modelling formalisms (and the related approaches to automated composition of parts), such as GEC[3] and Eugene,[4] are used to describe constraints for the composition of a single design. These constraints are an addition to the domain knowledge specified in the form of a library of circuit components, based on standardised, well-characterized biological parts. Circuit components have fixed properties and rules define fixed parameter values for interactions among components. Thus, all parameters in a valid circuit have fixed values, which eliminates the parametric uncertainty. The structural uncertainty in these formalisms comes from the availability of interchangeable parts for the desired physical composition. However, these formalisms do not support automated resolution of the structural uncertainty. Instead, experts can use them to manually generate and test different valid compositions of biological circuits to achieve a given design objective. Composition-oriented approaches to automated resolution of structural uncertainty based on a design objective, such as Proto[5] and Cello[6] have been recently developed. The design objective in these approaches is formulated as a Boolean function of the defined inputs and outputs of the circuit. These approaches infer an abstract network representation of a composition of standard parts. The inference method is additionally constrained by the intended physical implementation of the circuit, i.e. the components that can be used to implement the intermediary logic functions (logic gates) needed to achieve the design objective. The final compositions produced by Cello and Proto can be resolved into a specific physical construct, i.e. DNA sequence of the composition using Eugene or more advanced methods, such as MatchMaker.[7] In any case, the application of these methods is limited to the specific task of designing biological circuits that realize a given transition function based on a single form of interactions between the connected components (represented by a sigmoidal transfer function).

In contrast, the approaches in the second class can handle more general design objectives, such as, for example, constrained values of the components of the Fourier spectrum of the system trajectory that are indicators of oscillatory system behaviour. However, they provide only simple tools for formalizing structural and parametric uncertainty by letting the user specify a list of candidate designs in the form of model equations with unknown parameter values. These approaches transform structural uncertainty into parametric uncertainty. ABC-SysBio[8] requires the user to specify a list of equation-based models, each corresponding to a candidate design. It further reformulates structural uncertainty as parametric uncertainty by introducing an integer parameter whose value corresponds to the index of the candidate design in the list. Bayesian estimation methods are then employed to resolve the parametric uncertainty, producing a posterior distribution over the candidate model structures and the values of their parameters, which can be used for selecting designs that provide optimal fit to the design objectives. Other optimization-oriented approaches[9–13] require the user to specify equation fragments and integrate them within a single equation structure using Boolean parameters that indicate the presence of individual fragments in the structure. The structural uncertainty is thus transformed into parametric uncertainty that is then resolved by using parameter estimation. In both cases, the formalization of domain knowledge and uncertainty is rigid and requires the users to directly encode complete mathematical models of design candidates. Some of these approaches transform multiple design objectives into a single objective function for optimization,[9,10,12] while others[8,11,13] use multi-objective optimization methods that can consider multiple objectives simultaneously.

The central contribution of our work is a new approach to automated design that combines the flexibility of the design by composition approaches with the generality of the design by optimization paradigm. Our approach allows for flexible formalization of both structural and parametric uncertainty through a library of domain knowledge that specifies a taxonomy of design constituents and processes that describe their interaction composition. At the same time, the approach is able to handle a broad class of design objectives. In order to resolve the structural and parametric uncertainty, we bring together methods for combinatorial search and multi-objective optimization. The search space of candidate model structures is inferred from the specification of constituents and the potential interactions among them. The estimation of parameter values for each model structure employs a method for simultaneous optimization of multiple design objectives. The solution in each resulting Pareto front are aggregated by using a hypervolume based metric,[14] that is in turn used for the ranking and selection of candidate designs. Our approach builds upon the paradigm of process-based modelling[15,16] that integrates formalized domain-specific knowledge and observed/measured data for automated modelling of dynamical systems. In the new setting, the design objectives, representing the desired properties of the behaviour of the system replace the objective of fitting the observed data. We adapt the process-based modelling paradigm to the design task where no measured/observed data are available. We conjecture that the newly proposed approach is capable of reconstructing the results of previous design efforts. In addition, we conjecture that the approach, due to the more flexible formalism for specifying domain knowledge and uncertainties, is also capable of discovering new promising designs not considered before. To test the validity of the two hypotheses, we apply the newly developed approach on two tasks with archetypical design objectives, i.e. designing a toggle switch and an oscillator. The first is based on a stochastic model of a genetic switch without cooperation that can be used as a basic memory unit.[17] The second is a deterministic oscillator based on a negative-feedback loop of protein interaction.[18]

From process-based modelling to process-based design

Process-based modelling[15] is an automated modelling paradigm that takes two inputs: formalized knowledge about modelling dynamical systems in the domain of interest and measurements of the observed system that is subject to modelling. The output is a set of models ranked according to how well they correspond to the observed system (i.e. fit the measured data about the system). The domain-specific knowledge about modelling is formalized as a library of entities (that represent constituents of the dynamical systems in the domain) and processes (that correspond to interactions between the entities). The knowledge library, when instantiated for a particular set of entities observed in a dynamical system at hand, provides a set of components for building models of the observed system. Process-based modelling approaches make use of combinatorial search to explore the space of candidate model structures that can be built from these components. The values of the parameters in these structures are estimated by using optimization methods that minimize the discrepancy between the measured system behaviour and the behaviour obtained by simulating the model. Following the search and optimization, process-based modelling approaches provide at output a list of models ranked with respect to their likelihood of reconstructing the observed system behaviour.

Process-based modelling has been successfully applied to a variety of modelling tasks in biology[16,19,20] and other domains.[21] It has several advantages over other modelling paradigms that make it particularly suitable for adaptation to the task of design. First, process-based models retain the *understandability* and explanatory power of graphical model representations by providing clear insight into the structure of the observed system in terms of its constituents (entities) and interactions (processes) among them. Second, at the same time, they inherit the *utility* of mathematical models for simulation and analysis of system behaviour. Third, process-based models provide *general* model descriptions that support both stochastic and deterministic approaches to modelling, simulation and analysis. Finally, the knowledge representation formalism facilitates *modularity*: the knowledge library can be instantiated into a number of model components that are tailored to a particular modelling task at hand.

The most important and distinguishing aspect of process-based modelling is its ability to formally describe two different kinds of modelling uncertainty: uncertainty in the model structure and uncertainty in the model parameters. The structural uncertainty is captured in the formal description of domain knowledge (entities and processes) and is made explicit by transforming the latter into a space of candidate model structures (as opposed to the case of considering a single structure where we have no structural uncertainty). Process-based modelling approaches then employ combinatorial search methods to resolve the structural uncertainty. The very same methods can be used for process-based design. On the other hand, the parametric uncertainty is described by the formal specification of ranges of values for the model parameters. Given a model structure and measurements of the system behaviour, the values of these parameters are estimated by using standard optimization methods. The optimization objective functions (criteria) and the score for ranking the process-based models are derived from the following components:

C_1 Measured behaviour of the observed dynamical system.

C_2 Model behaviour obtained by simulation.

C_3 Model complexity, in terms of model entities, processes and parameters.

The basic, most commonly used objective function, stems from the maximum likelihood principle, and uses C_1 and C_2. It measures the discrepancy between the measured system behaviour (x_i) and the simulated model behaviour ($\widehat{x_i}$),

$$\mathrm{RMSE}(m) = \frac{1}{\sqrt{N}} \sum_i \|x_i - \widehat{x_i}\|, \tag{1}$$

where i iterates over the observed system variables and N denotes the number of measurement time points. Another commonly used objective function relies on the parsimony principle that takes into account model complexity C_3. If the complexity of the model is implicitly encoded by the values of the constant parameters, the objective function shown in equation (1) can be regularized by adding a component that takes into account the magnitude of the model parameters. A more general objective function can be obtained by following the minimum description length (MDL) principle,[22]

$$\mathrm{MDL}(m) = \mathrm{L}(m) + \mathrm{L}(D|m),$$

that takes into account $\mathrm{L}(m)$, the length of the minimal code necessary to completely encode the model (based on C_3), and $\mathrm{L}(D|m)$, the length of the code describing the discrepancy between the simulated (C_2) and measured behaviour (C_1). The criteria based on the parsimony principle are useful when there is a need to distinguish between the suitability of multiple competing models with different structures.

For the design task, C_1 cannot be used as a component of the objective function, since no measured data is available at input. For the design task, a second input is available that can replace the measured data:

$\mathbf{C}_1'$ Expected properties of the desired model behaviour.

Following this change, the RMSE criterion (equation (1)) is replaced with one that combines $C_1\prime$ and C_2. A suitable design might have to fulfil multiple design objectives (expected properties), which (in general) can be independent or even conflicting. The discrepancy between each expected property of the system behaviour and the same property of the model simulation can be observed and used in the new criterion. Examples of multiple expected independent properties used for designing a system with oscillatory behaviour include the oscillation frequency and amplitude.

The issue of satisfying/optimizing multiple objectives can be addressed either by aggregating the corresponding objectives (i.e. the discrepancies between the expected and the actual value of the property of the behaviour) and using a single-objective optimization method or by simultaneous optimization of the objectives using a multi-objective optimization method. The aggregation of multiple objectives requires the introduction of a subjective weighting of the individual objectives in the aggregation. The subjective weighting can cause loss of information, especially in cases where independent or conflicting objectives guide the optimization. This makes a strong case for using multi-objective optimization methods to simultaneously handle multiple design objectives.

Finally, note that we do not use C_3 as a component of the objective function for parameter estimation, since the complexity of the model structures in process-based modelling/design remains constant throughout the optimization of its parameter values. However, we take into account the component C_3 in the final ranking of the models, when models with varying structural complexity are considered.

Process-based design

We developed (designed and implemented) ProBMoTd, a tool for process-based design, as an extension of the process-based modelling tool ProBMoTs.[16] The extension proceeded in the following directions. First, we made use of our recent upgrade to the formalism for building process-based models from differential equations to reaction equations (also referred to as reaction networks), a formalism commonly used for stochastic and deterministic modelling in systems and synthetic biology.[23] Second, following the discussion from the previous section, we employed multi-objective (instead of single-objective) optimization methods for solving the parameter estimation task. Finally, we introduced a new model-selection score for ranking the candidate models that takes into account the design objectives and the complexity of the model structure.

Following the process-based modelling paradigm, the task of process-based design takes as input a library of knowledge about system constituents and interactions in the domain of interest, encoded by using template entities and processes. An entity represents a constituent of an observed system with its constant and variable properties. For instance, in a simple model of a system that involves protein binding, an entity corresponds to a protein with a single variable property `mol` denoting the number of its molecules present in the system. A formal description of a template entity corresponding to a protein is given at the top of Table 1: this template can be instantiated to multiple different proteins that need to be considered for a specific dynamical system.

Similarly, in this example, processes describe the binding interaction between protein entities. The first template process in Table 1 corresponds to an abstract binding interaction, while the following two template processes represent two more specific types of binding: irreversible and reversible binding. The template process `binding` specifies that a binding interaction involves three constituent entities, all of the same type — `protein`: `p1` and `p2` denote the binding proteins and `pc` denotes the protein complex resulting from the binding process. The two subordinate processes `irreversible_binding` and `reversible_binding` inherit the involved entity attributes (`p1, p2, pc`) as well as the constant parameter corresponding to the binding rate (`k1`) from their parent (the template process `binding`). Each of the subordinate binding processes specifies the final template reaction equations used to model the binding interactions; for a particular system being modelled, only one of these two alternatives with specific values of the constant parameters will apply.

Given a specific system with three proteins `A`, `B` and `AB`, an incomplete model can be specified that contains a single process instance `binding(A, B, AB)`. Note that, by this specific instantiation, the incomplete model formalizes the structural uncertainty: it defines a space of two candidate model structures, one containing a process of irreversible and the other a process of reversible binding. While each of these structures contains a different form of the binding process the values of the constant parameters `k1` and/or `k2` remain to be estimated in both; in other words, besides structural, we also have parametric uncertainty. To resolve it, we employ parameter estimation, where the parameter values are optimized with respect to the design objectives, i.e. the expected properties of the desired system behaviour. An example objective can aim at a specific steady-state of the system. In particular, we observe the property of the behaviour of the system $\mathrm{S}(x)$ that corresponds to the number of molecules of x when the system reaches a steady state. A possible formulation of the design objective is that $O = (\mathrm{S}(\mathtt{A}) + \mathrm{S}(\mathtt{B}))/2 - \mathrm{S}(\mathtt{AB})$ comes as close to the target value of 0 as possible. The parameter estimation will find optimal values of the model parameters for each of the two candidate models. The optimal value of the objective would indicate the suitability of the model candidate. In this simple example, the reversible binding is expected to be a more suitable alternative for achieving the selected objective.

Table 1. Formal representation of modelling knowledge for protein binding.

```
template entity protein {
  vars: mol {range: <0,100>};
}
template process binding
        (p1: protein, p2:protein, pc: protein) {
  consts: k1 {range: <0.1,5>};
}
template process irreversible_binding: binding
  equations:
    p1.mol + p2.mol -> pc.mol [k1];
}
template process reversible_binding: binding
  consts: k2 {range: <0.1,5>},
  equations:
    p1.mol + p2.mol -> pc.mol [k1],
    pc.mol -> p1.mol + p2.mol [k2];
}
```

In general, however, design tasks include a number of objectives corresponding to different expected properties of the desired behaviour. In order to support multi-objective parameter estimation, we integrated within ProBMoTd the implementation of Generalized Differential Evolution[24] from the Java-based framework for multi-objective optimization.[25] In contrast to a single optimal point obtained in the case of a single objective, the result of multi-objective parameter estimation (for a given model structure) is a set of Pareto-optimal points from the parameter space (referred to as the Pareto front) together with the corresponding values for each objective. To rank the candidate model structures, we ranked the corresponding Pareto fronts. To this end, we calculated the hyper-volume under each of the Pareto fronts, HVUPF,[14] i.e. the volume between the set of points on the Pareto front and the origin point (that corresponds to the optimal values of the objectives).[26]

To use HVUPF as a score for ranking the candidates, several assumptions (that do not limit the applicability of the approach), should be met. First, each objective should have a finite domain of possible values that is known a-priori. Second, all the objectives should be formulated in a manner that requires their minimization. Under these assumptions, a candidate model structure with a smaller HVUPF outperforms the candidates with larger volumes. A simple design selection strategy is to choose the model with the smallest HVUPF. However, as in the case of modelling, these estimates can be biased towards more complex models. To address this issue, we introduced a selection score that penalizes complex model structures, where complexity is measured as the number of reaction equations. For finite spaces of candidate model structures, M, both HVUPF and the model structure complexity were normalized to the $[0,1]$ interval and combined in an MDL-like score for a single model structure m as follows:

$$\text{MDLscore}(m|M) = \alpha \text{HVUPF}(m|M) + (1-\alpha)\text{C}(m|M), \tag{2}$$

where α is a parameter in the interval $[0,1]$ used to trade-off between the HVUPF and the model structure complexity (C). At output, ProBMoTd reports the list of the candidate models ranked with respect to the descending value of the score. When reporting the results of the empirical evaluation, we visualized the score profile of the ProBMoTd output as a bar plot, where x- and y-axes correspond to the candidate model ranks and the model scores (on a logarithmic scale), respectively.

Results

Methodological contribution

Before reporting the results of the empirical evaluation of the proposed approach (in terms of its ability to reconstruct known results of previous design efforts and propose new designs), we summarize the methodological contribution of the paper and restate its position within the context of related approaches to automated design. In particular, we present the workflow used to formalize and resolve structural and parametric uncertainties with the process-based design approach.

Figure 1 recapitulates the workflow of the process-based design approach that was introduced in the previous section. Given the two inputs to the design task, the domain-specific modelling knowledge and the expected properties of the desired behaviour, an expert has to prepare the input to ProBMoTd. First, following the composition-oriented approach to design, the domain knowledge is encoded in a library in the form of a taxonomy of template entities and processes that can be used for modelling

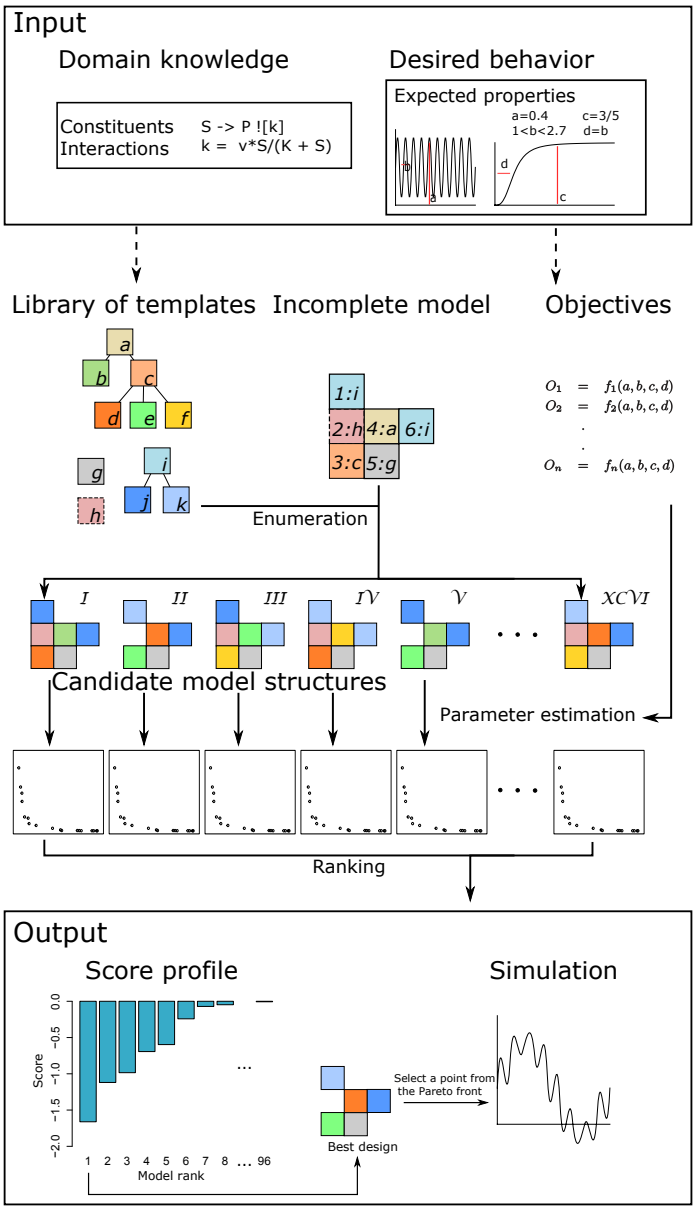

Figure 1. The workflow of our process-based design approach. At input, it takes domain-specific knowledge about modelling systems in the study domain and expected properties of the desired behaviour. The knowledge is formalized as a taxonomy of modelling templates and a specification of an incomplete model. The incomplete model uses inner nodes in the taxonomy to specify a set of alternative design choices; during the enumeration of the candidate model designs, these are instantiated with leaf nodes in the taxonomy that correspond to specific design choices. The parameters of each candidate model are estimated using multi-objective optimization with objectives corresponding to the expected properties of the desired system behaviour, yielding a Pareto front of solutions for each candidate. Finally, at output, the candidate designs (model structures) are ranked according to the hyper-volumes under their Pareto fronts obtained with multi-objective optimization. For each design, the output contains its structure, parameters and simulated behaviour.

any system in the domain at hand. Next, for the particular design task, constraints are specified on how entity and process templates from the library of domain knowledge can be instantiated and composed into candidate designs. This is done by the specification of an incomplete model, which formalizes the structural uncertainty. To resolve the structural uncertainty,

ProBMoTd enumerates the candidate model structures by combining the incomplete model specification with the library of entity and process templates. Components of the incomplete model that correspond to the inner nodes of the template taxonomy are instantiated with their subordinate leaves, leading to multiple candidate model structures. For example, position *1* in the incomplete model can be replaced by instances of two alternatives (*j* and *k*; see the incomplete model and library of templates visualization in Fig. 1). The instance of the alternative (*j*) appears in the candidate structures *I*, *III* and *V*. Components with dashed borders (such as component *h*) are optional. Position *2* in the incomplete model can thus either be instantiated using the template *h* (as in model structures *I*, *III*, *IV*) or omitted (as in model structures *II* and *V*).

The design objectives, i.e. the expected properties of the desired system behaviour, are reformulated as objective functions for optimization (following the optimization-oriented approaches to design). For each candidate model structure, a multi-objective optimization method is used to fit the parameter values (so as to minimize the objective functions) resulting in a Pareto front of sets of parameter estimates and their corresponding values for the objective functions. In turn, the HVUPF score is used to aggregate the values of the objective functions for the points on the Pareto front into a single score of the design that is used to rank the candidates (equation (2)). Experts can then analyse the ProBMoTd output, i.e. the score profile of the (top ranked) candidate designs and select some of them for further exploration. To obtain a simulation for a selected candidate design, the user has to select a single point from the Pareto front: the points on the Pareto front are by definition such that none of them is better/worse than all of the others on all design objectives. Note, however, that the decision making requirement for the user has been maximally postponed to the point where complete information is available about the best design structure and its possible parametrizations. This information characterizes the design's ability to achieve the desired behaviour and allows the user to make an informed choice. By default, it may be useful to select a point from the Pareto front positioned in a region in the objective space that is as close as possible to the origin point, for which the design can be considered to achieve satisfactory behaviour, but the user may select alternative points from the Pareto front based on their preferences.

Stochastic toggle switch without cooperativity

The synthetic toggle switch[27] is one of the first synthetically designed systems that can achieve bistable switch-like behaviour. The importance of a simple synthetic switch is its potential use as a basic memory unit able to hold one bit of information. Its simple design contains two genes coding for proteins that mutually inhibit their production. The system can be controlled by inducer molecules that change the steady state of the system from a state where one protein has a low number of molecules while the other has high to the opposite one. In our work, we approached the task of designing a toggle switch without cooperative binding. Lipshtat et al.[17] showed that the basic toggle switch without cooperativity might not always be able to achieve a switching behaviour, due to the possibility of a deadlock state where both the number of proteins A and B in the system is zero. They proposed different mechanisms to improve the design in order to achieve a more robust switch like behaviour. In a later study, Barnes et al.[8] considered the task of selecting a most suitable model among four candidates, which contain one of the proposed mechanisms by using a Bayesian approach. In both studies, the candidate model structures were explicitly and manually enumerated.

We next describe in detail the process of preparing the input to the task of process-based design, i.e. the library of templates and the incomplete model, which uses domain knowledge from the previous studies. The formal representations of the library of domain knowledge and the incomplete model are shown in Supplementary Table S1 and Supplementary Table S2, respectively. The system is composed of constituents that we described using five template entities: `gene`, `protein`, `bound_factor` (describing a protein bound to the promoter region of a gene), `inducer` (describing an external inducer controlling the number of molecules of a specific protein in the system) and `complex` (describing a complex formed by binding of an inducer to a protein). Each template entity has a single variable property `mol`, which represents the number of molecules of the corresponding entity present in the system in a given state at a given time. Additionally, in the template entity `protein`, we defined two constant properties (`trate` and `drate`) which correspond to the rates of production of a protein from its corresponding gene (accounting for both transcription and translation) and the rate of its degradation.

Figure 2 depicts the taxonomy of all template processes used to describe the (possible) interactions within the system. To represent the basic processes for a single gene coding for a protein we specified the template process `basic`. This template contains two subprocesses, a process of `production` of a protein product from its coding gene in an unbound state, and a process of protein `degradation`. Furthermore, we defined a template process (`single_reversible_binding`) describing a single reversible binding of a protein to the promoter region of a gene, forming a bound factor complex and thus inhibiting the production of the protein which the bound gene codes for. We made this template process optional: An optional template process represents a two level hierarchy in which the top-level template process has two subordinate (child) template processes; one describing an empty process (representing the absence of interaction) and the other representing the presence of interaction. To account for the function of the inducer, we defined a template process `complex_formation` which describes the irreversible binding of an inducer with a free protein and the formation of an inducer-protein complex.

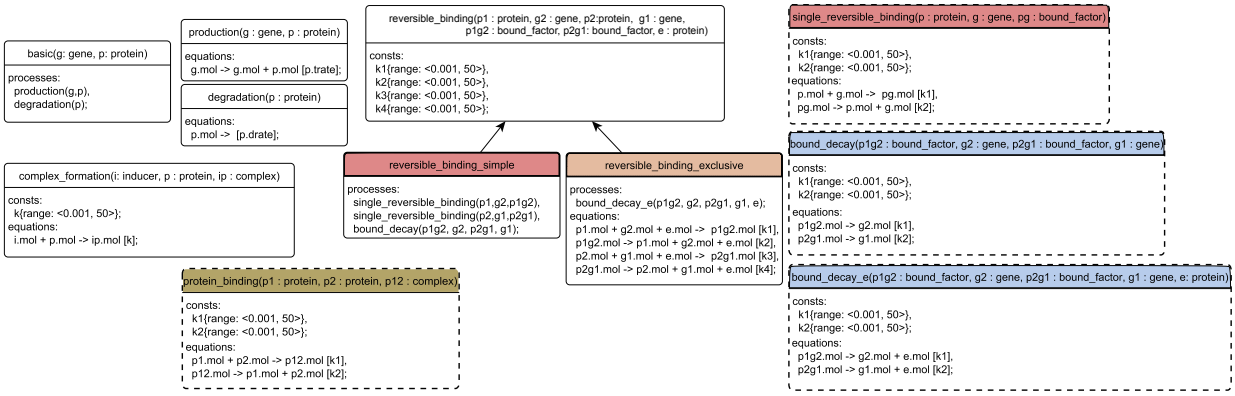

Figure 2. Taxonomy of template processes included in the library of domain knowledge for designing a stochastic toggle switch. The arrows in the taxonomy point from the child process template to the parent process template. The template process properties include a list of nested processes (components of the process in which they are nested), constants (usually corresponding to local kinetic rates) and equations (list of reaction equations describing the process). The template processes defined in a box with a dashed border represent optional processes.

In the library of domain knowledge, we next introduced the mechanisms suggested by Lipshtat et al.[17] as alternative or optional processes. First, we introduced an alternative to the simple reversible binding that allows for modelling exclusive binding (`reversible_binding_exclusive`). The exclusive binding assumes that both genes share a single binding site. Following Barnes et al.,[8] we introduced in the template process a protein entity `e` to which a protein should bind to before binding to the corresponding gene. Second, we introduced an optional template process describing reversible `protein_binding` of repressor proteins. Finally, we introduced two optional template processes that describe the degradation of a repressor protein while bound to the gene (`bound_decay` and `bound_decay_e`). These optional template processes define 2 modelling alternatives each, like the protein-protein binding template process.

Given the basic structure of the toggle switch, we were interested to find whether it has the optimally complex structure needed for achieving switching behaviour. Therefore, we explored the possibility of a simpler model structure (presence/absence of inhibitory interactions). An additional point of interest, which has not been addressed previously, is the possibility of achieving a better switch-like behaviour by considering combinations of the proposed mechanisms. To this end, we instantiated the specific constituent entities and processes as depicted in Fig. 3A. We instantiated two genes `gA` and `gB` from the template entity `gene`, two proteins `A` and `B`, two inducers `S` and `R`, the bound factors `AgB` and `BgA`, and the complexes `SA`, `RB` and `AB`.

The instantiation of the `basic` and the `complex_formation` template processes, for the gene-protein pairs gA-A and gB-B, and for the inducer-protein pairs S-A and R-B respectively, describes only one possible alternative of interaction. The instantiation of the top-level `reversible_binding` template process leads to 10 different alternatives of the process. Its mutually exclusive child template processes `reversible_binding_simple` and `reversible_binding_exclusive` describe $2 \times 2 \times 2 = 8$ and 2 alternatives correspondingly by their nested subprocesses: The subprocesses `single_reversible_binding`, `bound_decay` and `bound_decay_e` are optional and describe 2 alternatives each. The final instantiated top-level template process is the `protein_binding` optional process (2 alternatives) which finally led to definition of a space of $1 \times 1 \times 1 \times 1 \times 10 \times 2 = 20$ possible candidate model structures. By taking advantage of the knowledge available in the library, these 20 candidate model structures include, in addition to the four candidate models manually enumerated in the previous experiments, 16 other viable alternatives with simpler or more complex structure, containing some or all of the suggested mechanisms for achieving a switch-like behaviour. The reaction equations for the exclusive switch are shown in Fig. 3C, while the reaction equations for all candidate design structures are presented in the Supplementary Section 2.2.

Our experimental setup built on the experimental setups defined in previous work: We interpreted each model stochastically, i.e. the network of reaction equations with stochastic kinetic rates was automatically transformed into a continuous-time Markov chain with a finite state space. During each step of the optimization process for each model, we performed 100 realizations in the time frame $0 \leq t \leq 100$, sampled at each integer time point, which we used to calculate the values of the objective functions more accurately. The objective functions guiding the multi-objective optimization were defined as:

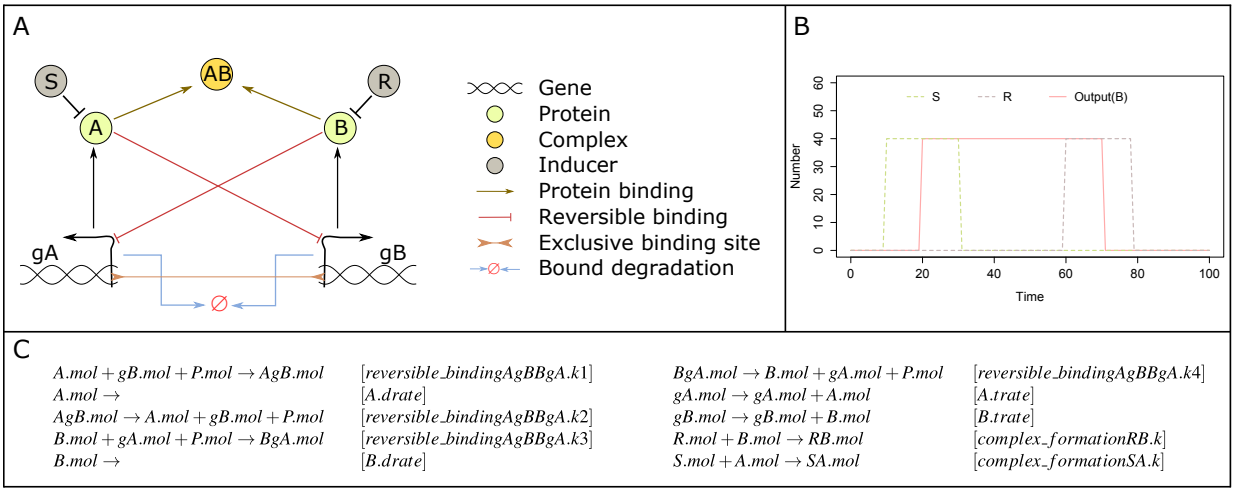

Figure 3. (A) Graphical representation of the toggle-switch interaction structure composed of instantiated templates from the library of domain knowledge. The colors of the instantiated components correspond to the color of the template processes from Fig. 2. (B) The expected behaviour of the observed output variable (number of molecules of protein B) and the input provided to the system (number of molecules of the inducers S and R) as a function of time. (C) The instantiated reaction equations for a single candidate design corresponding to the exclusive switch, where the symbol in the rectangular brackets denotes the rate (constant parameter) of the corresponding reaction.

$$O_1 = \sqrt{\sum_{t\in\alpha}(40-\widehat{x_t})^2/|\alpha|},$$
$$O_2 = \sqrt{\sum_{t\in\beta}(0-\widehat{x_t})^2/|\beta|},$$

where $\widehat{x_t}$ is the average simulated number of protein B molecules at time t, $\alpha = \{t : 30 < t \leq 60\}$, $\beta = \{t : 0 < t \leq 10 \vee 80 < t \leq 100\}$, and $|\alpha|$ and $|\beta|$ are the lengths of the intervals α and β. A model with optimal values for the objective functions ($O1 = 0, O2 = 0$) would have an ideal expected behaviour (as shown in Fig. 3B). Each candidate model structure has different complexity (number of reactions), which ranges from 6 for the least complex to 14 for the most complex structure. Therefore, the score used for ranking of each model structure was obtained by using the function from equation (2).

Figure 4A shows the obtained score profile. Considering the ranking of only the four candidate model structures considered in previous studies (bottom of Fig. 4A), the best ranked candidate (rank 1) has a structure containing an exclusive switch, the second best ranked candidate (rank 3) has a structure containing a protein-protein interaction, while the toggle switch with bound degradation and the original toggle switch have similar performance (being ranked 12 and 13, respectively). Our relative ranking of the four designs corresponds to the ranking obtained by Barnes et al.[8] and confirms the validity of our approach.

We further analysed the ranking and the structures of the models that perform better than the original toggle switch in relation to the aforementioned four models. Overall, the model structure containing an exclusive switch is ranked first, while the model structure with a protein-protein interaction is ranked third. Ranked second is the model structure containing both an exclusive switch and a protein-protein interaction, a model that has previously not been considered as a possible candidate model that can achieve the expected behaviour. It is worth noting that this model performs best in terms of HVUPF. The obtained Pareto fronts and simulations of these three models are shown in Fig. 4B and 4C. The model with bound degradation and the original toggle switch are ranked 12th and 13th. The Pareto fronts and simulations of the latter models are shown in Supplementary Fig. S1-S2.

Other than the fourth ranked model that contains exclusive binding and bound degradation, in between (rank 5 to 12) we observed models that could be structurally separated into two clusters, i.e. a cluster of model structures that contain a protein-protein interaction and a cluster of model structures that contain processes of bound degradation. Considering the better ranking of the model structure containing a protein-protein interaction, we noticed that this mechanism represents a good alternative to the inhibition by protein-gene binding (regarding the ability of the system to achieve toggle switch behaviour).

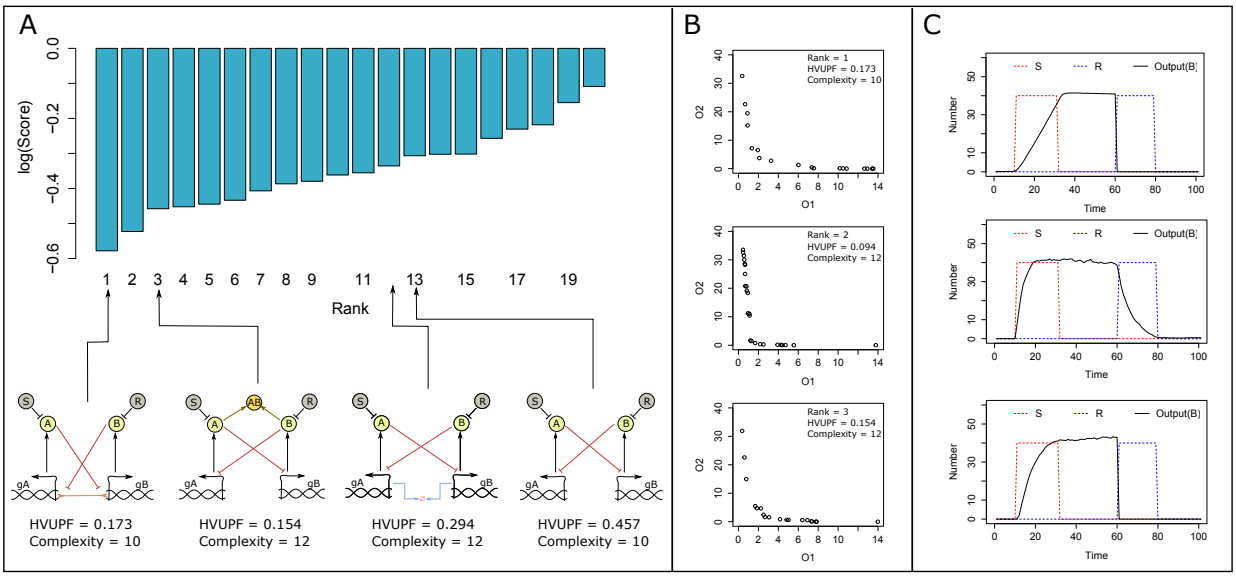

Figure 4. (A) Score profile containing all model structures ranked from the most appropriate to the least appropriate, for the stochastic toggle switch without cooperativity, according to the logarithm of the obtained score that takes into account complexity (lower score is better) (top). Sample candidate models containing structures considered in previous studies: from left to right according to their ranking, exclusive switch, switch with a protein-protein interaction, switch with bound repressor degradation, basic switch (bottom). (B) Pareto fronts of the three best performing models. (C) Simulations of the observed output variable (average of 100 realizations) of the three best performing models (using an arbitrarily selected point from the Pareto front).

Additionally, in both clusters we observed structures that contain only inhibition of the production of protein B by binding protein A to gene gB. This is due to the experimental design in which we defined an expected property of the behaviour that is dependent only on the number of protein B molecules observed in the system.

Robust negative feedback oscillations

Systems that can achieve stable oscillating behaviour are basic control parts that are critical for many biological systems. Therefore, the study of such systems, their improvement and implementation is an important task. Tsai et al.[18] studied a small set of five oscillating networks, based on a design consisting of three proteins and a negative feedback loop. The main point of interest there was whether adding a single auto interaction will lead to improved robustness of the system. In the study, the robustness of a candidate model was defined by its operational range of frequencies. The operational range was established by first taking a limited sample of the space of parameter values for each candidate model, then examining the frequency and the amplitude of the oscillations (if oscillation was achieved) for each set of parameter values and finally calculating the differences of the minimal and maximal achieved frequency. Given the high nonlinearity of the models, the relationship between the space of parameters and behaviours is complex. Consequently, a sampling approach might not accurately approximate the operational range.

In a follow up study, using a Bayesian design approach, Barnes et al.[8] focused on selecting a design out of five available candidates that can most likely achieve a fixed frequency and point-to-point amplitude. The Bayesian design approach considers concurrently different objective functions that guide the Markov chain Monte Carlo sampling process used to establish posteriors over the parameter values and model structures. However, it is computationally very demanding and therefore not feasible for use in the wide range of experiments that need to be performed to establish the robustness of oscillatory behaviour.

As shown in Fig. 5A, we encoded the knowledge available from the aforementioned studies into a library of domain knowledge by using the process-based formalism. In order to model a negative feedback loop of interacting proteins, we first introduced a template entity representing a protein with two variable properties: active concentration and passive concentration. We next introduced two top level processes: `Interaction` and `AutoInteraction`. The former describes a directed interaction between two protein entities. For modelling a negative feedback loop, we required only an inhibition interaction between two entities. Hence, we described the interaction as a change of the form of the protein (affected by the inhibition) from active to inactive, following a Michaelis-Menten rate law with cooperation (catalysed by the active form of the affecting protein).

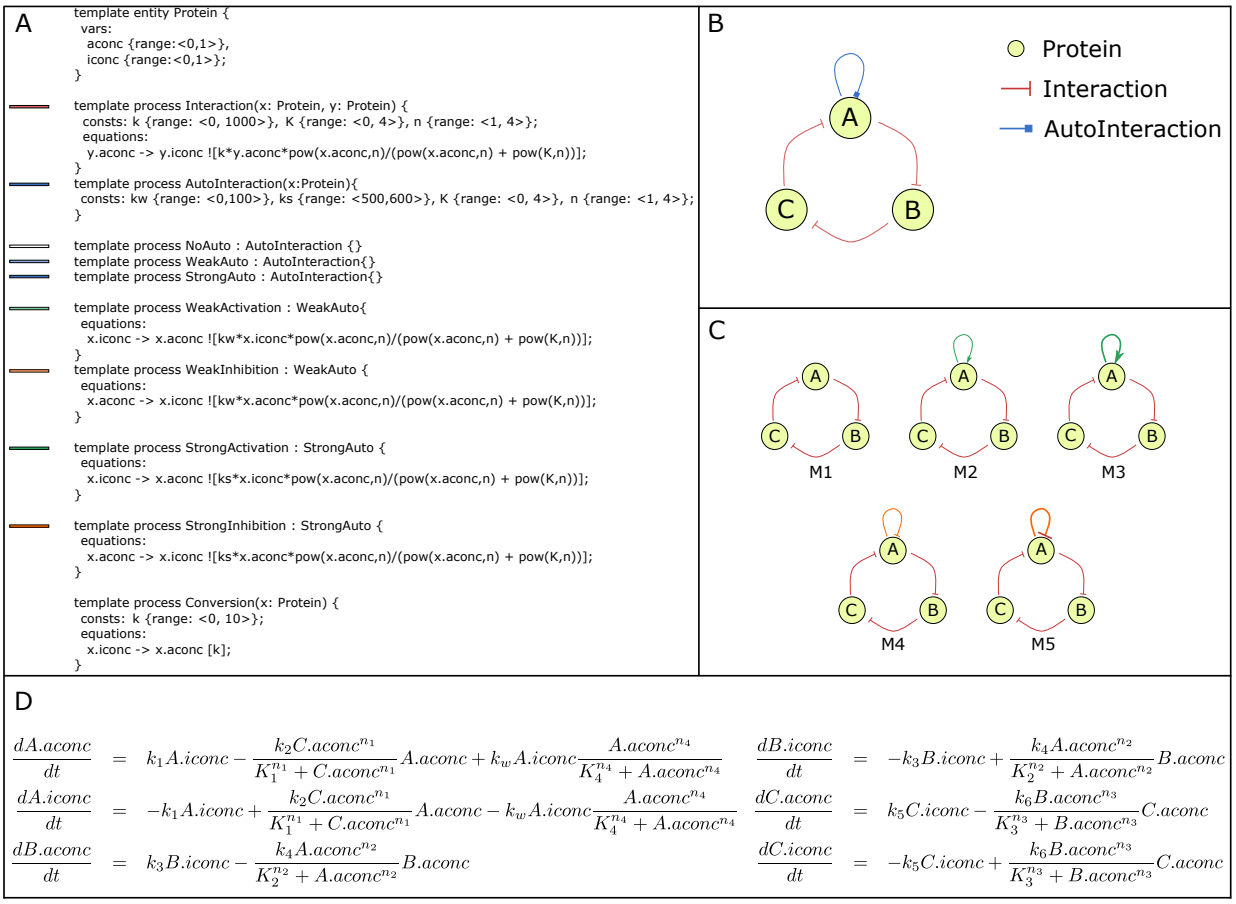

Figure 5. (A) The library of domain knowledge used for designing systems with robust negative feedback oscillations encoded in the process-based formalism (template entity and processes). (B) Graphical representation of the incomplete structure of the models as instantiated using components from the library of domain knowledge. (C) The five candidate model structures enumerated by the incomplete model. The thin loops in models 2 and 4 represent weak auto-interaction while the thicker lines in models 3 and 5 represent strong auto-interaction. The green lines represent auto-activation, while the orange lines represent auto-inhibition. (D) The system of coupled ordinary differential equations for the candidate design structure M2.

The latter was also used to describe an interaction loop for a single entity. In order to encode all possibilities as described by Tsai et al.,[18] we defined a taxonomy of template processes. The `AutoInteraction` template process is inherited by three second level template processes: `NoAuto`, which describes the case where there is no interaction loop; `WeakAuto`, which describes a weak interaction; and `StrongAuto`, which describes a strong interaction. The `WeakAuto` and `StrongAuto` template processes are inherited by leaf template processes which describe the corresponding weak and strong activation and inhibition interactions.

Using the described library of domain knowledge, we were able to define an incomplete model that can be refined to obtain all five candidate model structures described by Tsai et al.[18] The incomplete model is graphically depicted in Fig. 5B. The formal representation of the incomplete model is shown in Supplementary Table S3. We instantiated the template entity into three protein entities `A`, `B` and `C`. We defined an inhibitory loop by instantiating the `Interaction` template process into the three required inhibitory interactions. Finally, we instantiated the top-level `AutoInteraction` template process with protein `A` as its argument, effectively defining the space of possible candidate model structures. The five model structures described by the incomplete model are depicted in Fig. 5C. The system of coupled ordinary differential equations for the candidate design structure M2 is presented in Fig. 5D; the differential equations for the other four candidate designs are presented in the Supplementary Section 3.2.

To evaluate the performance of our approach and (at the same time) establish the most robust negative feedback oscillator

structure (in terms of its ability to achieve oscillations over a range of exact frequency - amplitude pairs), we performed two sets of nine design tasks. In order to compare our results to those of the previous studies, we considered (as target expected properties of the behaviour) the frequency-amplitude pairs formed by the Cartesian product of frequencies $f_t = \{0.1, 1, 10\}$ and amplitudes $A_t = \{0.01, 0.1, 1\}$, for the active concentrations of protein A and C (protein B is included by symmetry). Each candidate model was interpreted deterministically by automatically transforming the set of reaction equations to a system of ordinary differential equations, followed by a simulation in the time frame $0 \leq t \leq 28$ (sampled with a frequency of 40Hz). The objective functions guiding the multi-objective optimization were defined as:

$$\begin{aligned} O_1 &= |f_t - \hat{f}|, \\ O_2 &= |A_t - \hat{A}|, \\ O_3 &= \sum_n |\widehat{x_{t_0+nT}} - \widehat{x_{t_0+(n-1)T}}|, \end{aligned}$$

where f_t is the target frequency, $\hat{f}$ is the frequency obtained by calculating the largest component of the Fourier spectrum of the simulated trajectory of the model, A_t is the target amplitude, $\hat{A}$ is the amplitude determined from the simulated trajectory of the model, $\widehat{x_t}$ is the simulated active concentration of the target protein x at time t, $n \in \mathbb{Z}^*$ and $T = \hat{f}^{-1}$. All values were calculated using $t_0 = 2s$ in order to remove initial transients.

Figure 6 shows the HVUPF for each model for each design target. All of the models have the same number of reactions (considering M1 to contain an auto-interaction with $k = 0$). Therefore, the complexity component of the function used for scoring (equation (2)) is the same for all candidate models. Subsequently, we ranked the models only by their HVUPF. From both Fig. 6A and 6B, it is evident that the candidate models M2 and M3 consistently dominated the other models in all experimental setups, confirming the conclusions from the study by Tsai et al.[18] regarding the wide operational range (tunable frequency and constancy of amplitude), i.e. the robustness of the model structures containing an auto-activation loop and further

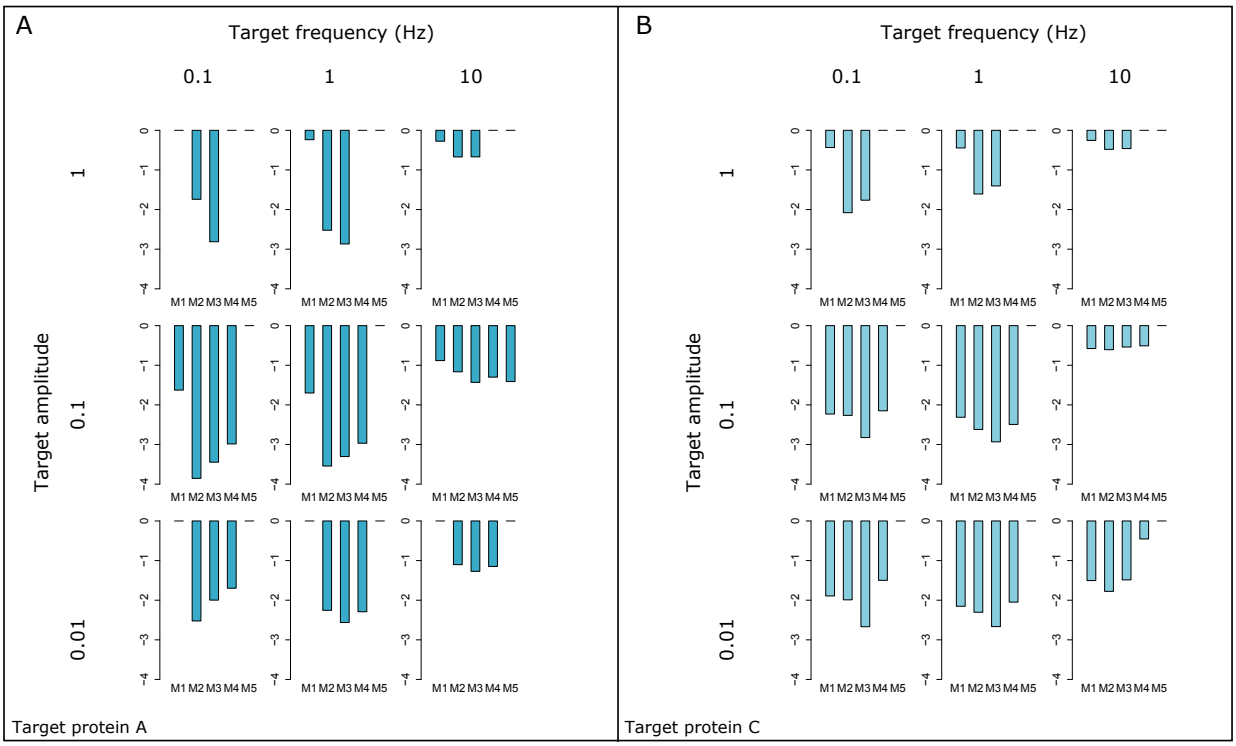

Figure 6. Unordered score profiles containing the logarithm of the HVUPF values obtained for each model structure under different experimental conditions (target frequency (Hz) and amplitude (conc.)) for observed active concentrations of (A) protein A and (B) protein C (a lower value is better). Each of the nine bar plots (score profiles) shows the obtained HVUPF (y-axis of the bar plots) of each model structure for the specific pair of target frequency and amplitude. The model enumeration (x-axis of the bar plots) is the same as in Fig. 5C. The obtained median ranks for (M1, M2, M3, M4, M5) are (4, 2, 1, 3, 4.5) and (3, 1, 2, 4, 5) for the observed target active concentrations of proteins A and C, respectively.

confirming the validity of our approach. For the model structures M2 and M3, the frequency operational range is extended towards the lower frequencies and the amplitude constancy range towards the higher end. The influence of the auto-activation loop is more expressed in the experiments where the target is the active concentration of protein A. The model structures containing auto-inhibition (M4, M5) do not offer significant improvement and extension of the operational range over the basic model structure M1.

The rankings for target frequency 1Hz and amplitude 0.1 correspond to the rankings obtained in the study by Barnes et al.[8] for the target expected properties of the behaviours of both proteins A and C. The difference in the rankings in both cases is the ranking of model structure M1 (negative feedback without auto-interaction) and model structure M4 (negative feedback with weak auto-inhibition). In our results, for this specific pair of target frequency and amplitude, M4 performs better than M1. Note that the two-dimensional projections of the Pareto front and the simulations for the best performing model structures M2 and M3, obtained using target amplitude $A_t = 0.1$ and different target frequencies, are given in the Supplementary. Supplementary Figures S3-S5 show those obtained for scenarios with observed target active concentrations of protein A and Supplementary Fig. S6-S8 show those obtained for scenarios with observed target active concentrations of protein C.

Discussion

The paper introduces process-based design, a novel approach to designing dynamical biological systems that exhibit desired behaviours. The process-based formalism we use allows for flexible and modular representation of modelling knowledge in the domain of interest, formalized as a taxonomy of template entities (constituents) and processes (interactions). We present ProBMoTd, an automated design tool that can make use of such knowledge by instantiating the templates into reusable components for building candidate models, which are put together into model structures in a manner similar to the one used by composition-based approaches.[5,6] It automatically resolves the structural uncertainty by enumerating and exploring the space of candidate models. Furthermore, ProBMoTd resolves the parametric uncertainty by using standard multi-objective optimization methods, as the ones used by optimization-based approaches.[8,11,13] It fits the values of the model parameters to the expected properties of the desired behaviour, obtaining a Pareto front on non-dominated optimal solutions. Finally, ProBMoTd combines the hyper-volume under the Pareto front with the complexity of the model structure to obtain a single score used for ranking the candidate designs. The process-based design is closely related to TinkerCell,[28] which employs a hierarchical representation of domain knowledge, but, in contrast to our approach, limits its focus to manual resolution of the structural and parametric uncertainties of the design task.

We illustrate the utility and generality of the process-based design approach on two design tasks. In the first, we design a stochastic toggle switch without cooperativity, while in the second, we design a deterministic oscillator. The experiments show that our approach is general enough to handle design tasks based on either deterministic or stochastic models. Furthermore, in both experiments, the ranking of candidate models based on the hyper-volume under the Pareto front of optimal solutions resembles the rankings reported in previous studies. This shows the utility of the hyper-volume measure as a design-selection strategy: It successfully summarizes the set of optimal solutions, produced by optimizing multiple competing objectives, into a single ranking score. In contrast to existing optimization-based approaches, process-based design facilitates modular knowledge representation, allowing for flexible specification of arbitrarily complex spaces of candidate model structures. This allowed us to easily specify a superset of the space of candidate model structures considered in previous studies, which subsequently led us to the discovery of previously unconsidered candidate designs of a stochastic toggle switch without cooperativity. Subsequently, we gained additional knowledge about the influence of the different choices of component processes on the overall model performance. Finally, when designing a robust oscillator, due to the automation of the complete ProBMoT workflow, we were able to efficiently perform a range of experiments with different setups, the results of which improved the confidence in the generality and robustness of the designs reported in previous studies.

The work presented in this paper lays the foundation for further development of process-based design. The first direction for further development concerns the use of heuristic search in resolving the structural uncertainty, i.e. efficient exploration of the space of candidate design. Composition-based approaches have been using parsimony heuristics for optimizing the complexity of logic circuits,[6] while optimization-based approaches have been using incomplete search strategies based on transforming the structural uncertainty into a parametric uncertainty.[12] This line of development will make our approach scalable to large spaces of candidate model structures by replacing the exhaustive enumeration of candidate designs with incomplete, heuristic strategies[29] for searching the space of candidate designs. Another direction of further work is the integration of our approach to process-based design with synthetic biology standards. Currently, process-based models produced by ProBMoTd can be recoded into the Systems Biology Markup Language.[30] We plan to develop computational methods for transferring knowledge from registries of standard parts[31] or similar standardization efforts[32] to process-based libraries of domain knowledge. This will allow for creating designs partly or completely composed of readily available biological components. It will also allow for translation of process-based models into standard formats specific to synthetic biology, such as the Synthetic Biology Open Language.[33]

References

1. Khalil, A. S. & Collins, J. J. Synthetic biology: applications come of age. *Nature Reviews. Genetics* **11**, 367–379 (2010).
2. Wolkenhauer, O. Why Model? *Frontiers in Physiology* **5** (2014).
3. Pedersen, M. & Phillips, A. Towards programming languages for genetic engineering of living cells. *Journal of the Royal Society Interface* **6**, S437–S450 (2009).
4. Bilitchenko, L. *et al.* Eugene a domain specific language for specifying and constraining synthetic biological parts, devices, and systems. *PLoS One* **6** (2011).
5. Beal, J., Lu, T. & Weiss, R. Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks. *PLoS One* **6**, 1–13 (2011).
6. Nielsen, A. A. K. *et al.* Genetic circuit design automation. *Science* **352** (2016).
7. Yaman, F., Bhatia, S., Adler, A., Densmore, D. & Beal, J. Automated selection of synthetic biology parts for genetic regulatory networks. *ACS Synthetic Biology* **1**, 332–344 (2012).
8. Barnes, C. P., Silk, D., Sheng, X. & Stumpf, M. P. H. Bayesian design of synthetic biological systems. *Proceedings of the National Academy of Sciences* **108**, 15190–15195 (2011).
9. Dasika, M. S. & Maranas, C. D. OptCircuit: an optimization based method for computational design of genetic circuits. *BMC Systems Biology* **2** (2008).
10. Rodrigo, G., Carrera, J. & Jaramillo, A. Genetdes: automatic design of transcriptional networks. *Bioinformatics* **23**, 1857–1858 (2007).
11. Sendin, J., Exler, O. & Banga, J. Multi-objective mixed integer strategy for the optimisation of biological networks. *Systems Biology, IET* **4**, 236–248 (2010).
12. Rodrigo, G., Carrera, J. & Jaramillo, A. Computational design of synthetic regulatory networks from a genetic library to characterize the designability of dynamical behaviors. *Nucleic Acids Research* **39**, e138 (2011).
13. Otero-Muras, I. & Banga, J. R. Multicriteria global optimization for biocircuit design. *BMC Systems Biology* **8**, 1–12 (2014).
14. Lu, L. & Anderson-Cook, C. M. Adapting the hypervolume quality indicator to quantify trade-offs and search efficiency for multiple criteria decision making using pareto fronts. *Quality and Reliability Engineering International* **29**, 1117–1133 (2013).
15. Bridewell, W., Langley, P., Todorovski, L. & Džeroski, S. Inductive process modelling. *Machine Learning* **71**, 109–130 (2008).
16. Tanevski, J., Todorovski, L. & Džeroski, S. Learning stochastic process-based models of dynamical systems from knowledge and data. *BMC Systems Biology* **10**, 30 (2016).
17. Lipshtat, A., Loinger, A., Balaban, N. Q. & Biham, O. Genetic toggle switch without cooperative binding. *Phys. Rev. Lett.* **96**, 188101 (2006).
18. Tsai, T. Y.-C. *et al.* Robust, tunable biological oscillations from interlinked positive and negative feedback loops. *Science* **321**, 126–129 (2008).
19. Džeroski, S. & Todorovski, L. Equation discovery for systems biology: finding the structure and dynamics of biological networks from time course data. *Current Opinion in Biotechnology* **19**, 360–368 (2008).
20. Tanevski, J., Todorovski, L., Kalaidzidis, Y. & Dzeroski, S. Domain-specific model selection for structural identification of the rab5-rab7 dynamics in endocytosis. *BMC Systems Biology* **9**, 31 (2015).
21. Čerepnalkoski, D., Taškova, K., Todorovski, L., Atanasova, N. & Džeroski, S. The influence of parameter fitting methods on model structure selection in automated modeling of aquatic ecosystems. *Ecological Modelling* **245**, 136 – 165 (2012).
22. Grünwald, P. D. *The Minimum Description Length Principle* (The MIT Press, 2007).
23. Machado, D. *et al.* Modeling formalisms in systems biology. *AMB Express* **1**, 1–14 (2011).
24. Kukkonen, S. & Lampinen, J. GDE3: The third Evolution Step of Generalized Differential Evolution. In *IEEE Congress on Evolutionary Computation (CEC'2005)*, 443 – 450 (2005).
25. Durillo, J. J. & Nebro, A. J. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software* **42**, 760–771 (2011).

26. Zitzler, E., Knowles, J. & Thiele, L. Quality assessment of pareto set approximations. In Branke, J., Deb, K., Miettinen, K. & Słowiński, R. (eds.) *Multiobjective Optimization*, vol. 5252 of *Lecture Notes in Computer Science*, 373–404 (Springer, 2008).

27. Gardner, T. S., Cantor, C. R. & Collins, J. J. Construction of a genetic toggle switch in escherichia coli. *Nature* **403**, 339–342 (2000).

28. Chandran, D. & Sauro, H. M. Hierarchical modeling for synthetic biology. *ACS Synthetic Biology* **1**, 353–364 (2012).

29. Eiben, A. E. & Smith, J. E. *Introduction to Evolutionary Computing* (Springer, 2003).

30. Hucka, M. *et al.* The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* **19**, 524–531 (2003).

31. Canton, B., Labno, A. & Endy, D. Refinement and standardization of synthetic biological parts and devices. *Nature Biotechnology* **26**, 787–793 (2008).

32. Ham, T. S. *et al.* Design, implementation and practice of JBEI-ICE: an open source biological part registry platform and tools. *Nucleic Acids Research* **40**, e141–e141 (2012).

33. Galdzicki, M. *et al.* The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nature Biotechnology* **32**, 545–550 (2014).

Acknowledgements

JT received funding from the Slovene Human Resources Development and Scholarship Fund (http://www.sklad-kadri.si/) and from the European Commission (http://ec.europa.eu/, Grant ICT-2013-604102 HBP). LT received funding from the Slovenian Research Agency (https://www.arrs.gov.si/, Grant P5-0093 (B)). SD received funding from the Slovenian Research Agency (https://www.arrs.gov.si/, Grant P2-0103) and the European Commision (http://ec.europa.eu/, Grants ICT-2013-612944 MAESTRA and ICT-2013-604102 HBP).

Author contributions statement

JT and SD initiated the work. JT implemented and integrated the new approach within ProBMoTs. JT encoded the domain knowledge in the process-based formalism. JT and LT conceived the experiments. JT performed the experiments. JT and LT analysed the results. JT and LT drafted the manuscript. All authors reviewed and approved the final manuscript.

Additional information

A combined supplementary information file is available. It contains a note on the availability of the software used in this study, listings of the formal representation of the libraries of domain knowledge and incomplete models, complete model equations, and additional figures for the tasks of designing a stochastic toggle switch without cooperativity and a deterministic oscillator.

Competing financial interests

The authors declare no competing financial interests.

5.3 Supplementary Material

Process-based design of dynamical biological systems –Supplementary material–

Jovan Tanevski, Ljupčo Todorovski and Sašo Džeroski

Contents

1 Availability of software

The ProBMoTd tool is an extension of ProBMoT that addresses the task of process-based design as described in the manuscript. Both ProBMoTd and ProBMoT are released as open-source software packages available for download at http://probmot.ijs.si. They are available under the terms of the three-clause BSD license (http://probmot.ijs.si/licence.html)

2 Stochastic toggle switch without cooperativity

2.1 Formal representation of the library and incomplete model

The complete library used for the task of design of a stochastic toggle switch without cooperativity (graphically depicted in Fig. 2 in the manuscript) is shown in Table S1. The incomplete model used to enumerate the 20 candidate models considered in this task is shown in Table S2.

Table S1: Formal representation of the library of domain knowledge used for the design of a stochastic toggle switch without cooperativity.

```
library ToggleLibrary;

template entity gene{
  vars: mol {range:<0,100>};
}

template entity protein{
  vars: mol {range:<0,100>};
  consts:
    trate {range: <0.001,50>},
    drate {range: <0.001,5>}
}

template entity complex{
  vars: mol {range:<0,100>};
}

template entity inducer{
  vars: mol {range:<0,100>};
}

template entity bound_factor{
  vars: mol {range:<0,100>};
}
```

continued . . .

Table S1: Formal representation of the library of domain knowledge used for the design of a stochastic toggle switch without cooperativity.

```
template process basic(g: gene, p: protein){
  processes:
    production(g, p),
    degradation(p);
}

template process production(g: gene, p: protein){
  equations:
    g.mol -> g.mol + p.mol [p.trate];
}

template process degradation(p: protein){
  equations:
    p.mol ->  [p.drate];
}

template process reversible_binding(p1: protein, g2: gene,
                  p2:protein, g1: gene, p1g2: bound_factor,
                            p2g1: bound_factor, e: protein){
  consts:
    k1{range: <0.001, 50>},
    k2{range: <0.001, 50>},
    k3{range: <0.001, 50>},
    k4{range: <0.001, 50>};
}

template process reversible_binding_simple : reversible_binding{
  processes:
    single_reversible_binding(p1, g2, p1g2),
    single_reversible_binding(p2, g1, p2g1),
    bound_decay(p1g2, g2, p2g1, g1);
}

template process reversible_binding_exclusive
                                          : reversible_binding{
  processes:
    bound_decay_e(p1g2, g2, p2g1, g1, e);
  equations:
    p1.mol + g2.mol + e.mol ->  p1g2.mol [k1],
    p1g2.mol -> p1.mol + g2.mol + e.mol [k2],
    p2.mol + g1.mol + e.mol ->  p2g1.mol [k3],
    p2g1.mol -> p2.mol + g1.mol + e.mol [k4];
}
```

continued . . .

Table S1: Formal representation of the library of domain knowledge used for the design of a stochastic toggle switch without cooperativity.

```
template process single_reversible_binding(p: protein, g: gene,
                                           pg: bound_factor){
  consts:
    k1{range: <0.001, 50>},
    k2{range: <0.001, 50>};
}

template process single_reversible_binding_none
                                    : single_reversible_binding{}

template process single_reversible_binding_simple
                                    : single_reversible_binding{
  equations:
    p.mol + g.mol ->  pg.mol [k1],
    pg.mol -> p.mol + g.mol [k2];
}

template process bound_decay(p1g2: bound_factor, g2: gene,
                             p2g1: bound_factor, g1: gene){
  consts:
    k1{range: <0.001, 50>},
    k2{range: <0.001, 50>};
}

template process bound_decay_none : bound_decay{}

template process bound_decay_present : bound_decay{
  equations:
    p1g2.mol -> g2.mol [k1],
    p2g1.mol -> g1.mol [k2];
}

template process bound_decay_e(p1g2: bound_factor, g2: gene,
                  p2g1: bound_factor, g1: gene, e: protein){
  consts:
    k1{range: <0.001, 50>},
    k2{range: <0.001, 50>};
}
```

continued . . .

Table S1: Formal representation of the library of domain knowledge used for the design of a stochastic toggle switch without cooperativity.

```
template process bound_decay_e_none : bound_decay_e{}

template process bound_decay_e_present : bound_decay_e{
  equations:
    p1g2.mol -> g2.mol + e.mol [k1],
    p2g1.mol -> g1.mol + e.mol [k2];
}

template process complex_formation(i: inducer, p: protein,
                                                ip: complex){
  consts:
    k{range: <0.001, 50>};
  equations:
    i.mol + p.mol -> ip.mol [k];
}

template process protein_binding(p1: protein, p2: protein,
                                               p1p2: complex){
  consts:
    k1{range: <0.001, 50>},
    k2{range: <0.001, 50>};
}

template process protein_binding_none : protein_binding {}

template process protein_binding_present : protein_binding{
  equations:
    p1.mol + p2.mol -> p1p2.mol [k1],
    p1p2.mol -> p1.mol + p2.mol [k2];
}
```

Table S2: Formal representation of the incomplete model used for the design of a stochastic toggle switch without cooperativity.

```
incomplete model ToggleSwitch : ToggleLibrary;

entity gA : gene{
  vars: mol{role: endogenous; initial: 1;};
}

entity gB : gene{
  vars: mol{role: endogenous; initial: 1;};
}

entity A : protein{
  vars: mol{role: endogenous; initial: 0;};
  consts: trate, drate;
}

entity B : protein{
  vars: mol{role: endogenous; initial: 0;};
  consts: trate, drate;
}

entity P : protein{
  vars: mol{role: endogenous; initial: 1;};
  consts: trate, drate;
}

entity SA : complex{
  vars: mol{role: endogenous; initial: 0;};
}

entity RB : complex{
  vars: mol{role: endogenous; initial: 0;};
}

entity AB : complex{
  vars: mol{role: endogenous; initial: 0;};
}

entity S : inducer{
  vars: mol{role: exogenous; initial: 0};
}
```

continued . . .

Table S2: Formal representation of the incomplete model used for the design of a stochastic toggle switch without cooperativity.

```
entity R : inducer{
  vars: mol{role: exogenous; initial: 0};
}

entity AgB : bound_factor{
  vars: mol{role: endogenous; initial: 0;};
}

entity BgA : bound_factor{
  vars: mol{role: endogenous; initial: 0;};
}

process basicA(gA, A) : basic {}
process basicB(gB, B) : basic {}

process complex_formationSA(S, A, SA) : complex_formation{
  consts: k;
}

process complex_formationRB(R,B,RB) : complex_formation{
  consts: k;
}

process reversible_bindingAgBBgA(A, gB, B, gA, AgB, BgA, P)
                                  : reversible_binding{
  consts: k1, k2, k3, k4;
}

process protein_bindingAB(A, B, AB) : protein_binding {
  consts: k1, k2;
}
```

2.2 Candidate model equations

Below we give the complete descriptions of the reactions and the corresponding rates for all candidate models considered for the task of designing a stochastic toggle switch without cooperativity.

Model 1: HVUPF=0.712234502, Complexity=6, Rank=15

Reaction	Rate
$A.mol \rightarrow$	$A.drate$
$B.mol \rightarrow$	$B.drate$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 2: HVUPF=0.426312245, Complexity=8, Rank=6

Reaction	Rate
$A.mol + gB.mol \rightarrow AgB.mol$	$single\_reversible\_binding.k1$
$A.mol \rightarrow$	$A.drate$
$AgB.mol \rightarrow A.mol + gB.mol$	$single\_reversible\_binding.k2$
$B.mol \rightarrow$	$B.drate$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 3: HVUPF=0.698638372, Complexity=8, Rank=17

Reaction	Rate
$A.mol \rightarrow$	$A.drate$
$B.mol + gA.mol \rightarrow BgA.mol$	$single\_reversible\_binding.k1$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow B.mol + gA.mol$	$single\_reversible\_binding.k2$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 4: HVUPF=0.457644189, Complexity=10, Rank=13

Reaction	Rate
$A.mol + gB.mol \rightarrow AgB.mol$	$single\_reversible\_binding\_AgB.k1$
$A.mol \rightarrow$	$A.drate$
$AgB.mol \rightarrow A.mol + gB.mol$	$single\_reversible\_binding\_AgB.k2$
$B.mol + gA.mol \rightarrow BgA.mol$	$single\_reversible\_binding\_BgA.k1$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow B.mol + gA.mol$	$single\_reversible\_binding\_BgA.k2$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 5: HVUPF=0.837428418, Complexity=8, Rank=19

Reaction	Rate
$A.mol \rightarrow$	$A.drate$
$AgB.mol \rightarrow gB.mol$	$bound\_decay.k1$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow gA.mol$	$bound\_decay.k2$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 6: HVUPF=0.392625279, Complexity=10, Rank=11

Reaction	Rate
$A.mol + gB.mol \rightarrow AgB.mol$	$single\_reversible\_binding.k1$
$A.mol \rightarrow$	$A.drate$
$AgB.mol \rightarrow A.mol + gB.mol$	$single\_reversible\_binding.k2$
$AgB.mol \rightarrow gB.mol$	$bound\_decay.k1$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow gA.mol$	$bound\_decay.k2$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 7: HVUPF=0.810606879, Complexity=10, Rank=20

Reaction	Rate
$A.mol \rightarrow$	$A.drate$
$AgB.mol \rightarrow gB.mol$	$bound\_decay.k1$
$B.mol + gA.mol \rightarrow BgA.mol$	$single\_reversible\_binding.k1$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow B.mol + gA.mol$	$single\_reversible\_binding.k2$
$BgA.mol \rightarrow gA.mol$	$bound\_decay.k2$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 8: HVUPF=0.294097735, Complexity=12, Rank=12

Reaction	Rate
$A.mol + gB.mol \rightarrow AgB.mol$	$single\_reversible\_binding\_AgB.k1$
$A.mol \rightarrow$	$A.drate$
$AgB.mol \rightarrow A.mol + gB.mol$	$single\_reversible\_binding\_AgB.k2$
$AgB.mol \rightarrow gB.mol$	$bound\_decay.k1$
$B.mol + gA.mol \rightarrow BgA.mol$	$single\_reversible\_binding\_BgA.k1$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow B.mol + gA.mol$	$single\_reversible\_binding\_BgA.k2$
$BgA.mol \rightarrow gA.mol$	$bound\_decay.k2$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 9: HVUPF=0.173166305, Complexity=10, Rank=1

Reaction	Rate
$A.mol + gB.mol + P.mol \rightarrow AgB.mol$	$reversible\_bindingAgBBgA.k1$
$A.mol \rightarrow$	$A.drate$
$AgB.mol \rightarrow A.mol + gB.mol + P.mol$	$reversible\_bindingAgBBgA.k2$
$B.mol + gA.mol + P.mol \rightarrow BgA.mol$	$reversible\_bindingAgBBgA.k3$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow B.mol + gA.mol + P.mol$	$reversible\_bindingAgBBgA.k4$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 10: HVUPF=0.159207006, Complexity=12, Rank=4

Reaction	Rate
$A.mol + gB.mol + P.mol \rightarrow AgB.mol$	$reversible\_bindingAgBBgA.k1$
$A.mol \rightarrow$	$A.drate$
$AgB.mol \rightarrow A.mol + gB.mol + P.mol$	$reversible\_bindingAgBBgA.k2$
$AgB.mol \rightarrow gB.mol + P.mol$	$bound\_decay\_e.k1$
$B.mol + gA.mol + P.mol \rightarrow BgA.mol$	$reversible\_bindingAgBBgA.k3$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow B.mol + gA.mol + P.mol$	$reversible\_bindingAgBBgA.k4$
$BgA.mol \rightarrow gA.mol + P.mol$	$bound\_decay\_e.k2$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 11: HVUPF=0.414773371, Complexity=8, Rank=5

Reaction	Rate
$A.mol + B.mol \rightarrow AB.mol$	$protein\_bindingAB.k1$
$A.mol \rightarrow$	$A.drate$
$AB.mol \rightarrow A.mol + B.mol$	$protein\_bindingAB.k2$
$B.mol \rightarrow$	$B.drate$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 12: HVUPF=0.331451333, Complexity=10, Rank=7

Reaction	Rate
$A.mol + B.mol \rightarrow AB.mol$	$protein\_bindingAB.k1$
$A.mol + gB.mol \rightarrow AgB.mol$	$single\_reversible\_binding.k1$
$A.mol \rightarrow$	$A.drate$
$AB.mol \rightarrow A.mol + B.mol$	$protein\_bindingAB.k2$
$AgB.mol \rightarrow A.mol + gB.mol$	$single\_reversible\_binding.k2$
$B.mol \rightarrow$	$B.drate$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 13: HVUPF=0.363104873, Complexity=10, Rank=9

Reaction	Rate
$A.mol + B.mol \rightarrow AB.mol$	$protein\_bindingAB.k1$
$A.mol \rightarrow$	$A.drate$
$AB.mol \rightarrow A.mol + B.mol$	$protein\_bindingAB.k2$
$B.mol + gA.mol \rightarrow BgA.mol$	$single\_reversible\_binding.k1$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow B.mol + gA.mol$	$single\_reversible\_binding.k2$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 14: HVUPF=0.153544691, Complexity=12, Rank=3

Reaction	Rate
$A.mol + B.mol \rightarrow AB.mol$	$protein\_bindingAB.k1$
$A.mol + gB.mol \rightarrow AgB.mol$	$single\_reversible\_binding\_AgB.k1$
$A.mol \rightarrow$	$A.drate$
$AB.mol \rightarrow A.mol + B.mol$	$protein\_bindingAB.k2$
$AgB.mol \rightarrow A.mol + gB.mol$	$single\_reversible\_binding\_AgB.k2$
$B.mol + gA.mol \rightarrow BgA.mol$	$single\_reversible\_binding\_BgA.k1$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow B.mol + gA.mol$	$single\_reversible\_binding\_BgA.k2$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 15: HVUPF=0.384784585, Complexity=10, Rank=10

Reaction	Rate
$A.mol + B.mol \rightarrow AB.mol$	$protein\_bindingAB.k1$
$A.mol \rightarrow$	$A.drate$
$AB.mol \rightarrow A.mol + B.mol$	$protein\_bindingAB.k2$
$AgB.mol \rightarrow gB.mol$	$bound\_decay.k1$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow gA.mol$	$bound\_decay.k2$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 16: HVUPF=0.339729811, Complexity=12, Rank=14

Reaction	Rate
$A.mol + B.mol \rightarrow AB.mol$	$protein\_bindingAB.k1$
$A.mol + gB.mol \rightarrow AgB.mol$	$single\_reversible\_binding.k1$
$A.mol \rightarrow$	$A.drate$
$AB.mol \rightarrow A.mol + B.mol$	$protein\_bindingAB.k2$
$AgB.mol \rightarrow A.mol + gB.mol$	$single\_reversible\_binding.k2$
$AgB.mol \rightarrow gB.mol$	$bound\_decay.k1$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow gA.mol$	$bound\_decay.k2$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 17: HVUPF=0.40783702, Complexity=12, Rank=16

Reaction	Rate
$A.mol + B.mol \rightarrow AB.mol$	$protein\_bindingAB.k1$
$A.mol \rightarrow$	$A.drate$
$AB.mol \rightarrow A.mol + B.mol$	$protein\_bindingAB.k2$
$AgB.mol \rightarrow gB.mol$	$bound\_decay.k1$
$B.mol + gA.mol \rightarrow BgA.mol$	$single\_reversible\_binding.k1$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow B.mol + gA.mol$	$single\_reversible\_binding.k2$
$BgA.mol \rightarrow gA.mol$	$bound\_decay.k2$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 18: HVUPF=0.347634959, Complexity=14, Rank=18

Reaction	Rate
$A.mol + B.mol \rightarrow AB.mol$	$protein\_bindingAB.k1$
$A.mol + gB.mol \rightarrow AgB.mol$	$single\_reversible\_binding\_AgB.k1$
$A.mol \rightarrow$	$A.drate$
$AB.mol \rightarrow A.mol + B.mol$	$protein\_bindingAB.k2$
$AgB.mol \rightarrow A.mol + gB.mol$	$single\_reversible\_binding\_AgB.k2$
$AgB.mol \rightarrow gB.mol$	$bound\_decay.k1$
$B.mol + gA.mol \rightarrow BgA.mol$	$single\_reversible\_binding\_BgA.k1$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow B.mol + gA.mol$	$single\_reversible\_binding\_BgA.k2$
$BgA.mol \rightarrow gA.mol$	$bound\_decay.k2$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 19: HVUPF=0.093538518, Complexity=12, Rank=2

Reaction	Rate
$A.mol + B.mol \rightarrow AB.mol$	$protein\_bindingAB.k1$
$A.mol + gB.mol + P.mol \rightarrow AgB.mol$	$reversible\_bindingAgBBgA.k1$
$A.mol \rightarrow$	$A.drate$
$AB.mol \rightarrow A.mol + B.mol$	$protein\_bindingAB.k2$
$AgB.mol \rightarrow A.mol + gB.mol + P.mol$	$reversible\_bindingAgBBgA.k2$
$B.mol + gA.mol + P.mol \rightarrow BgA.mol$	$reversible\_bindingAgBBgA.k3$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow B.mol + gA.mol + P.mol$	$reversible\_bindingAgBBgA.k4$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

Model 20: HVUPF=0.106389079, Complexity=14, Rank=8

Reaction	Rate
$A.mol + B.mol \rightarrow AB.mol$	$protein\_bindingAB.k1$
$A.mol + gB.mol + P.mol \rightarrow AgB.mol$	$reversible\_bindingAgBBgA.k1$
$A.mol \rightarrow$	$A.drate$
$AB.mol \rightarrow A.mol + B.mol$	$protein\_bindingAB.k2$
$AgB.mol \rightarrow A.mol + gB.mol + P.mol$	$reversible\_bindingAgBBgA.k2$
$AgB.mol \rightarrow gB.mol + P.mol$	$bound\_decay\_e.k1$
$B.mol + gA.mol + P.mol \rightarrow BgA.mol$	$reversible\_bindingAgBBgA.k3$
$B.mol \rightarrow$	$B.drate$
$BgA.mol \rightarrow B.mol + gA.mol + P.mol$	$reversible\_bindingAgBBgA.k4$
$BgA.mol \rightarrow gA.mol + P.mol$	$bound\_decay\_e.k2$
$gA.mol \rightarrow gA.mol + A.mol$	$A.trate$
$gB.mol \rightarrow gB.mol + B.mol$	$B.trate$
$R.mol + B.mol \rightarrow RB.mol$	$complex\_formationRB.k$
$S.mol + A.mol \rightarrow SA.mol$	$complex\_formationSA.k$

2.3 Supplementary figures

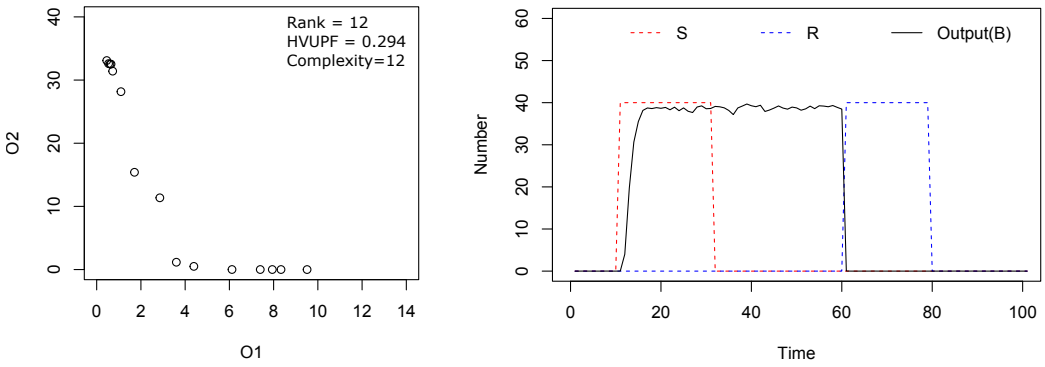

Figure S1: (left) Pareto front for the model structure with added bound degradation only (ranked 12th). (right) Simulation of the observed output variable (average of 100 realizations) using the model structure and an arbitrarily selected set of optimal parameter values from the Pareto front (single point).

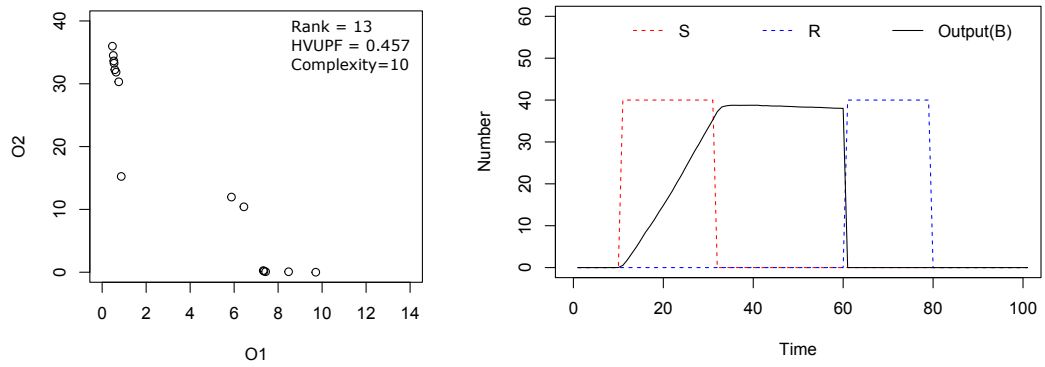

Figure S2: (left) Pareto front for the model structure describing a simple double repressed toggle switch without cooperativity (ranked 13th). (right) Simulation of the observed output variable (average of 100 realizations) using the model structure and an arbitrarily selected set of optimal parameter values from the Pareto front (single point).

3 Robust negative feedback oscillations

3.1 Formal representation of the library and incomplete model

The complete library is shown on Figure 4A in the manuscript. The incomplete model is shown in Table S3.

Table S3: Formal representation of the incomplete model used for the design of a robust negative feedback oscillatior.

```
incomplete model RobustOscillator : OscillatorNetworks;

entity A : Protein{
  vars:
    aconc{role: endogenous; initial: 0},
    iconc{role: endogenous; initial: 1};
}
entity B : Protein{
  vars:
    aconc{role: endogenous; initial: 0},
    iconc{role: endogenous; initial: 1};
}
entity C : Protein{
  vars:
    aconc{role: endogenous; initial: 0},
    iconc{role: endogenous; initial: 1};
}

process linkAB(A, B) : Inhibition { consts: k, K, n; }
process linkBC(B, C) : Inhibition { consts: k, K, n; }
process linkCA(C, A) : Inhibition { consts: k, K, n; }

process linkAA(A) : AutoInteraction {
  consts: kw, ks, K, n;
}

process AConv(A) : Conversion { consts: k; }
process BConv(B) : Conversion { consts: k; }
process CConv(C) : Conversion { consts: k=1; }
```

3.2 Candidate model equations

Model1

$$
\begin{aligned}
\frac{dA.aconc}{dt} &= k_1 A.iconc - \frac{k_2 C.aconc^{n_1}}{K_1^{n_1} + C.aconc^{n_1}} A.aconc \\
\frac{dA.iconc}{dt} &= -k_1 A.iconc + \frac{k_2 C.aconc^{n_1}}{K_1^{n_1} + C.aconc^{n_1}} A.aconc \\
\frac{dB.aconc}{dt} &= k_3 B.iconc - \frac{k_4 A.aconc^{n_2}}{K_2^{n_2} + A.aconc^{n_2}} B.aconc \\
\frac{dB.iconc}{dt} &= -k_3 B.iconc + \frac{k_4 A.aconc^{n_2}}{K_2^{n_2} + A.aconc^{n_2}} B.aconc \\
\frac{dC.aconc}{dt} &= k_5 C.iconc - \frac{k_6 B.aconc^{n_3}}{K_3^{n_3} + B.aconc^{n_3}} C.aconc \\
\frac{dC.iconc}{dt} &= -k_5 C.iconc + \frac{k_6 B.aconc^{n_3}}{K_3^{n_3} + B.aconc^{n_3}} C.aconc
\end{aligned}
$$

Model 2

$$
\begin{aligned}
\frac{dA.aconc}{dt} &= k_1 A.iconc - \frac{k_2 C.aconc^{n_1}}{K_1^{n_1} + C.aconc^{n_1}} A.aconc + k_w A.iconc \frac{A.aconc^{n_4}}{K_4^{n_4} + A.aconc^{n_4}} \\
\frac{dA.iconc}{dt} &= -k_1 A.iconc + \frac{k_2 C.aconc^{n_1}}{K_1^{n_1} + C.aconc^{n_1}} A.aconc - k_w A.iconc \frac{A.aconc^{n_4}}{K_4^{n_4} + A.aconc^{n_4}} \\
\frac{dB.aconc}{dt} &= k_3 B.iconc - \frac{k_4 A.aconc^{n_2}}{K_2^{n_2} + A.aconc^{n_2}} B.aconc \\
\frac{dB.iconc}{dt} &= -k_3 B.iconc + \frac{k_4 A.aconc^{n_2}}{K_2^{n_2} + A.aconc^{n_2}} B.aconc \\
\frac{dC.aconc}{dt} &= k_5 C.iconc - \frac{k_6 B.aconc^{n_3}}{K_3^{n_3} + B.aconc^{n_3}} C.aconc \\
\frac{dC.iconc}{dt} &= -k_5 C.iconc + \frac{k_6 B.aconc^{n_3}}{K_3^{n_3} + B.aconc^{n_3}} C.aconc
\end{aligned}
$$

Model 3

$$
\begin{aligned}
\frac{dA.aconc}{dt} &= k_1 A.iconc - \frac{k_2 C.aconc^{n_1}}{K_1^{n_1} + C.aconc^{n_1}} A.aconc + k_s A.iconc \frac{A.aconc^{n_4}}{K_4^{n_4} + A.aconc^{n_4}} \\
\frac{dA.iconc}{dt} &= -k_1 A.iconc + \frac{k_2 C.aconc^{n_1}}{K_1^{n_1} + C.aconc^{n_1}} A.aconc - k_s A.iconc \frac{A.aconc^{n_4}}{K_4^{n_4} + A.aconc^{n_4}} \\
\frac{dB.aconc}{dt} &= k_3 B.iconc - \frac{k_4 A.aconc^{n_2}}{K_2^{n_2} + A.aconc^{n_2}} B.aconc \\
\frac{dB.iconc}{dt} &= -k_3 B.iconc + \frac{k_4 A.aconc^{n_2}}{K_2^{n_2} + A.aconc^{n_2}} B.aconc \\
\frac{dC.aconc}{dt} &= k_5 C.iconc - \frac{k_6 B.aconc^{n_3}}{K_3^{n_3} + B.aconc^{n_3}} C.aconc \\
\frac{dC.iconc}{dt} &= -k_5 C.iconc + \frac{k_6 B.aconc^{n_3}}{K_3^{n_3} + B.aconc^{n_3}} C.aconc
\end{aligned}
$$

Model 4

$$
\begin{aligned}
\frac{dA.aconc}{dt} &= k_1 A.iconc - \frac{k_2 C.aconc^{n_1}}{K_1^{n_1} + C.aconc^{n_1}} A.aconc - k_w A.aconc \frac{A.aconc^{n_4}}{K_4^{n_4} + A.aconc^{n_4}} \\
\frac{dA.iconc}{dt} &= -k_1 A.iconc + \frac{k_2 C.aconc^{n_1}}{K_1^{n_1} + C.aconc^{n_1}} A.aconc + k_w A.aconc \frac{A.aconc^{n_4}}{K_4^{n_4} + A.aconc^{n_4}} \\
\frac{dB.aconc}{dt} &= k_3 B.iconc - \frac{k_4 A.aconc^{n_2}}{K_2^{n_2} + A.aconc^{n_2}} B.aconc \\
\frac{dB.iconc}{dt} &= -k_3 B.iconc + \frac{k_4 A.aconc^{n_2}}{K_2^{n_2} + A.aconc^{n_2}} B.aconc \\
\frac{dC.aconc}{dt} &= k_5 C.iconc - \frac{k_6 B.aconc^{n_3}}{K_3^{n_3} + B.aconc^{n_3}} C.aconc \\
\frac{dC.iconc}{dt} &= -k_5 C.iconc + \frac{k_6 B.aconc^{n_3}}{K_3^{n_3} + B.aconc^{n_3}} C.aconc
\end{aligned}
$$

Model 5

$$
\begin{aligned}
\frac{dA.aconc}{dt} &= k_1 A.iconc - \frac{k_2 C.aconc^{n_1}}{K_1^{n_1} + C.aconc^{n_1}} A.aconc - k_s A.aconc \frac{A.aconc^{n_4}}{K_4^{n_4} + A.aconc^{n_4}} \\
\frac{dA.iconc}{dt} &= -k_1 A.iconc + \frac{k_2 C.aconc^{n_1}}{K_1^{n_1} + C.aconc^{n_1}} A.aconc + k_s A.aconc \frac{A.aconc^{n_4}}{K_4^{n_4} + A.aconc^{n_4}} \\
\frac{dB.aconc}{dt} &= k_3 B.iconc - \frac{k_4 A.aconc^{n_2}}{K_2^{n_2} + A.aconc^{n_2}} B.aconc \\
\frac{dB.iconc}{dt} &= -k_3 B.iconc + \frac{k_4 A.aconc^{n_2}}{K_2^{n_2} + A.aconc^{n_2}} B.aconc \\
\frac{dC.aconc}{dt} &= k_5 C.iconc - \frac{k_6 B.aconc^{n_3}}{K_3^{n_3} + B.aconc^{n_3}} C.aconc \\
\frac{dC.iconc}{dt} &= -k_5 C.iconc + \frac{k_6 B.aconc^{n_3}}{K_3^{n_3} + B.aconc^{n_3}} C.aconc
\end{aligned}
$$

3.3 Supplementary figures

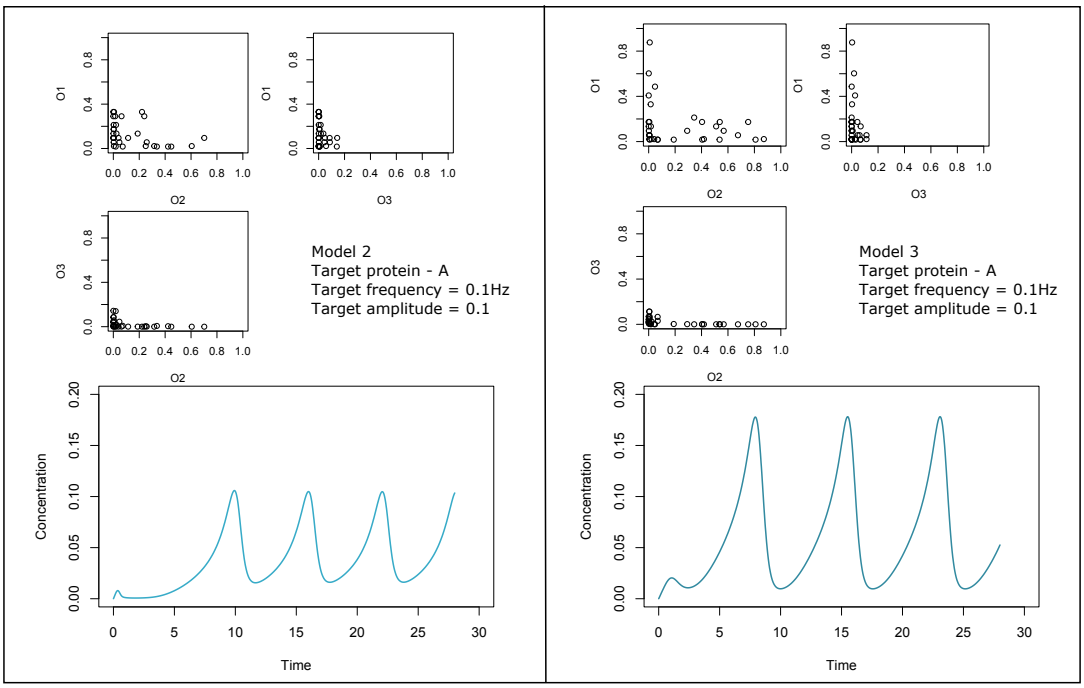

Figure S3: (top) Three two-dimensional projections of the Pareto front for the best performing models: (left) Model 2 and (right) Model 3, considering the behavior of the target active concentration of protein A described by a target frequency $f_t = 0.1Hz$ and a target amplitude $A_t = 0.1$. (bottom) Simulation of the active concentration of protein A using the structure of (left) Model 2 and (right) Model 3, and an arbitrarily selected set of optimal parameter values from the corresponding Pareto front (single point).

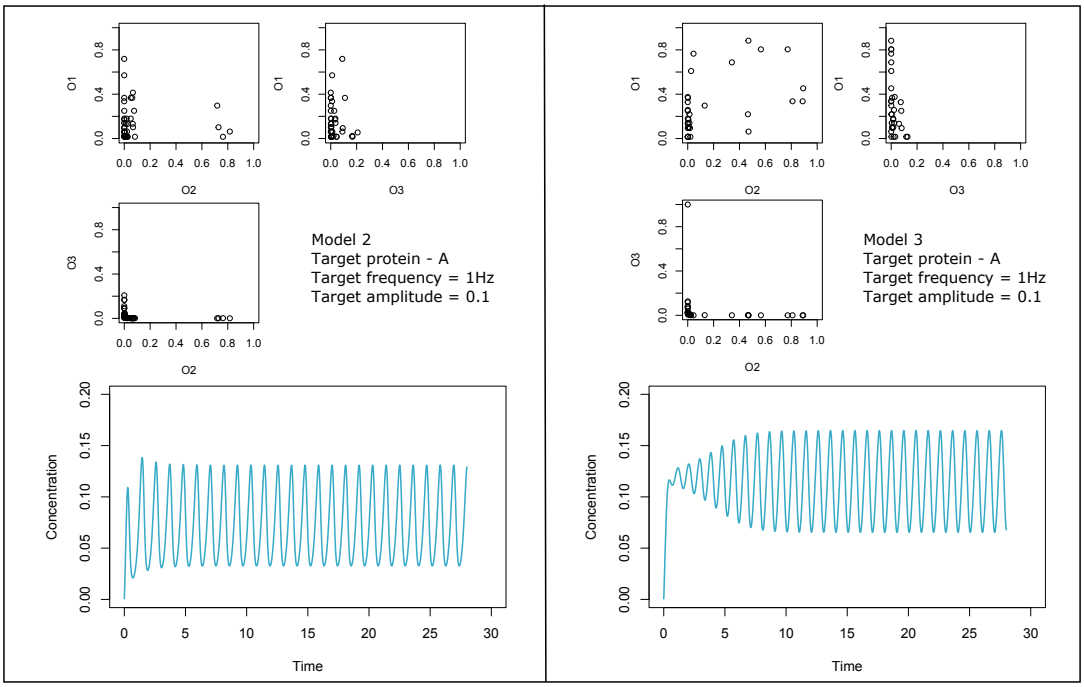

Figure S4: (top) Three two-dimensional projections of the Pareto front for the best performing models: (left) Model 2 and (right) Model 3, considering the behavior of the target active concentration of protein A described by a target frequency $f_t = 1Hz$ and a target amplitude $A_t = 0.1$. (bottom) Simulation of the active concentration of protein A using the structure of (left) Model 2 and (right) Model 3, and an arbitrarily selected set of optimal parameter values from the corresponding Pareto front (single point).

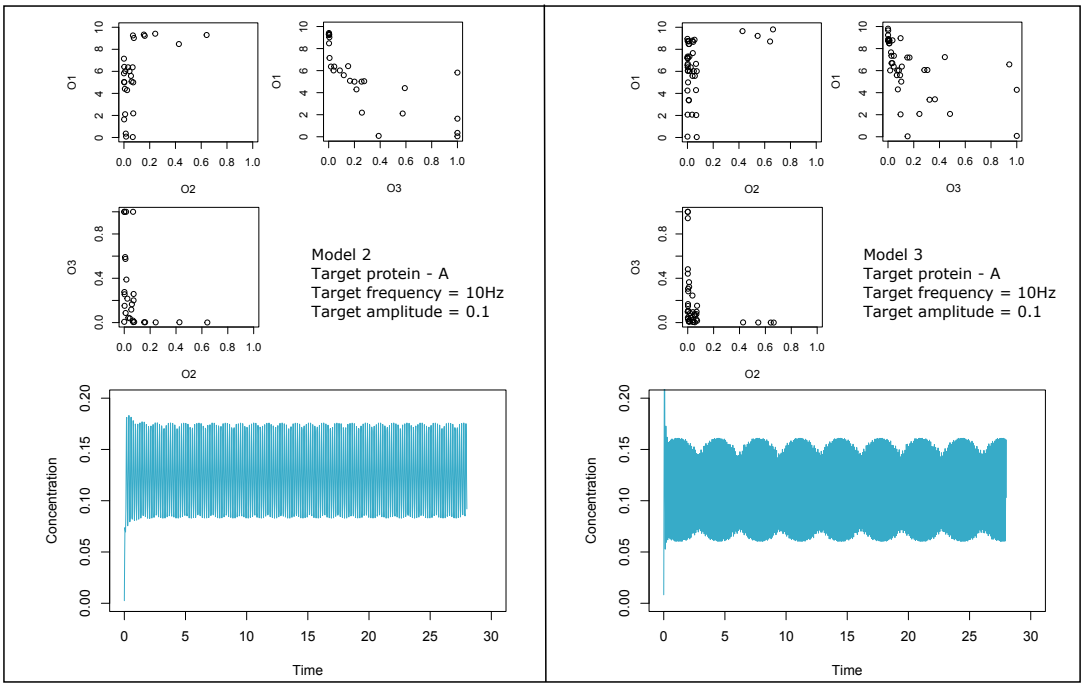

Figure S5: (top) Three two-dimensional projections of the Pareto front for the best performing models: (left) Model 2 and (right) Model 3, considering the behavior of the target active concentration of protein A described by a target frequency $f_t = 10Hz$ and a target amplitude $A_t = 0.1$. (bottom) Simulation of the active concentration of protein A using the structure of (left) Model 2 and (right) Model 3, and an arbitrarily selected set of optimal parameter values from the corresponding Pareto front (single point).

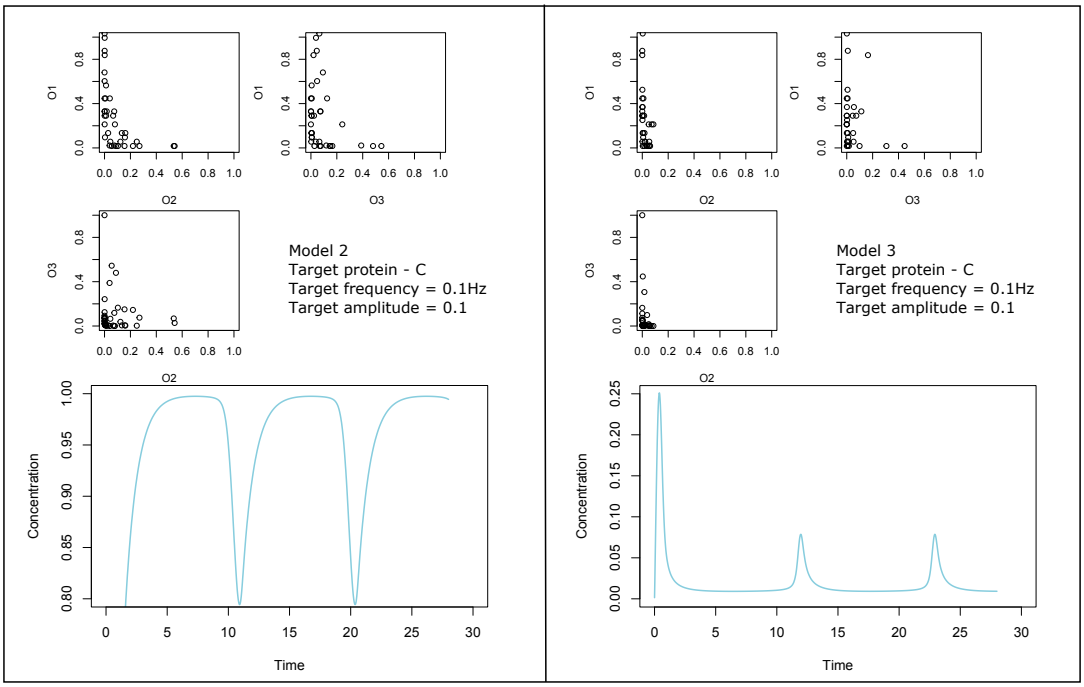

Figure S6: (top) Three two-dimensional projections of the Pareto front for the best performing models: (left) Model 2 and (right) Model 3, considering the behavior of the target active concentration of protein C described by a target frequency $f_t = 0.1Hz$ and a target amplitude $A_t = 0.1$. (bottom) Simulation of the active concentration of protein C using the structure of (left) Model 2 and (right) Model 3, and an arbitrarily selected set of optimal parameter values from the corresponding Pareto front (single point).

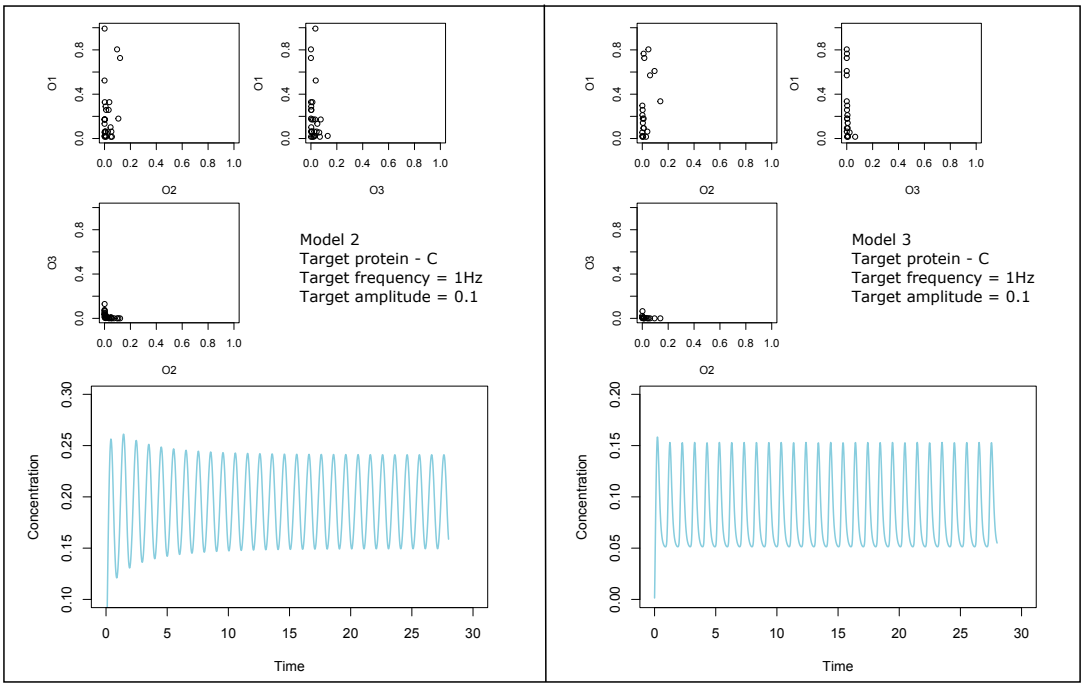

Figure S7: (top) Three two-dimensional projections of the Pareto front for the best performing models: (left) Model 2 and (right) Model 3, considering the behavior of the target active concentration of protein C described by a target frequency $f_t = 1Hz$ and a target amplitude $A_t = 0.1$. (bottom) Simulation of the active concentration of protein C using the structure of (left) Model 2 and (right) Model 3, and an arbitrarily selected set of optimal parameter values from the corresponding Pareto front (single point).

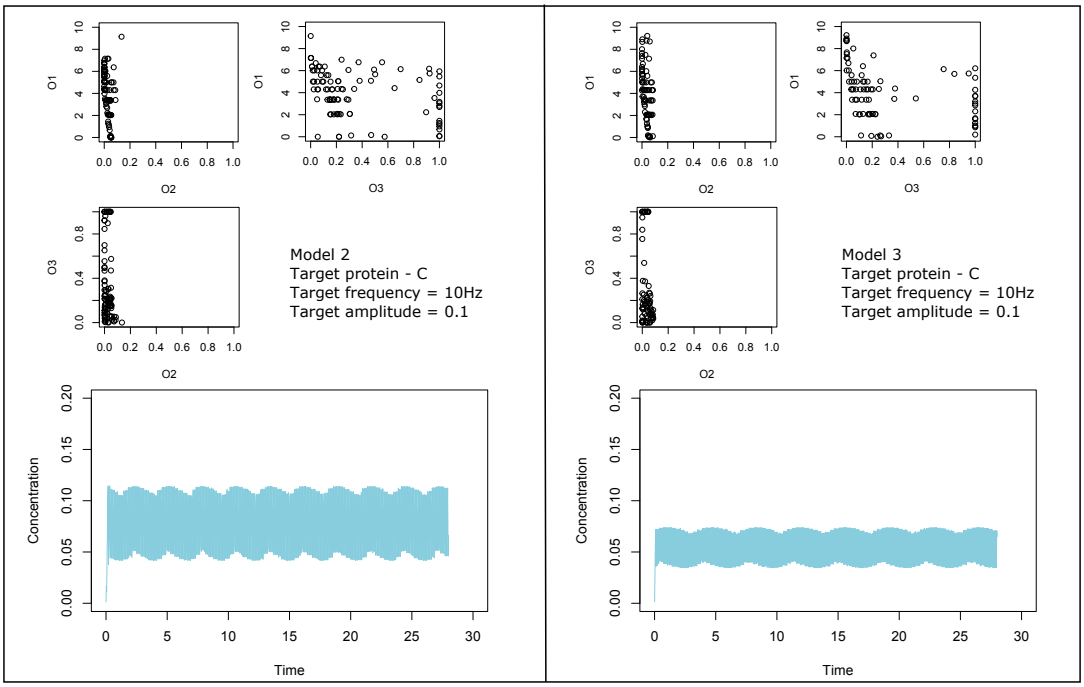

Figure S8: (top) Three two-dimensional projections of the Pareto front for the best performing models: (left) Model 2 and (right) Model 3, considering the behavior of the target active concentration of protein C described by a target frequency $f_t = 10Hz$ and a target amplitude $A_t = 0.1$. (bottom) Simulation of the active concentration of protein C using the structure of (left) Model 2 and (right) Model 3, and an arbitrarily selected set of optimal parameter values from the corresponding Pareto front (single point).

Chapter 6

Conclusion

In this thesis, we have presented an approach to the automated modeling and design of dynamical biological systems that bring together and extend machine learning approaches for process-based modeling (PBM), on one hand, and well established modeling approaches to systems and synthetic biology on the other hand. We have addressed several limitations of process-based modeling that have limited their use in the domains of systems and synthetic biology. These limitations concern the understandability and interpretability, model selection and dependence on data availability.

From a machine learning viewpoint, we have extended the scope of the formalism used in PBM to represent models and spaces of candidate models towards representing stochastic models (and domain knowledge). We extended the existing PBM tools to learn stochastic models represented in the new formalism from data and domain knowledge. We improved the PBM approach by adopting a regularization-based model selection scheme. We further improved the robustness of the PBM approach to limited observability, by using domain specific model selection criteria for bias strengthening. Finally, we completely relaxed the dependence of the PBM approach on observations, allowing for process-based design of novel systems by simultaneous optimization of multiple criteria related to the desired properties of the model system behavior.

From the perspective of systems and synthetic biology, the formalism allows for encoding domain specific knowledge and models with uncertainty in the structure and the values of the parameters in an understandable and easily communicable manner. The improved PBM approaches we have developed use the new formalism, resolve the structural and parametric uncertainty and select appropriate models.

The models encoded using the new process-based formalism can be interpreted both deterministically and stochastically. We performed proof of principle empirical evaluation of the new PBM approaches on multiple tasks of modeling and design from the domain of systems and synthetic biology. Through the results of the evaluation, we show that our approach is able to deal with complex parameter and structure search spaces in an automated fashion and produce reliable models and novel designs. The process-based modeling and design approach is successful in reconstructing results from the literature obtained by manual modeling and design or by using related approaches. It performs well on both synthetic and real world problems and is applicable to tasks of modeling and design of deterministic and stochastic systems.

6.1 Summary of Contributions

By improving the existing methods towards achieving the goals we have set, we have made the following contributions:

- We have broadened the scope of the formalisms used for process-based modeling of dynamical systems. We have extended the possibilities for interpretation of such models, mainly in the direction of capturing the inherent stochasticity of dynamical systems in biology. We have adapted for use within process-based modeling a formalism for representing complex dynamics that is both fine-grained and intuitive/comprehensible. Each process is represented by atomic interactions in the form of simple reaction equations. This allows for multiple deterministic and stochastic interpretations of process-based models. We have evaluated the approach to stochastic process-based modeling on four tasks coming from two domains, using synthetic and real world data.

- We have addressed the problem of model selection for process-based modeling. We have proposed a regularization based approach to model selection, based on the parsimony principle, adapted to the distinguishing characteristics of the task of process-based modeling. We have further proposed a bias strengthening approach to model selection, based on the use of domain-specific criteria, for tasks with limited and noisy observability. We have demonstrated on a modeling task concerning the dynamics of the Rab5-Rab7 conversion switch during the early stages of endocytosis, that additional bias strengthening based on domain-specific criteria, can alleviate the hard model selection problem and outperform the standard regularization approach.

- We have extended the scope of process-based modeling to cover the task of design of biological systems that can achieve desired behavior. We have developed a methodology for completely knowledge-driven process-based design of dynamical biological systems. We make use of the complete information available from the set (Pareto front) of optimal solutions obtained by optimizing multiple competing objectives, which correspond to the desired properties of candidate designs. We have evaluated the approach on two tasks concerning the design of a bistable switch and an oscillator. We have demonstrated the utility of the hyper-volume (under the Pareto front) indicator for overall ranking (and selection) of candidate process-based designs.

6.2 Further Work

During the process of development and evaluation of the approach for deterministic and stochastic process-based modeling and design of dynamical biological systems, we have identified several issues that lead to the following directions for further work.

The first direction is related to the potential improvement of the formalism for representing models and libraries of modeling domain knowledge, the integration of additional information within the formalism and the exploitation of other sources of knowledge. The process-based modeling formalism, in principle, allows for the integration of a range of other formalisms able to quantify the causal relations between entity properties within a self-contained process. Among the modeling formalisms for systems and synthetic biology, process algebras and rule-based formalisms are the most likely candidates to integrate with PBM, promising improvement of the efficiency and scalability of the knowledge representation. Both classes of formalisms are generalizations of the reaction-equation-based representation and are well accepted within the community. Integration with these formalisms would require the template entities to have more structured representations/additional properties. The template processes could then be used to define common structures by different sets of rules that transform these properties.

Another improvement of the formalism used to represent libraries of domain-knowledge would be to allow the annotation (Le Novére et al., 2005) of libraries and models. This

would lead to the enrichment of instantiated model components with information about the biological context, such as entity properties and compatibility, parameter ranges, or process alternatives from various biological databases. This would improve their understandability and communicability. A step further would be to develop methods for automated extraction and reuse of domain knowledge from publicly available repositories of models, such as BioModels (Le Novère et al., 2006) or CellML (Lloyd, Lawson, Hunter, & Nielsen, 2008).

The second direction of further work is closely related to the first and concerns the constraints on the search space of candidate models entailed by the library of domain knowledge.

The constraints currently used in process-based modeling are efficient, but limited. The incomplete model includes a set of instantiated entities which constrain the selection and instantiation of processes to be included in the model. The incomplete model is further constrained by the mutual exclusivity relations within the hierarchy of template processes from which the processes included in the model are instantiated. More expressive constraints can be considered that will assess the suitability/feasibility of models for further optimization, such as constraints based on knowledge of design principles (Alon, 2007; Tyson & Novak, 2010) and other structural properties. This will reduce the required computational time and facilitate the model selection problem, thus improving the efficiency of solving the overall modeling task. Topology based constraints could also be considered, expressed using a first-order predicate calculus or a more expressive higher order formalism that takes into account available annotations, compatibility issues or, in the case of design, the availability of standard parts and modules.

Another direction of further research is concerned with the process of parameter estimation and model selection. As discussed in the second chapter, there is an abundance of methods available for the estimation of parameters of dynamical biological systems. Their performance has been evaluated on tasks concerning a limited number of fixed and unrelated model structures. Although the performance of process-based modeling tasks is in the most part reliant on the efficiency of the parameter estimation step, a limited effort to understand this dependence has been made only in the context of deterministic process-based modeling of aquatic ecosystems (Čerepnalkoski et al., 2012). A more extensive comparison of the different available parameter estimation methods is needed for different tasks of process-based modeling and design from the domains of systems and synthetic biology. The tasks should have various amounts of structural uncertainty, involve both single and multiple objectives, and consider both deterministic and stochastic interpretation of models. The conclusions from previous studies point towards the consideration of global metaheuristic methods and their hybridization with local optimization methods.

The model selection method considered in this work is partly based on regularization. An analysis of different model selection schemes can (and should) be performed in the context of process-based modeling. Additionally, we have shown that various properties of model behavior can be optimized and used to improve model selection. An important aspect of the modeling problem is the analysis of how the different properties of behavior are related to the model structure (Kaltenbach, Dimopoulos, & Stelling, 2009; Shinar & Feinberg, 2010; Babtie, Kirk, & Stumpf, 2014). This further includes practical and structural parameter identifiability (Chis, Banga, & Balsa-Canto, 2011; Raue, Karlsson, Saccomani, Jirstrand, & Timmer, 2014) and robustness of the model (Stelling, Sauer, Szallasi, Doyle III, & Doyle, 2004; Kitano, 2007; Komorowski, Costa, Rand, & Stumpf, 2011). The results from such analyses are an important source of relevant information that can be considered within the model selection process.

The search based on exhaustive enumeration of and parameter optimization for every candidate model structure is computationally expensive and does not scale well with the

number of candidate models. In order to be able to apply process-based modeling to problems with a larger space of candidate models, it is necessary to develop new and better representations for model spaces and integration of methods for heuristic search through such space of candidate models in PBM.

There are two possible approaches to resolve this issue. One approach is in the development of better representations for the structural uncertainty currently encoded within an incomplete process-based model that could be directly considered by an optimization method. In this way, process-based modeling and design would become a completely optimization-driven approach. Another approach is the definition of structural refinement operators that can be applied to an incomplete process-based model and can be used by standard search methods to efficiently explore the space of candidate models. In order to establish their utility, a comparative evaluation of different approaches to search and optimization should be performed where the current results obtained with exhaustive enumeration would serve as a baseline.

Finally, an important direction is the further evaluation and application of the developed approach. In the domain of systems and synthetic biology, various real-world problems of identification and design of biological systems with desired behaviors, with increasing complexity, can be tackled by process-based modeling and design. The in-silico evaluation should be ideally complemented by wet lab experiments.

The systems approach to biology has had an immense impact on the development of the systems approach to medicine (Medina, 2012; Wolkenhauer, Auffray, Jaster, Steinhoff, & Dammann, 2013) and pharmacology (Iyengar, Zhao, Chung, Mager, & Gallo, 2012; E. I. Nielsen & Friberg, 2013). The application of process-based modeling to different problems from these domains is a final direction for proposed further work. Such applications will undoubtedly open up new directions for improvement of the methods we have proposed, which can in turn lead to more efficient inference of new, better models and designs.

Appendix A

Additional Information for Stochastic Process-Based Models of Dynamical Systems

A.1 Libraries of Domain Knowledge

All libraries are part of the supplementary data and materials for the contribution presented in Chapter 3 (Tanevski et al., 2016a).

A.1.1 Library of domain knowledge for modeling gene regulatory networks with global kinetic rates

```
library SyntheticNetworkGlobal;

template entity gene{
  vars:
    Pmol{range: <0, 200>},
    mRNAmol{range: <0, 200>};
}

template entity global{
  consts:
    alpha0{range: <0, 10>},
    alpha{range: <0, 500>},
    beta{range: <0, 10>},
    delta{range: <0, 10>},
    n{range: <0, 10>};
}

template process regulation(p1: gene, p2: gene, g: global){}

template process none : regulation {}

template process inhibition : regulation{
  equations:
      -> p2.mRNAmol ![g.alpha/(1 + pow(p1.Pmol,g.n))];
}
```

```
template process activation : regulation{
  equations:
      -> p2.mRNAmol ![g.alpha * pow(p1.Pmol,g.n)/(1 + pow(p1.Pmol,g.n))];
}

template process translation(p: gene, g: global){
  equations:
       -> p.mRNAmol [g.alpha0],
     p.mRNAmol -> p.Pmol + p.mRNAmol [g.beta];
}

template process degradation(p: gene, g: global){
  equations:
    p.Pmol -> [g.beta],
    p.mRNAmol -> [g.delta];
}
```

A.1.2 Library of domain knowledge for modeling gene regulatory networks with local kinetic rates

```
library SyntheticNetworkLocal;

template entity gene{
  vars:
    Pmol{range: <0, 200>},
    mRNAmol{range: <0, 200>};
  consts:
    alpha0{range: <0, 10>},
    beta{range: <0, 10>},
    delta{range: <0, 10>};
}

template process regulation(p1: gene, p2: gene){
  consts:
    alpha{range: <0, 500>},
    n{range: <0, 10>};
}

template process none : regulation {}

template process inhibition : regulation{
  equations:
      -> p2.mRNAmol ![alpha/(1 + pow(p1.Pmol,n))];
}

template process activation : regulation{
  equations:
      -> p2.mRNAmol ![alpha * pow(p1.Pmol,n)/(1 + pow(p1.Pmol,n))];
}
```

```
template process translation(p: gene){
  equations:
       -> p.mRNAmol [p.alpha0],
     p.mRNAmol -> p.Pmol + p.mRNAmol [p.beta];
}

template process degradation(p: gene){
  equations:
    p.Pmol -> [p.beta],
    p.mRNAmol -> [p.delta];
}
```

A.1.3 Library of domain knowledge for compartmental epidemiological modeling

```
library Epidemiology;

template entity pop_compartment{
  vars:
    noi{range: <0, 300>};
}

template process root(S: pop_compartment, L: pop_compartment, A: pop_compartment,
 I: pop_compartment, Q: pop_compartment, R: pop_compartment){
  equations:
    L.noi -> [0],
    A.noi -> [0],
    Q.noi -> [0],
    R.noi -> [0];
}

template process symptomatic : root{
  processes:
    infection_symptomatic(S, L, I),
    recovery_symptomatic(S, I, Q, R),
    control(I, Q);
}

template process asymptomatic : root{
  processes:
    infection_asymptomatic(S, L, A, I),
    recovery_asymptomatic(S, A, I, Q, R),
    control(I, Q);
}
```

```
template process infection_symptomatic(S: pop_compartment, L: pop_compartment,
 I: pop_compartment){
  consts:
    i1{range:<1e-4, 10>},
    i2{range:<1e-4, 10>},
    i3{range:<1e-4, 10>};
}

template process si : infection_symptomatic{
  equations: S.noi + I.noi -> I.noi + I.noi [i1];
}

template process sli : infection_symptomatic{
  equations:
    S.noi + I.noi -> L.noi + I.noi [i2],
    L.noi -> I.noi [i3];
}

template process recovery_symptomatic(S: pop_compartment, I: pop_compartment,
 Q:pop_compartment, R:pop_compartment){
  consts:
    r1 {range:<1e-4,10>},
    r2 {range:<1e-4,10>};
}

template process is : recovery_symptomatic{
  equations:
    I.noi -> S.noi [r1],
    Q.noi -> S.noi [r1];
}

template process ir : recovery_symptomatic{
  equations:
    I.noi -> R.noi [r1],
    Q.noi -> R.noi [r1];
}

template process irs : recovery_symptomatic{
  equations:
    I.noi -> R.noi [r1],
    Q.noi -> R.noi [r1],
    R.noi -> S.noi [r2];
}
```

```
template process infection_asymptomatic(S: pop_compartment, L: pop_compartment,
 A: pop_compartment, I: pop_compartment){
  consts:
    i1a{range: <1e-4, 10>},
    p{range: <1e-3, 1>},
    n{range: <1, 10>},
    i2a{range: <1e-4, 10>},
    i3a{range:<1e-4, 10>};
}

template process sia : infection_asymptomatic{
  equations:
    S.noi + I.noi -> I.noi + I.noi [p*i1a],
    S.noi + I.noi -> A.noi + I.noi [(1-p)*i1a],
    S.noi + A.noi -> I.noi + A.noi [p*n*i1a],
    S.noi + A.noi -> A.noi + A.noi [(1-p)*n*i1a];
}

template process slia : infection_asymptomatic{
  equations:
    S.noi + I.noi -> L.noi + I.noi [i2a],
    L.noi -> I.noi [p*i3a],
    S.noi + A.noi -> L.noi + A.noi [n*i2a],
    L.noi -> A.noi [(1-p)*i3a];
}

template process recovery_asymptomatic(S: pop_compartment, A: pop_compartment,
 I: pop_compartment, Q: pop_compartment, R: pop_compartment){
  consts:
    r1a{range: <1e-4, 10>},
    r2a{range: <1e-4, 10>};
}

template process ias : recovery_asymptomatic{
  equations:
    I.noi -> S.noi [r1a],
    A.noi -> S.noi [r1a],
    Q.noi -> S.noi [r1a];
}

template process iar : recovery_asymptomatic{
  equations:
    I.noi -> R.noi [r1a],
    A.noi -> R.noi [r1a],
    Q.noi -> R.noi [r1a];
}
```

```
template process iars : recovery_asymptomatic{
  equations:
    I.noi -> R.noi [r1a],
    A.noi -> R.noi [r1a],
    Q.noi -> R.noi [r1a],
    R.noi -> S.noi [r2a];
}


template process control(I: pop_compartment, Q: pop_compartment){
  consts: q{range: <1e-4, 10>};
}

template process control_none : control{}

template process control_quarantine : control{
  equations: I.noi -> Q.noi [q];
}
```

A.2 Incomplete Models

All incomplete models are part of the supplementary data and materials for the contribution presented in Chapter 3 (Tanevski et al., 2016a).

A.2.1 Incomplete model of a gene regulatory network with global kinetic rates

```
incomplete model SyntheticNetworkGlobal : SyntheticNetworkGlobal;

entity TetR : gene{
  vars:
    Pmol{role: endogenous; initial: 5;},
    mRNAmol{role: endogenous; initial: 0;};
}

entity LacI : gene{
  vars:
    Pmol{role: endogenous; initial: 0;},
    mRNAmol{role: endogenous; initial: 0;};
}

entity cI : gene{
  vars:
    Pmol{role: endogenous; initial: 15;},
    mRNAmol{role: endogenous; initial: 0;};
}
```

```
entity g : global{
  consts: alpha0, alpha, beta, delta = 1, n;
}

process regulation1(TetR, cI, g) : regulation{}
process regulation2(cI, LacI, g) : regulation{}
process regulation3(LacI, TetR, g) : regulation{}
process regulation4(TetR, LacI, g) : regulation{}
process regulation5(LacI, cI, g) : regulation{}
process regulation6(cI, TetR, g) : regulation{}

process TetRtranslation(TetR, g) : translation{}
process LacItranslation(LacI, g) : translation{}
process cItranslation(cI, g) : translation{}

process TetRdegradation(TetR, g) : degradation{}
process LacIdegradation(LacI, g) : degradation{}
process cIdegradation(cI, g) : degradation{}
```

A.2.2 Incomplete model of a gene regulatory network with local kinetic rates

```
incomplete model SyntheticNetworkLocal : SyntheticNetworkLocal;

entity TetR : gene{
  vars:
    Pmol{role: endogenous; initial: 5;},
    mRNAmol{role: endogenous; initial: 0;};
  consts: alpha0, beta, delta = 1;
}

entity LacI : gene{
  vars:
    Pmol{role: endogenous; initial: 0;},
    mRNAmol{role: endogenous; initial: 0;};
  consts: alpha0, beta, delta = 1;
}

entity cI : gene{
  vars:
    Pmol{role: endogenous; initial: 15;},
    mRNAmol{role: endogenous; initial: 0;};
  consts: alpha0, beta, delta = 1;
}

process regulation1(TetR, cI) : regulation{
  consts: alpha, n;
}
process regulation2(cI, LacI) : regulation{
  consts: alpha, n;
}
```

```
process regulation3(LacI, TetR) : regulation{
  consts: alpha, n;
}
process regulation4(TetR, LacI) : regulation{
  consts: alpha, n;
}
process regulation5(LacI, cI) : regulation{
  consts: alpha, n;
}
process regulation6(cI, TetR) : regulation{
  consts: alpha, n;
}

process TetRtranslation(TetR) : translation {}
process LacItranslation(LacI) : translation {}
process cItranslation(cI) : translation {}

process TetRdegradation(TetR) : degradation {}
process LacIdegradation(LacI) : degradation {}
process cIdegradation(cI) : degradation {}
```

A.2.3 Incomplete model of the Eyam plague outbreak

```
incomplete model Eyam : Epidemiology;

entity S : pop_compartment{
  vars: noi{role: endogenous; initial: 254;};
}

entity L : pop_compartment{
  vars: noi{role: endogenous; initial: 0;};
}

entity A : pop_compartment{
  vars: noi{role: endogenous; initial: 0;};
}

entity I : pop_compartment{
  vars: noi{role: endogenous; initial: 7;};
}

entity Q : pop_compartment{
  vars: noi{role: endogenous; initial: 0;};
}

entity R : pop_compartment {
  vars: noi{role: endogenous; initial: 0;};
}

process eyam_root(S,L,A,I,Q,R) : root{}
```

A.2.4 Incomplete model of the Tristan da Cunha influenza outbreak

```
incomplete model TdC : Epidemiology;

entity S : pop_compartment{
  vars: noi{role: endogenous; initial: null;};
}

entity L : pop_compartment{
  vars: noi{role: endogenous; initial: 0;};
}


entity A : pop_compartment{
  vars: noi{role: endogenous; initial: 0;};
}

entity I : pop_compartment{
  vars: noi{role: endogenous; initial: 1;};
}

entity Q : pop_compartment{
  vars: noi{role: endogenous; initial: 0;};
}

entity R : pop_compartment {
  vars: noi{role: endogenous; initial: 0;};
}

process tdc_root(S,L,A,I,Q,R) : root{}
```

Appendix B

Additional Information for Domain Specific Criteria for Process-Based Modeling

B.1 Library of Domain Knowledge for Modeling the Rab5-Rab7 Dynamics in Endocytosis

Part of Additional file 2 from the contribution presented in Chapter 4 (Tanevski et al., 2015).

```
library EndocytosisLibrary;

template entity Protein{
  vars:
    GDP_bound_state_conc{range: <0, 2>},
    GTP_bound_state_conc{range: <0, 2>},
    GEF,
    GAP,
    t;
  consts:
    GDI_dissociation_flux{range: <0.001, 4>},
    GDI_association_rate{range: <0.001, 4>};
}

template process Root(p1 : Protein, p2: Protein){
  consts: td{range: <5, 195>};
  processes:
    GDI_GDP_membrane_interaction(p1),
    GDI_GDP_membrane_interaction(p2),
    GEFProcess(p1, p2),
    GEFCombined(p1, p2),
    GAPProcessPlus(p1, p2),
    GAPProcess(p2, p1);
```

```
  equations:
    td(p1.GDP_bound_state_conc) = -p1.GEF * (p1.t/(p1.t+td))*p1.GDP_bound_state_conc
    + p1.GAP * p1.GTP_bound_state_conc,
    td(p1.GTP_bound_state_conc) =  p1.GEF * (p1.t/(p1.t+td))*p1.GDP_bound_state_conc
    - p1.GAP * p1.GTP_bound_state_conc,
    td(p2.GDP_bound_state_conc) = -p2.GEF * p2.GDP_bound_state_conc + p2.GAP
    * p2.GTP_bound_state_conc,
    td(p2.GTP_bound_state_conc) =  p2.GEF * p2.GDP_bound_state_conc - p2.GAP
    * p2.GTP_bound_state_conc;
}

template process GDI_GDP_membrane_interaction(p: Protein){
  processes:
    Disassociation_from_GDI(p),
    Association_with_GDI(p);
}

template process Disassociation_from_GDI(p: Protein){
  equations: td(p.GDP_bound_state_conc) = p.GDI_dissociation_flux;
}

template process Association_with_GDI(p: Protein){
  equations:
    td(p.GDP_bound_state_conc) = -p.GDI_association_rate * p.GDP_bound_state_conc;
}

template process GEFProcess(p1: Protein, p2: Protein){
  consts:
    ke{range: <0.001, 4>}, kf{range: <0.001, 4>}, kg{range: <0.001, 4>},
    km{range: <0.001, 4>}, ki{range: <0.001, 4>};
}

template process MMKinetics : GEFProcess{
  equations: p1.GEF = ke*p1.GTP_bound_state_conc/(kg + p1.GTP_bound_state_conc);
}

template process Sigmoidal_response : GEFProcess{
  equations: p1.GEF = ke/(1 + exp(kg - p1.GTP_bound_state_conc)*kf);
}

template process Exchange_inhibition : GEFProcess{
  equations: p1.GEF = ke*p1.GTP_bound_state_conc/(km*(1+p2.GTP_bound_state_conc/ki)
  + p1.GTP_bound_state_conc);
}

template process GEFCombined(p1: Protein, p2: Protein){
  consts:
    ke{range: <0.001, 4>}, kf{range: <0.001, 4>}, kg{range: <0.001, 4>},
    km{range: <0.001, 4>}, ki{range: <0.001, 4>}, kE{range: <0.001, 4>};
}
```

```
template process MMActivation : GEFCombined{
  processes: MMKinetics(p2, p1);
  equations: p2.GEF = ke*p1.GTP_bound_state_conc/(kg + p1.GTP_bound_state_conc);
}

template process MMAuto_Sigmoidal : GEFCombined{
  processes: MMKinetics(p2, p1);
  equations: p2.GEF = ke/(1 + exp(kg - p1.GTP_bound_state_conc)*kf)
}

template process Sigmoidal_MMActivation : GEFCombined{
  processes : Sigmoidal_response(p2, p1);
  equations : p2.GEF = ke*p1.GTP_bound_state_conc/(kg + p1.GTP_bound_state_conc);
}

template process Sigmoidal_Sigmoidal : GEFCombined{
  processes: Sigmoidal_response(p2, p1);
  equations: p2.GEF = ke/(1 + exp(kg - p1.GTP_bound_state_conc)*kf)
}

template process Sigmoidal_only : GEFCombined{
  processes: Sigmoidal_response(p2,p1);
}

template process NoAct_MM : GEFCombined{
  equations: p2.GEF = ke + kE*p1.GTP_bound_state_conc/(kg + p1.GTP_bound_state_conc);
}

template process NoAct_Sigmoidal : GEFCombined{
  equations: p2.GEF = ke + kE/(1 + exp(kg - p1.GTP_bound_state_conc)*kf);
}

template process GAPProcessPlus(p1: Protein, p2: Protein){
  consts: kh{range: <0.001, 4>}, kH{range: <0.001, 4>}, ky{range: <0.001, 4>};
}
template process GAPProcess : GAPProcessPlus{}

template process Sigmoidal : GAPProcessPlus{
  equations: p1.GAP = kh/(1 + exp(kH - p2.GTP_bound_state_conc)*ky);
}

template process Intrinsic_Hydrolysis : GAPProcess{
  equations: p1.GAP = kh;
}

template process MM : GAPProcess{
  processes: Intrinsic_Hydrolysis(p1, p2);
  equations: p1.GAP = kH*p2.GTP_bound_state_conc/(ky + p2.GTP_bound_state_conc);
}
```

B.2 Incomplete Model of the Rab5-Rab7 Dynamics in Endocytosis

Part of Additional file 2 from the contribution presented in Chapter 4 (Tanevski et al., 2015).

```
incomplete model EndocytosisModel : EndocytosisLibrary;

entity rab5 : Protein{
  vars:
    GDP_bound_state_conc{role: endogenous; initial: null;},
    GTP_bound_state_conc{role: endogenous; initial: null;},
    GEF{role: endogenous},
    GAP{role: endogenous},
    t{role: exogenous};
  consts:
    GDI_dissociation_flux,
    GDI_association_rate;
}

entity rab7 : Protein {
  vars:
    GDP_bound_state_conc{role: endogenous; initial: null;},
    GTP_bound_state_conc{role: endogenous; initial: null;},
    GEF{role: endogenous},
    GAP{role: endogenous},
    t{role: exogenous};
  consts:
    GDI_dissociation_flux,
    GDI_association_rate;
}

process root(rab5, rab7) : Root{
  consts: td;
  processes:
    GDI_GDP_membrane_interaction5,
    GDI_GDP_membrane_interaction7,
    GEF5Process,
    GEF7Process,
    GAP5Process,
    GAP7Process;
}

process GDI_GDP_membrane_interaction5(rab5) :  GDI_GDP_membrane_interaction{
  processes: Disassociation_from_GDI5, Association_with_GDI5;
}

process GDI_GDP_membrane_interaction7(rab7) :  GDI_GDP_membrane_interaction{
  processes: Disassociation_from_GDI7, Association_with_GDI7;
}
```

```
process Disassociation_from_GDI5(rab5) : Disassociation_from_GDI{}

process Association_with_GDI5(rab5) : Association_with_GDI{}

process Disassociation_from_GDI7(rab7) : Disassociation_from_GDI{}

process Association_with_GDI7(rab7) : Association_with_GDI{}

process GEF5Process(rab5, rab7): GEFProcess{
  consts: ke,kf,kg,km,ki;
}

process GEF7Process(rab5, rab7): GEFCombined{
  consts: ke,kf,kg,km,ki,kE;
}

process GAP5Process(rab5, rab7) : GAPProcessPlus{
  consts: kh,kH,ky;
}

process GAP7Process(rab7, rab5) : GAPProcess{
  consts: kh,kH,ky;
}
```

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723.

Akutsu, T., Miyano, S., & Kuhara, S. (2000). Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, *16*(8), 727–734.

Alon, U. (2007). *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman and Hall/CRC.

Arkin, A., Ross, J., & McAdams, H. H. (1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage λ - infected Escherichia coli cells. *Genetics*, *149*(4), 1633–1648.

Ashyraliyev, M., Fomekong-Nanfack, Y., Kaandorp, J., & Blom, J. (2009). Systems biology: parameter estimation for biochemical models. *FEBS Journal*, *276*(4), 886–902.

Babtie, A. C., Kirk, P., & Stumpf, M. P. H. (2014). Topological sensitivity analysis for systems biology. *Proceedings of the National Academy of Sciences*, *111*(52), 18507–18512.

Bansal, M., Belcastro, V., Ambesi-Impiombato, A., & di Bernardo, D. (2006). How to infer gene networks from expression profiles. *Molecular Systems Biology*, *3*, 78–78.

Bansal, M., Gatta, G. D., & di Bernardo, D. (2006). Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics*, *22*(7), 815–822.

Barnes, C., Silk, D., Sheng, X., & Stumpf, M. (2011). Bayesian design of synthetic biological systems. *Proceedings of the National Academy of Sciences*, *108*(37), 15190–15195.

Bartocci, E. & Lió, P. (2016). Computational modeling, formal analysis, and tools for systems biology. *PLoS Computational Biology*, *12*(1), 1–22.

Basso, K., Margolin, A. A., Stolovitzky, G., Klein, U., Dalla-Favera, R., & Califano, A. (2005). Reverse engineering of regulatory networks in human b cells. *Nature Genetetics*, *37*(4), 382–390.

Beal, J., Lu, T., & Weiss, R. (2011). Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks. *PLoS One*, *6*(8), 1–13.

Bilitchenko, L., Liu, A., Cheung, S., Weeding, E., Xia, B., Leguia, M., & Anderson, J. (2011). Eugene a domain specific language for specifying and constraining synthetic biological parts, devices, and systems. *PLoS One*, *6*(4), 1–12.

Blinov, M. L., Yang, J., Faeder, J. R., & Hlavacek, W. S. (2006). Transactions on Computational Systems Biology VII. In C. Priami, A. Ingólfsdóttir, B. Mishra, & H. Riis Nielson (Eds.), (Chap. Graph Theory for Rule-Based Modeling of Biochemical Networks, pp. 89–106). Springer Berlin Heidelberg.

Bonneau, R., Reiss, D., Shannon, P., Facciotti, M., Hood, L., Baliga, N., & Thorsson, V. (2006). The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biology*, *7*(5), R36.1–R36.16.

Bridewell, W. & Langley, P. (2010). Two kinds of knowledge in scientific discovery. *Topics in Cognitive Science*, *2*(1), 36–52.

Bridewell, W., Langley, P., Todorovski, L., & Džeroski, S. (2008). Inductive Process Modelling. *Machine Learning*, *71*, 109–130.

Cai, Y., Hartnett, B., Gustafsson, C., & Peccoud, J. (2007). A syntactic model to design and verify synthetic genetic constructs derived from standard biological parts. *Bioinformatics*, *23*(20), 2760–2767.

Cai, Y., Lux, M. W., Adam, L., & Peccoud, J. (2009). Modeling Structure-Function Relationships in Synthetic DNA Sequences using Attribute Grammars. *PLoS Computational Biology*, *5*(10), 1–10.

Cameron, D. E., Bashor, C. J., & Collins, J. J. (2014). A brief history of synthetic biology. *Nature Reviews Microbiology*, *12*(5), 381–390.

Canton, B., Labno, A., & Endy, D. (2008). Refinement and standardization of synthetic biological parts and devices. *Nature Biotechnology*, *26*, 787–793.

Cardelli, L. (2005). Brane calculi. In *Proceedings of Computational Methods in Systems Biology* (pp. 257–280). Springer, Berlin.

Carr, P. A. & Church, G. M. (2009). Genome engineering. *Nature Biotechnology*, *27*(12), 1151–1162.

Cedersund, G. & Roll, J. (2009). Systems biology: model based evaluation and comparison of potential explanations for given biological data. *FEBS Journal*, *276*(4), 903–922.

Čerepnalkoski, D. (2013, September). *Process-based models of dynamical systems: representation and induction* (Doctoral dissertation, Jožef Stefan International Postgraduate School, Ljubljana, Slovenia).

Čerepnalkoski, D., Taškova, K., Todorovski, L., Atanasova, N., & Džeroski, S. (2012). The influence of parameter fitting methods on model structure selection in automated modeling of aquatic ecosystems. *Ecological Modelling*, *245*, 136–165.

Chaouiya, C. (2007). Petri net modelling of biological networks. *Briefings in Bioinformatics*, *8*(4), 210–219.

Chaves, M., Albert, R., & Sontag, E. D. (2005). Robustness and fragility of boolean models for genetic regulatory networks. *Journal of Theoretical Biology*, *235*(3), 431–449.

Chis, O.-T., Banga, J. R., & Balsa-Canto, E. (2011, November). Structural identifiability of systems biology models: a critical comparison of methods. *PLoS ONE*, *6*(11), 1–16.

Chou, I.-C. & Voit, E. O. (2009). Recent developments in parameter estimation and structure identification of biochemical and genomic systems. *Mathematical Biosciences*, *219*(2), 57–83.

Chylek, L. A., Harris, L. A., Tung, C.-S., Faeder, J. R., Lopez, C. F., & Hlavacek, W. S. (2014). Rule-based modeling: a computational approach for studying biomolecular site dynamics in cell signaling systems. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, *6*(1), 13–36.

Ciocchetta, F. & Hillston, J. (2009). Bio-PEPA: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science*, *410*(33–34), 3065–3084.

Cong, L., Ran, F. A., Cox, D., Lin, S., Barretto, R., Habib, N., ... Zhang, F. (2013). Multiplex Genome Engineering Using CRISPR/Cas Systems. *Science*, *339*(6121), 819–823.

Danos, V. & Krivine, J. (2007). Formal Molecular Biology Done in CCS-R. *Electronic Notes in Theoretical Computer Science*, *180*(3), 31–49. Proceedings of the First Workshop on Concurrent Models in Molecular Biology (BioConcur 2003).

Dasika, M. & Maranas, C. (2008). OptCircuit: An optimization based method for computational design of genetic circuits. *BMC Systems Biology*, *2*(1), 1–19.

Del Conte-Zerial, P., Brusch, L., Rink, J. C., Collinet, C., Kalaidzidis, Y., Zerial, M., & Deutsch, A. (2008). Membrane identity and gtpase cascades regulated by toggle and cut-out switches. *Molecular Systems Biology*, *4*, 206–206.

Doudna, J. A. & Charpentier, E. (2014). The new frontier of genome engineering with crispr-cas9. *Science*, *346*(6213).

Džeroski, S. & Todorovski, L. (2002). Logical and computational aspects of model-based reasoning. In L. Magnani, N. J. Nersessian, & C. Pizzi (Eds.), (Chap. Encoding and Using Domain Knowledge on Population Dynamics for Equation Discovery, pp. 227–247). Springer Netherlands.

Džeroski, S. & Todorovski, L. (Eds.). (2007). *Computational Discovery of Scientific Knowledge*. Springer Berlin Heidelberg.

Džeroski, S. & Todorovski, L. (2008). Equation discovery for systems biology: finding the structure and dynamics of biological networks from time course data. *Current Opinion in Biotechnology*, *19*(4), 360–368.

Elowitz, M. & Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, *403*, 335–338.

Faeder, J. R., Blinov, M. L., & Hlavacek, W. S. (2009). Rule-based modeling of biochemical systems with BioNetGen. In *Systems biology* (Vol. 500, pp. 113–167). Methods in Molecular Biology. New York: Humana Press.

Fages, F., Soliman, S., & Chabrier-Rivier, N. (2004). Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. *Journal of Biological Physics and Chemistry*, *4*, 64–73.

Feret, J., Danos, V., Krivine, J., Harmer, R., & Fontana, W. (2009). Internal coarse-graining of molecular systems. *Proceedings of the National Academy of Sciences*, *106*(16), 6453–6458.

Forbus, K. D. (1984). Qualitative process theory. *Artificial Intelligence*, *24*(1-3), 85–168.

François, P. & Hakim, V. (2004). Design of genetic networks with specified functions by evolution in silico. *Proceedings of the National Academy of Sciences*, *101*(2), 580–585.

Gábor, A. & Banga, J. R. (2015). Robust and efficient parameter estimation in dynamic models of biological systems. *BMC Systems Biology*, *9*(1), 1–25.

Gaj, T., Gersbach, C. A., & Barbas III, C. F. (2013). ZFN, TALEN, and CRISPR/Cas-based methods for genome engineering. *Trends in Biotechnology*, *31*(7), 397–405.

Gardner, T. S., Cantor, C. R., & Collins, J. J. (2000). Construction of a genetic toggle switch in Escherichia coli. *Nature*, *403*(6767), 339–342.

Gardner, T. S., di Bernardo, D., Lorenz, D., & Collins, J. J. (2003). Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, *301*(5629), 102–105.

Gibson, M. A. & Bruck, J. (2000). Efficient exact stochastic simulation of chemical systems with many species and many channels. *The Journal of Physical Chemistry A*, *104*(9), 1876–1889.

Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, *22*(4), 403–434.

Gillespie, D. T. (1992). A rigorous derivation of the chemical master equation. *Physica A: Statistical Mechanics and its Applications*, *188*(1–3), 404–425.

Gillespie, D. T. (2000). The chemical Langevin equation. *The Journal of Chemical Physics*, *113*(1), 297–306.

Gillespie, D. T. (2007). Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, *58*(1), 35–55.

Gillespie, D. T. & Petzold, L. R. (2003). Improved leap-size selection for accelerated stochastic simulation. *The Journal of Chemical Physics*, *119*(16), 8229–8234.

Grünwald, P. D. (2007). *The Minimum Description Length Principle*. The MIT Press.

Handl, J., Kell, D. B., & Knowles, J. (2007). Multiobjective optimization in bioinformatics and computational biology. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *4*(2), 279–292.

Hardy, S. & Robillard, P. N. (2004). Modeling and simulation of molecular biology systems using Petri nets: Modeling goals of various approaches. *Journal of Bioinformatics and Computational Biology*, *2*(4), 619–637.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning. Data Mining, Inference, and Prediction* (2nd ed.). Springer New York.

Heiner, M., Gilbert, D., & Donaldson, R. (2008). Formal Methods for Computational Systems Biology: 8th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2008 Bertinoro, Italy, June 2-7, 2008 Advanced Lectures. In M. Bernardo, P. Degano, & G. Zavattaro (Eds.), (Chap. Petri Nets for Systems and Synthetic Biology, pp. 215–264). Springer Berlin Heidelberg.

Henriques, D., Rocha, M., Saez-Rodriguez, J., & Banga, J. R. (2015). Reverse engineering of logic-based differential equation models using a mixed-integer dynamic optimization approach. *Bioinformatics*, *31*(18), 2999–3007.

Higuera, C., Villaverde, A. F., Banga, J. R., Ross, J., & Morán, F. (2012). Multi-criteria optimization of regulation in metabolic networks. *PLoS ONE*, *7*(7), 1–10.

Hindmarsh, A. C., Brown, P. N., Grant, K. E., Lee, S. L., Serban, R., Shumaker, D. E., & Woodward, C. S. (2005). SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers. *ACM Transactions of Matemathical Software*, *31*(3), 363–396.

Hodgkin, A. L. & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, *117*(4), 500–544.

Ideker, T., Galitski, T., & Hood, L. (2001). A new approach to decoding life: Systems Biology. *Annual Review of Genomics and Human Genetics*, *2*(1), 343–372.

Iyengar, R., Zhao, S., Chung, S.-W., Mager, D. E., & Gallo, J. M. (2012). Merging systems biology with pharmacodynamics. *Science Translational Medicine*, *4*(126), 1–7.

Jaqaman, K. & Danuser, G. (2006). Linking data to models: data regression. *Nature Reviews Molecular Cell Biology*, *7*(11), 813–819.

Jeong, H., Mason, S. P., Barabasi, A.-L., & Oltvai, Z. N. (2001). Lethality and centrality in protein networks. *Nature*, *411*(6833), 41–42.

Jinek, M., Chylinski, K., Fonfara, I., Hauer, M., Doudna, J. A., & Charpentier, E. (2012). A Programmable Dual-RNA–Guided DNA Endonuclease in Adaptive Bacterial Immunity. *Science*, *337*(6096), 816–821.

Kaltenbach, H.-M., Dimopoulos, S., & Stelling, J. (2009). Systems analysis of cellular networks under uncertainty. *FEBS Letters*, *583*(24), 3923–3930.

Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, *22*(3), 437–467.

Kaznessis, Y. N. (2007). Models for synthetic biology. *BMC Systems Biology*, *1*(1), 1–47.

Kelwick, R., MacDonald, J. T., Webb, A. J., & Freemont, P. (2014). Developments in the tools and methodologies of synthetic biology. *Frontiers in Bioengineering and Biotechnology*, *2*(60).

Khalil, A. S. & Collins, J. J. (2010). Synthetic Biology: Applications Come of Age. *Nature reviews. Genetics*, *11*(5), 367–379.

Kirk, P., Silk, D., & Stumpf, M. P. H. (2016). Uncertainty in Biology: A Computational Modeling Approach. In L. Geris & D. Gomez-Cabrero (Eds.), (Chap. Reverse Engineering Under Uncertainty, pp. 15–32). Springer International Publishing.

Kirk, P., Thorne, T., & Stumpf, M. P. H. (2013). Model selection in systems and synthetic biology. *Current Opinion in Biotechnology*, *24*(4), 767–774.

Kitano, H. (2000). Perspectives on Systems Biology. *New Generation Computing*, *18*(3), 199–216.

Kitano, H. (2002a). Computational systems biology. *Nature*, *420*, 206–210.

Kitano, H. (2002b). Systems biology: A brief overview. *Science*, *295*(5560), 1662–1664.

Kitano, H. (2007). Towards a theory of biological robustness. *Molecular Systems Biology*, *3*, 137–137.

Klamt, S., Saez-Rodriguez, J., Lindquist, J. A., Simeoni, L., & Gilles, E. D. (2006). A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC Bioinformatics*, *7*, 56–56.

Komorowski, M., Costa, M. J., Rand, D. A., & Stumpf, M. P. H. (2011). Sensitivity, robustness, and identifiability in stochastic chemical kinetics models. *Proceedings of the National Academy of Sciences*, *108*(21), 8645–8650.

Küffner, R., Zimmer, R., & Lengauer, T. (2000). Pathway analysis in metabolic databases via differential metabolic display (DMD). *Bioinformatics*, *16*(9), 825–836.

Langley, P., Sanchez, J., Todorovski, L., & Džeroski, S. (2002). Inducing Process Models from Continuous Data. In *Proceedings of the Nineteenth International Conference on Machine Learning*. Morgan Kaufmann.

Langley, P., Simon, H. A., Bradshaw, G. L., & Zytkow, J. M. (1992). *Scientific Discovery: Computational Explorations of the Creative Processes*. MIT Press.

Le Novère, N., Bornstein, B., Broicher, A., Courtot, M., Donizelli, M., Dharuri, H., ... Hucka, M. (2006). BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Research*, *34*(Database issue), D689–D691.

Le Novére, N., Finney, A., Hucka, M., Bhalla, U. S., Campagne, F., Collado-Vides, J., ... Wanner, B. L. (2005). Minimum information requested in the annotation of biochemical models (MIRIAM). *Nature Biotechnology*, *23*(12), 1509–1515.

Lecca, P., Laurenzi, I., & Jordan, F. (2013). *Deterministic Versus Stochastic Modelling in Biochemistry and Systems Biology*. Woodhead Publishing.

Lemons, N. W., Hu, B., & Hlavacek, W. S. (2011). Hierarchical graphs for rule-based modeling of biochemical systems. *BMC Bioinformatics*, *12*(1), 1–13.

Ljung, L. (1999). *System Identification: Theory for the User* (2nd ed.). Prentice Hall PTR.

Lloyd, C. M., Lawson, J. R., Hunter, P. J., & Nielsen, P. F. (2008). The CellML Model Repository. *Bioinformatics*, *24*(18), 2122–2123.

Lu, L. & Anderson-Cook, C. M. (2013). Adapting the hypervolume quality indicator to quantify trade-offs and search efficiency for multiple criteria decision making using pareto fronts. *Quality and Reliability Engineering International*, *29*(8), 1117–1133.

Machado, D., Costa, R., Rocha, M., Ferreira, E., Tidor, B., & Rocha, I. (2011). Modeling formalisms in systems biology. *AMB Express*, *1*(1), 1–14.

Marchisio, M., Colaiacovo, M., Whitehead, E., & Stelling, J. (2013). Modular, rule-based modeling for the design of eukaryotic synthetic gene circuits. *BMC Systems Biology*, *7*(1), 1–42.

Marchisio, M. & Stelling, J. (2009). Computational design tools for synthetic biology. *Current Opinion in Biotechnology*, *20*(4), 479–485.

Marguet, P., Balagadde, F., Tan, C., & You, L. (2007). Biology by design: reduction and synthesis of cellular components and behaviour. *Journal of The Royal Society Interface*, *4*(15), 607–623.

McAdams, H. H. & Arkin, A. (1997). Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences*, *94*(3), 814–819.

McQuarrie, D. A. (1967). Stochastic approach to chemical kinetics. *Journal of Applied Probability*, *4*(3), 413–478.

Medina, M. Á. (2012). Systems biology for molecular life sciences and its impact in biomedicine. *Cellular and Molecular Life Sciences*, *70*(6), 1035–1053.

Mendes, P. & Kell, D. (1998). Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics*, *14*(10), 869–883.

Mesarović, M. D. (Ed.). (1968). *Systems Theory and Biology: Proceedings of the III Systems Symposium at Case Institute of Technology*. Springer Berlin Heidelberg.

Milner, R. (1980). *A Calculus of Communicating Systems*. Lecture Notes in Computer Science. Springer-Verlag.

Milner, R., Parrow, J., & Walker, D. (1992). A Calculus of Mobile Processes, I and II. *Information and Computation*, *100*(1), 1–77.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

Moles, C., Mendes, P., & Banga, J. (2003). Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome Research*, *13*, 2467–2474.

Murray, J. D. (1993). *Mathematical Biology* (2nd ed.). Biomathematics. Springer Berlin Heidelberg.

Newell, A. & Simon, H. A. (1972). *Human problem solving*. Prentice-Hall.

Nielsen, A. A. K., Der, B. S., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalski, E. A., ... Voigt, C. A. (2016). Genetic circuit design automation. *Science*, *352*(6281).

Nielsen, E. I. & Friberg, L. E. (2013). Pharmacokinetic-pharmacodynamic modeling of antibacterial drugs. *Pharmacological Reviews*, *65*(3), 1053–1090.

Novak, B. & Tyson, J. J. (2008). Design principles of biochemical oscillators. *Nature Reviews Molecular Cell Biology*, *9*(12), 981–991.

Otero-Muras, I. & Banga, J. R. (2014). Multicriteria global optimization for biocircuit design. *BMC Systems Biology*, *8*(1), 1–12.

Pedersen, M. & Phillips, A. (2009). Towards programming languages for genetic engineering of living cells. *Journal of The Royal Society Interface*, *6*(Suppl 4), S437–S450.

Penfold, C. A. & Wild, D. L. (2011). How to infer gene networks from expression profiles, revisited. *Interface Focus*, 1–14.

Petri, C. A. (1962). *Kommunikation mit Automaten* (Doctoral dissertation, Rheinisch-Wwestfälischen Instituts für Instrumentelle Mathematik an der Universität Bonn).

Priami, C. & Quaglia, P. (2004). Modelling the dynamics of biosystems. *Briefings in Bioinformatics*, *5*(3), 259–269.

Priami, C. & Quaglia, P. (2005). Beta binders for biological interactions. In *Computational methods in systems biology* (pp. 20–33). Springer Berlin Heidelberg.

Priami, C., Regev, A., Shapiro, E., & Silverman, W. (2001). Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, *80*(1), 25–31.

Price, K., Storn, R. M., & Lampinen, J. A. (2005). *Differential Evolution: A Practical Approach to Global Optimization*. Springer Berlin Heidelberg.

Purnick, P. E. & Weiss, R. (2009). The second wave of synthetic biology: from modules to systems. *Nature Reviews. Molecular Cell Biology*, *10*, 410–422.

Ramsey, S., Orrell, D., & Bolouri, H. (2005). Dizzy: Stochastic simulation of large-scale genetic regulatory networks. *Journal of Bioinformatics and Computational Biology*, *3*(2), 415–436.

Raue, A., Karlsson, J., Saccomani, M. P., Jirstrand, M., & Timmer, J. (2014). Comparison of approaches for parameter identifiability analysis of biological systems. *Bioinformatics*, *30*(10), 1440–1448.

Reddy, V. N., Liebman, M. N., & Mavrovouniotis, M. L. (1996). Qualitative analysis of biochemical reaction systems. *Computers in Biology and Medicine*, *26*(1), 9–24.

Regev, A., Silverman, W., & Shapiro, E. (2001). Representation and simulation of biochemical processes using the π-calculus process algebra. In *Pacific Symposium on Biocomputing* (Vol. 6, pp. 459–470).

Rodrigo, G., Carrera, J., & Jaramillo, A. (2007). Genetdes: automatic design of transcriptional networks. *Bioinformatics*, *23*(14), 1857–1858.

Rodrigo, G. & Jaramillo, A. (2013). AutoBioCAD: Full Biodesign Automation of Genetic Circuits. *ACS Synthetic Biology*, *2*(5), 230–236.

Russell, S. & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach* (3rd ed.). Prentice Hall.

Saez-Rodriguez, J., Alexopoulos, L. G., Epperlein, J., Samaga, R., Lauffenburger, D. A., Klamt, S., & Sorger, P. K. (2009). Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction. *Molecular Systems Biology*, *5*(1), 1–19.

Samoilov, M., Plyasunov, S., & Arkin, A. P. (2005). Stochastic amplification and signaling in enzymatic futile cycles through noise-induced bistability with oscillations. *Proceedings of the National Academy of Sciences*, *102*(7), 2310–2315.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464.

Sendin, J., Exler, O., & Banga, J. (2010). Multi-objective mixed integer strategy for the optimisation of biological networks. *IET Systems Biology*, *4*(3), 236–248.

Shinar, G. & Feinberg, M. (2010). Structural sources of robustness in biochemical reaction networks. *Science*, *327*(5971), 1389–1391.

Shmulevich, I., Dougherty, E. R., Kim, S., & Zhang, W. (2002). Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, *18*(2), 261–274.

Silk, D., Kirk, P. D. W., Barnes, C. P., Toni, T., Rose, A., Moon, S., ... Stumpf, M. P. H. (2011). Designing attractive models via automated identification of chaotic and oscillatory dynamical regimes. *Nature Communications*, *2*.

Simidjievski, N., Todorovski, L., & Džeroski, S. (2015). Learning ensembles of population dynamics models and their application to modelling aquatic ecosystems. *Ecological Modelling*, *306*, 305–317.

Simidjievski, N., Todorovski, L., & Džeroski, S. (2016). Modeling dynamic systems with efficient ensembles of process-based models. *PLoS One*, *11*(4), 1–27.

Škerjanec, M., Atanasova, N., Čerepnalkoski, D., Džeroski, S., & Kompare, B. (2014). Development of a knowledge library for automated watershed modeling. *Environmental Modelling & Software*, *54*, 60–72.

Stelling, J., Sauer, U., Szallasi, Z., Doyle III, F. J., & Doyle, J. (2004). Robustness of cellular functions. *Cell*, *118*(6), 675–685.

Sun, J., Garibaldi, J., & Hodgman, C. (2012). Parameter estimation using metaheuristics in systems biology: A comprehensive review. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *9*(1), 185–202.

Tanevski, J., Simidjievski, N., & Džeroski, S. (2012). Biocircuit design with equation discovery. In *Proceedings of the ECML-PKDD 2012 Workshop on Learning and Discovery in Symbolic Systems Biology* (pp. 2–16). University of Bristol. Bristol, UK.

Tanevski, J., Todorovski, L., & Džeroski, S. (2013). Automated modeling of Rab5-Rab7 conversion in endocytosis. In *Proceedings of the 5th Jožef Stefan International Postgraduate School Students' Conference* (pp. 209–218). Jožef Stefan International Postgraduate School. Ljubljana, Slovenia.

Tanevski, J., Todorovski, L., & Džeroski, S. (2016a). Learning stochastic process-based models of dynamical systems from knowledge and data. *BMC Systems Biology*, *10*(1), 1–30.

Tanevski, J., Todorovski, L., & Džeroski, S. (2016b). Process-based design of dynamical biological systems. *Scientific Reports*. Under review.

Tanevski, J., Todorovski, L., Kalaidzidis, Y., & Džeroski, S. (2013). Inductive process modeling of Rab5-Rab7 conversion in endocytosis. In *Proceeedings of the 16th International Conference on Discovery Science* (pp. 265–280). Springer Berlin.

Tanevski, J., Todorovski, L., Kalaidzidis, Y., & Džeroski, S. (2015). Domain-specific model selection for structural identification of the Rab5-Rab7 dynamics in endocytosis. *BMC Systems Biology*, *9*(1), 1–31.

Tashkova, K., Korošec, P., Šilc, J., Todorovski, L., & Džeroski, S. (2011). Parameter estimation with bio-inspired meta-heuristic optimization: Modeling the dynamics of endocytosis. *BMC Systems Biology*, *5*(1), 1–26.

Todorovski, L., Bridewell, W., Shiran, O., & Langley, P. (2005). Inducing hierarchical process models in dynamic domains. In *Proceedings of the Twentieth National Conference on Artificial Intelligence* (pp. 892–897). AAAI Press.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., & Stumpf, M. (2009). Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, *6*, 187–202.

Turner, T., Schnell, S., & Burrage, K. (2004). Stochastic approaches for modelling in vivo reactions. *Computational Biology and Chemistry*, *28*(3), 165–178.

Tyson, J. J. (2007). Bringing cartoons to life. *Nature*, *445*(7130), 823–823.

Tyson, J. J. & Novak, B. (2010). Functional Motifs in Biochemical Reaction Networks. *Annual Review of Physical Chemistry*, *61*(1), 219–240.

Umbarger, H. E. & Brown, B. (1957). Threonine deamination in Escherichia coli II. : Evidence for Two l-Threonine Deaminases. *Journal of Bacteriology*, *73*(1), 105–112.

Villaverde, A. F. & Banga, J. R. (2013). Reverse engineering and identification in systems biology: strategies, perspectives and challenges. *Journal of The Royal Society Interface*, *11*(91).

von Bertalanffy, L. (1968). *General System Theory: Foundations, Development, Applications*. George Braziller.

Wahl, S., Haunschild, M., Oldiges, M., & Wiechert, W. (2006). Unravelling the regulatory structure of biochemical networks using stimulus response experiments and large-scale model selection. *IEE Proceedings - Systems Biology*, *153*, 275–285.

Wang, R.-S., Saadatpour, A., & Albert, R. (2012). Boolean modeling in systems biology: an overview of methodology and applications. *Physical Biology*, *9*(5), 1–14.

Westerhoff, H. V. & Palsson, B. O. (2004). The evolution of molecular biology into systems biology. *Nature Biotechnology*, *22*(10), 1249–1252.

Wilkinson, D. J. (2006). *Stochastic Modelling for Systems Biology*. CRC Press.

Wilkinson, D. J. (2007). Bayesian methods in bioinformatics and computational systems biology. *Briefings in Bioinformatics*, *8*(2), 109–116.

Wilson-Kanamori, J., Danos, V., Thomson, T., & Honorato-Zimmer, R. (2015). Computational methods in synthetic biology. In A. M. Marchisio (Ed.), (Chap. Kappa Rule-Based Modeling in Synthetic Biology, pp. 105–135). Springer New York.

Wolkenhauer, O. (2001). Systems biology: The reincarnation of systems theory applied in biology? *Briefings in Bioinformatics*, *2*(3), 258–270.

Wolkenhauer, O. (2014). Why Model? *Frontiers in Physiology*, *5*(21).

Wolkenhauer, O., Auffray, C., Jaster, R., Steinhoff, G., & Dammann, O. (2013). The road from systems biology to systems medicine. *Pediatric Research*, *73*(4-2), 502–507.

Yaman, F., Bhatia, S., Adler, A., Densmore, D., & Beal, J. (2012). Automated Selection of Synthetic Biology Parts for Genetic Regulatory Networks. *ACS Synthetic Biology*, *1*(8), 332–344.

Yates, R. A. & Pardee, A. B. (1957). Control by uracil of formation of enzymes required for orotate synthesis. *Journal of Biological Chemistry*, *227*(2), 677–692.

Zevedei-Oancea, I. & Schuster, S. (2003). Topological analysis of metabolic networks based on Petri net theory. *In Silico Biology*, *3*(3), 323–345.

Zhou, T., Zhang, J., Yuan, Z., & Chen, L. (2008). Synchronization of genetic oscillators. *Chaos*, *18*(3), 1–20.

Zitzler, E., Knowles, J., & Thiele, L. (2008). Multiobjective Optimization: Interactive and Evolutionary Approaches. In J. Branke, K. Deb, K. Miettinen, & R. Słowiński (Eds.), (Chap. Quality Assessment of Pareto Set Approximations, pp. 373–404). Springer Berlin Heidelberg.

Bibliography

Publications Related to the Thesis

Journal Articles

Tanevski, J., Todorovski, L., & Džeroski, S. (2016a). Learning stochastic process-based models of dynamical systems from knowledge and data. *BMC Systems Biology*, *10*(1), 1–30.

Tanevski, J., Todorovski, L., & Džeroski, S. (2016b). Process-based design of dynamical biological systems. *Scientific Reports*. Under review.

Tanevski, J., Todorovski, L., Kalaidzidis, Y., & Džeroski, S. (2015). Domain-specific model selection for structural identification of the Rab5-Rab7 dynamics in endocytosis. *BMC Systems Biology*, *9*(1), 1–31.

Conference Papers

Tanevski, J., Simidjievski, N., & Džeroski, S. (2012). Biocircuit design with equation discovery. In *Proceedings of the ECML-PKDD 2012 Workshop on Learning and Discovery in Symbolic Systems Biology* (pp. 2–16). University of Bristol. Bristol, UK.

Tanevski, J., Todorovski, L., & Džeroski, S. (2013). Automated modeling of Rab5-Rab7 conversion in endocytosis. In *Proceedings of the 5th Jožef Stefan International Postgraduate School Students' Conference* (pp. 209–218). Jožef Stefan International Postgraduate School. Ljubljana, Slovenia.

Tanevski, J., Todorovski, L., Kalaidzidis, Y., & Džeroski, S. (2013b). Inductive process modeling of Rab5-Rab7 conversion in endocytosis. In *Proceeedings of the 16th International Conference on Discovery Science* (pp. 265–280). Springer Berlin.

Conference Abstracts

Tanevski, J., Todorovski, L., & Džeroski, S. (2014). Automated process-based modeling and design of dynamic biological systems. In *Human Brain Project : 1st HBP School, Abstract Collection - Students* (p. 26). Alpbach, Austria.

Tanevski, J., Todorovski, L., & Džeroski, S. (2015). Process-based design of synthetic biological systems. In *Book of abstracts - 10th CFGBC Symposium with ISBE and CASyM workshops From functional genomics to systems biology and systems medicine & Hands-on tutorial systems biology/medicine* (p. 33). Ljubljana, Slovenia.

Tanevski, J., Todorovski, L., Kalaidzidis, Y., & Džeroski, S. (2013a). Discovering a model of Rab5-Rab7 conversion in endocytosis. In *Book of abstracts - ICSB 2013, The 14th International Conference on Systems Biology* (p. 139). Copenhagen, Denmark.

Biography

Jovan Tanevski was born on September 28, 1987 in Skopje, Macedonia. In 2006, he enrolled in a four year first cycle Bachelor of Science program in the area of Informatics and Computer Engineering at the Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University in Skopje, which he successfully finished in 2010. The same year he enrolled in a second cycle program in Bioinformatics at the same Faculty, which he successfully finished and was awarded with a Master of Science degree in 2011. During his undergraduate studies he held a state scholarship for talented students awarded by the Ministry of Education and Science of Macedonia.

In the fall of 2011, he started his PhD studies in Information and Communication Technologies at the Jožef Stefan International Postgraduate School in Ljubljana, Slovenia, under the supervision of Professor Sašo Džeroski and co-supervision of Professor Ljupčo Todorovski. He held a doctoral studies scholarship awarded by the Slovene Human Resources and Scholarship Fund and a research scholarship awarded by the Department of Knowledge Technologies at the Jožef Stefan Institute in Ljubljana, Slovenia. During his studies he took part in several EU funded projects PHAGOSYS (Systems biology of phagosome formation and maturation - modulation by intracellular pathogens), SUMO (Super modeling by combining imperfect models) and HBP (The Human Brain Project).

His research is in the field of computational biology and machine learning. His current research is concerned with the development of approaches to computational scientific discovery and their application to problems of modeling and design of dynamical systems from the domains of systems and synthetic biology. He has published his work in several journals and has presented it at several international conferences and workshops.