Chapter 1

COMPUTATIONAL APPROACHES IN SYNTHETIC AND SYSTEMS BIOLOGY

Miha Moškon*[*], *Jure Bordon, Miha Mraz, Nikolaj Zimic and Mattia Petroni
University of Ljubljana,
Faculty of Computer and Information Science, Ljubljana, Slovenia

Abstract

Computational biology is an emerging scientific field which employs computational methods in the study of various biological systems. This chapter presents a review of methods that have been introduced to the fields of synthetic and systems biology in recent years. Approaches presented mainly rely on the establishment of computational models. These models allow us to observe the behaviour of a certain biological system in a given environment. Exact kinetic data that describe underlying dynamics are usually necessary to establish accurate computational models. Kinetic data are on the other hand hard or even impossible to obtain experimentally in some cases. Parameter estimation techniques that also rely on computational approaches can be used to accurately evaluate missing kinetic data. With the establishment of computational models, computational analyses can be conducted, such as performance, robustness, sensitivity or stability analysis. These techniques can be used further on to reduce the amount of experimental work and enable straightforward design of novel biological systems in the context of synthetic biology.

Keywords: Computational modelling, computer simulation, network inference, parameter estimation techniques, performance evaluation, sensitivity analysis, stability analysis, computational design

1. Introduction

Use of engineering tools that rely on various computational approaches is more and more common in the field of systems biology [1]. Being a combination of chemistry, biology and engineering [2], these approaches gained even larger role in the field of synthetic

[*] E-mail address: miha.moskon@fri.uni-lj.si

biology. Computational approaches used in these two disciplines are mainly based on the establishment of various computational models. Given a predicted environment and initial conditions, these models are capable to mimic the behaviour of an arbitrary biological system. Regarding their design, two main approaches exist, i.e., *top-down* and *bottom-up* approach. Top-down design approach is a basis for the establishment of computational models using experimentally measured data from existent biological systems (usually obtained using DNA microarrays) [3, 4, 5]. This approach is also known as *network inference* and is based on reverse engineering of the computational model from experimentally measured data. The complementary, i.e. bottom-up approach, is on the other hand used to simplify the design of novel biological systems with predefined functionalities. These models are usually constructed with the association of biological modelling primitives, which can be obtained with the top-down approach [6].

The accuracy of computational models is tightly correlated with the accuracy of parameter values, which define the dynamic and behavioural properties of observed biological systems. These parameters are sometimes very hard or even impossible to determine accurately. In order to bypass this problem, several parameter estimation techniques can be used [7, 8] for bottom-up modelling. There are also several top-down modelling approaches that aim to reconstruct an existent biological system from experimental data without trying to estimate the parameter values (for example with the employment of fuzzy logic methods [9]).

Established models that reflect a satisfactory accuracy can be further used to get the specific insights into the behaviour of a certain biological system. When we are dealing with reverse engineered models, these systems already exists. The models are established to perform various analyses and deduct behavioural properties of the system which are hard or even impossible to obtain experimentally [10]. Moreover, modelling results can be used to optimize the behaviour of observed biological system with certain modifications [11]. On the other hand, bottom-up models can be applied to the computational design of biological systems, thus reducing the quantity of experimental work. Several computational approaches for automatic design already exist, such as heuristic optimization techniques which seek for optimal solutions regarding the given objective functions [12]. These functions can be established in different ways, e.g. with the evaluation of specific metrics that are able to quantify the behaviour of biological system [13].

In this chapter we review the computational approaches in the field of systems and synthetic biology. We especially focus on *gene regulatory networks* (GRNs), but most of the presented concepts can also be applied in the same way on *signal transduction* and *metabolic networks*. First we present state of the art modelling approaches in the field. We briefly review parameter estimation techniques that allow us to bridge the problem of parameter evaluation. Further on we present advanced model analysis techniques for performance, robustness and stability analysis of biological systems. In the end we present computational methods for computer-aided design, automatic design and optimization of biological systems.

2. Computational Modelling Approaches

The development in the field of computational modelling of biological systems goes in hand with the progress of systems and synthetic biology. One can select among various modelling approaches which were introduced recently, each with its own benefits and its own potential applications. The decision for an appropriate modelling strategy is vital and depends on several factors, i.e. the complexity of observed system, the availability of its kinetic data and the type of information the user would like to obtain. Several classifications exist with the aim to organize the modelling approaches in different groups and consequently ease the decision of choosing the appropriate ones. The first distinction can be made among *static* and *dynamic* models. Static models do not incorporate time component and only yield topologies or qualitative networks of gene interactions. Dynamic gene network models on the other hand describe the changes of gene expression in time [14]. Both of these two groups can be further classified in *continuous* and *discrete* models. Continuous models observe the abundances of chemical species as non-negative real values with certain upper limits. *Discrete models* however observe the system on the molecular level and therefore describe its current state with discrete quantity of molecular numbers. Further distinction among dynamic models classifies them in *qualitative models* (also referred to as *logical models* [15]), which are relatively simple and easy to infer even from imprecise data [16], but can only answer qualitative questions, and *quantitative models*, which are able to reflect higher accuracy, but on the account of their complexity and demand accurate values of kinetic data. These models are usually inappropriate for exact modelling of more complex biological systems, but are on the other hand more informative about the system's properties when used in accordance with their limitations. Quantitative models can be further divided among *deterministic models* which are only able to capture average response within a population of identical cells or average response in a single cell over a long time period and *stochastic models* which also consider the probabilistic nature of chemical reactions and can thus be used for the modelling of population heterogeneity and effects of intrinsic noise [17]. Both modelling types yield similar results when we are dealing with large systems (high numbers of molecules and large cell volumes) with fast promoter kinetics [18]. Considerable attention has been devoted to the concept of stochasticity in GRNs since the advent of synthetic biology [17, 19]. More details about stochastic approaches in synthetic biology can be found in [20]. Another distinction of modelling approaches can be made regarding the *level of details* (granularity) they consider, i.e. parts, topology, control logic [21] or coarse, average and fine grained models [22], where fine grained models can mostly be applied only to small systems and coarse grained models to large and complex systems.

Here we will present few examples of different modelling techniques belonging to different groups listed above (see Tab. 1). Since the field of computational modelling has expanded rapidly in last years, we will have to omit some of the modelling methods from this review.

Table 1. Different modelling techniques classified in groups described within the main text, where DG denotes Directed Graphs, (D)BYN (Dynamic) Bayesian Networks, (P)BN (Probabilistic) Boolean Networks, ODE Ordinary Differential Equations, TM Thermodynamic Models, CME Chemical Master Equations and PN Petri Nets. ST denotes static and DN dynamic models, CT continuous and DS discrete models, QL qualitative and QN quantitative models, DT deterministic and ST stochastic models, LG large and SM small networks. Tab. 1 is a modification of the table presenting the summary of properties of different modelling formalisms in [22].

	ST/DN	CT/DS,	QL/QN,	DT/ST,	LG/SM
DG	ST	DS	QL	DT	LG
BYN	ST	DS	QN	ST	LG
DBYN	DN	DS	QN	ST	LG
BN	DN	DS	QL	DT/ST	LG
PBN	DN	DS	QL	ST	LG
ODE	DN	CT	QN	DT/ST	SM
TM	ST	CT	QN	DT	SM
CME	DN	DS	QN	ST	SM
PN	ST/DN	CT/DS	QL/QN	DT/ST	SM

2.1. Directed Graphs

Directed graphs (DGs) can be regarded as the most basic presentation of biological regulatory networks. Simplicity of this approach makes it applicable to the modelling of large biological systems (i.e. for several magnitudes larger than with other modelling approaches), but on the account of its static nature. Various graph operations can be performed on these models in order to make relevant predictions about the structure and behaviour of observed regulatory system, e.g. search for paths among two genes or investigation of existence of cycles that may point at possible feedback relations [22]. A review of such approaches is presented in [23].

Mathematical description of DG is presented with two sets, i.e. set of vertices V and set of edges E. When dealing with gene regulatory networks, vertices usually refer to genes (or chemical species that are results of their expression) and edges to interactions among them. Each edge is defined as $\langle y,x,r \rangle$, where y is the head and x the tail of the edge. This can be interpreted as x regulates the expression of y. Regulation type is defined in r, i.e. y can be activated (+) or inhibited (–) by x (for graphical presentation see Fig. 1). Basic graph presentation can be extended with hypergraphs, in which each vertex is defined as $\langle y,X,R \rangle$. Here X corresponds to the vertices that cooperatively regulate y and R to the types of regulation for each one of them.

Deduction of DGs can be made on the basis of existent knowledge or with the application of different automated inference procedures. Various *clustering algorithms* can be used, which are motivated by the fact that two genes may regulate each other or may be coregulated by a third gene if they both reflect similar expression patterns [22]. Many novel alternative inference approaches were reported recently. We will briefly describe a solution which combines *artificial neural-networks* with *genetic algorithms, particle swarm*

$$x \xrightarrow{+} y \qquad x \longrightarrow y \qquad x \xrightarrow{-} y \qquad x \longrightarrow\!\!\!\!\!\dashv y$$

(a) (b) (c) (d)

Figure 1. Symbols used in graphical presentations of directed graph models. Figs. (a) and (b) present a situation where tail vertex activates the expression of head vertex. Figs. (c) and (d) present a situation where tail vertex, inhibits the expression of head vertex. Fig. (a) is compatible with Fig. (c), and Fig. (b) is compatible with Fig. (d).

optimization (PSO) and *fuzzy logic* into a *multi-layer evolutionary trained neuro-fuzzy recurrent network* (ENFRN) [24].

ENFRN is able to successfully extract the regulatory interactions from (noisy) data obtained with gene expression profiling. Evolutionary training of artificial neural network based on the PSO automatically generates an adaptive number of temporal fuzzy rules that describe the relationships between the input and the output genes. The training is performed on the basis of gene expression profiles in two phases: (a) the structure (topology) learning, in which fuzzy IF-THEN rules are generated and feedback configuration is established, and (b) parameter learning, in which free parameters which define the established fuzzy rules are tuned. Trained ENFRN structure is able to determine if specified input regulates the output, the kind of regulation and also provides a score, that specifies the confidence in retrieved relation.

2.2. Boolean Networks

Boolean networks (BNs) are the most basic dynamic extension of DG modelling approach in which a single gene is considered to be in one of two possible states, i.e. *on* or *off*. Direct regulatory interactions among genes are modelled as Boolean functions [22]. Let's presume, that the state of the system in time t is defined with the vector $\mathbf{x}(t) = [x_1(t), x_2(t),...,x_N(t)]^T$, where $x_i(t)$ is a boolean variable corresponding to activity (presence) or inactivity (absence) of gene i (or a chemical species that results in its expression) in time t. Model dynamics is determined with a set of Boolean functions $B = \{b_1,b_2,...,b_N\}$, where $x_i(t+1) = b_i(\mathbf{x}(t))$. BNs can be directly projected to DG models. Direct projection cannot be made in both directions, since BNs contain additional information to directed and signed network diagrams [16]. For example of such projection see Fig. 2. While the number of possible states of the system is limited, i.e. 2^N possible states for N Boolean variables, all initial states may eventually reach a *steady state* (point attractor) or a *state cycle* (dynamic attractor). All possible system trajectories from an initial configuration can be analysed with a *state transition graph*. However, these graphs can be constructed only for small systems or only for limited amount of initial configurations.

BNs do not consider intermediate levels of gene expression which results in possible information loss. In basic BNs, transitions between states, i.e. from $\mathbf{x}(t)$ to $\mathbf{x}(t+1)$, occur synchronously, meaning that all state changes happen simultaneously and in deterministic regime. Simplicity of this approach, its qualitative features and parameter-free nature makes it applicable for inference, dynamic modelling and efficient analysis of large

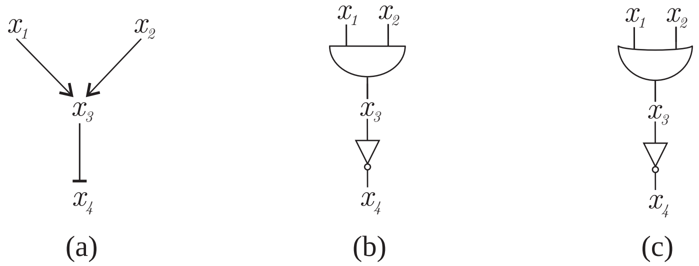

Figure 2. An example of directed graph with two possible projections to Boolean networks, i.e. $B = \{b_3(x_1,x_2) = x_1$ OR $x_2, b_4(x_3) =$ NOT $x_3\}$ in Fig. (b) and $B = \{b_3(x_1,x_2) = x_1$ AND $x_2, b_4(x_3) =$ NOT $x_3\}$ in Fig. (c).

scale regulatory networks [16]. Several simplifications can make this approach inappropriate for general modelling of biological systems [22]. However, BNs can be extended in some degree. *Probabilistic Boolean Networks* (PBNs) are able to incorporate the stochastic nature of chemical reactions [25]. PBN annotation extends a set of functions B into $B = \{B_1,B_2,...,B_N\}$, where B_i is a set of Boolean functions $\{f_1^{(i)}, f_2^{(i)},...,f_l^{(i)}\}$. It contains all possible functions that may define the state of Boolean variable x_i. Each of these functions can be chosen to update the state vector in each iteration according to its probability. Moreover, basic and probabilistic BNs can be extended with the introduction of non-synchronous transition schemes, in which the states of the nodes are updated in dependence on the time-scales of individual biological events regarding their underlying regulatory processes. These schemes can be further divided to deterministic and stochastic ones. While time-scales are fixed for each node in deterministic schemes, stochastic schemes randomly select a node to be updated or update all nodes in random order [16].

2.3. Bayesian Networks

Bayesian networks (BYNs) (also referred to as *probabilistic networks*) are a combination of probability calculus, i.e. they incorporate the stochastic nature of gene regulation, and graph theory [3]. BYNs describe the regulatory interaction in regulatory networks with *directed acyclic graph* presentation, where each vertex has a similar interpretation as in *directed graph* presentation (see section 2.1). Random variables that describe the expression levels correspond to each vertex. Conditional distribution is defined for each random variable, i.e. $p(X_i|parents(X_i))$, where i is the vertex, X_i random variable, that describes its expression level and $parents(X_i)$ random variables describing direct regulators of i. BYN description is therefore defined with the directed acyclic graph G and the conditional probability distributions Θ, which defines local conditional probability distributions for each vertex in graph [22]. An example of a BYN graph is presented in Fig. 3(a). Described approach is also known as *static Bayesian network* modelling. Its main disadvantages are in the fact that it is unable to capture feedback loops in gene regulation and can only be used to analyse dependencies between genes. It is therefore not suitable for simulation of system's dynamics.

Dynamic Bayesian networks (DBYNs) (also referred to as *dynamic probabilistic networks*) are a temporal extension of BYNs, that separate input nodes from output nodes. This modelling approach is therefore suitable for simulating the time-dependant dynamics of the system and is able to describe regulatory feedback loops with directed acyclic graphs. DBYN can be defined with a pair $(B_0{,}B_1)$, where $B_0 = (G_0{,}\Theta_0)$ is initial BYN and $B_1 = (G_1{,}\Theta_1)$ a transition BYN which specifies transition probabilities, i.e. $P(\mathbf{X(t)}|\mathbf{X(t-1)})$ [26]. An example of a transition graph is presented in Fig. 3(b).

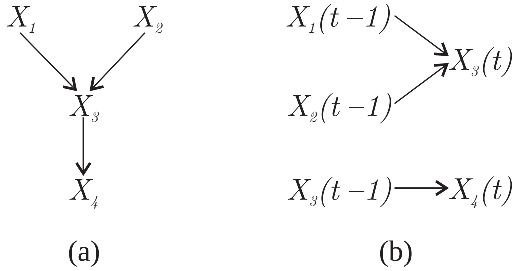

Figure 3. Directed acyclic graphs presenting an example of static Bayesian network (a) and a transition graph of the dynamic Bayesian network for the same regulatory network (b).

The main advantages of BYNs are in their capabilities to model large systems, but on the account of their coarse granularity. These networks are also easy to reconstruct, even if incomplete knowledge about the system is available or from the combination of different types of data. BYNs as such have many features which are suitable for the modelling of regulatory networks [22, 3]. While DBYN solve basic limitations of static BYNs, computational costs drastically increase when inferring these models from experimental data and are therefore not suitable for large regulatory networks.

2.4. Ordinary Differential Equations

Ordinary differential equation (ODE) models consider the concentrations of observed chemical species (such as mRNAs and proteins) as time-dependant variables [22]. Although ODE based models are mainly deterministic, their extensions with the introduction of Poisson random variables has also been reported [27]. These models mainly rely on mass action kinetics or on phenomenological representations of reaction mechanisms [17]. Let's presume, that state of the system in time t is defined as $\mathbf{x}(t) = [x_1(t), x_2(t),...,x_N(t)]^T$, where $x_i(t)$ is current abundance of chemical species i as non-negative real value. ODE model can therefore be defined with a set of ordinary differential equations of form:

$$\frac{dx_i}{dt} = f_i(\mathbf{x}(t){,}\mathbf{u}(t)); 1 \leq i \leq N, \tag{1}$$

where $\mathbf{u}(t)$ is a vector of concentrations of input components and $f_i(\cdot)$ in most cases a nonlinear function. Due to the nonlinearity of ODE models their solutions are usually obtained numerically (e.g. with Euler method). Qualitative properties of the system can also be derived from these models (see Section 4).

In the context of gene regulatory networks three main reaction types are described with the given set of differential equations, namely *transcription*, *translation* and *degradation*. While translation and degradation usually follow ordinary mass-action kinetics, transcription regulation, i.e. activation and repression, is in most cases modelled with the sigmoid shape functions, such as Hill functions [28]. Let's consider that an activator X increases the rate of transcription when it binds to its regulatory region. Promoter activity can be expressed as

$$f_A(X) = \frac{\beta \cdot X^n}{K_d^n + X^n}, \tag{2}$$

where β is maximal promoter activity, K_d dissociation constant and n nonlinearity (Hill) coefficient. Hill coefficient defines the steepness of the transitions and is often interpreted as cooperativity coefficient although it may also arise from other factors [29]. Let's presume that the expression of protein Y is activated by transcription factor X. Simplified model of its dynamics can be presented by

$$\frac{dY}{dt} = \frac{\beta \cdot X^n}{K_d^n + X^n} + \beta_0 - \delta Y, \tag{3}$$

where β_0 is basal transcription rate and δ protein degradation/dilution rate. Complementary regulation type of activation is repression of the transcription, where promoter activity can be expressed as

$$f_R(X) = \frac{\beta}{1 + \left(\frac{X}{K_d}\right)^n}. \tag{4}$$

Even though Hill equations are a relatively good approximation of transcriptional activity, they may reflect inappropriate results in some cases. They imply that the transcription factors are bound to their corresponding binding sites simultaneously when their cooperativity is larger than 1 [30]. If oligomerisation rates are on the same time-scales as other reaction rates, different approaches, such as classical mass-action kinetics, have to be used.

Several other approaches have been derived from ODE models [22]. *Piecewise linear differential equation models* are obtained if sigmoid curves are approximated with step functions and are used in order to simplify the qualitative analysis. ODE models can be extended with the transition to *delayed differential equations* (DDEs) [31] of the form

$$\frac{dx_i}{dt} = f_i(\mathbf{x}(t-\tau),\mathbf{u}(t)); 1 \leq i \leq N, \tag{5}$$

where τ is an N dimensional non-negative vector of discrete time delays. This approach allows us to model slow biochemical reactions, such as gene transcription and translation and protein diffusion, more precisely.

Models based on ODEs are relatively simple and therefore easy to construct when kinetic rates of observed reactions are available. On the other hand the quality and quantity of data needed to derive these rates makes them difficult to apply to poorly characterized or noisy systems. Even when precise data are available, number of parameters that need to be estimated may present major difficulties when inferring larger networks. If inference is successfully performed, high computational costs can present another problem when dealing with such networks. These models are therefore hard to scale up to more complex systems.

2.5. Thermodynamic Modelling

Thermodynamic (TD) or *fractional occupancy* modelling derives from the presumption, that gene expression is proportional to the level of bound activators and inversely proportional to the level of bound repressors [32]. It considers DNA binding and protein interactions in equilibrium condition [33]. The benefit of TD modelling approach is that we can predict occupancies of different binding site types very accurately, e.g. when transcription factors are competing for overlapping binding sites or cooperatively interacting at nearby binding sites. It can also account for very specific nonlinear regulatory responses, such as transcription synergy [34].

Given a set of binding sites, concentrations of transcription factors and their binding affinities, relative probabilities of each binding site configuration can be calculated in the first step. The probability of each configuration can be calculated according to its statistical weight which depends on the number and affinities of occupied binding sites in the configuration and interactions among bound transcription factors. Probabilities of binding sites that lead to transcription activation can be summed in *fractional occupancy*, which may also be expressed as a ratio of weights of binding site occupancies, that lead to transcription activation, to weights of all possible binding site occupancies.

In the second step gene expression level is calculated according to determined fractional occupancies, for which different techniques can be used. For example, calculation can be performed with the product among fractional occupancy of each promoter and its corresponding expression level.

TD models cannot describe dynamical nature of biological systems by themselves [33]. However, it is possible to combine them with the differential equation modelling in order to incorporate changes of gene expression over time. In each iteration fractional occupancies of binding sites and their corresponding expression levels are calculated in dependence of transcription factors concentrations. Concentration values of chemical species are on the other hand derived from the set of ordinary differential equations in each time step.

Let's again consider the same example as described in section 2.4, where protein X activates the transcription. Its fractional occupancy may be expressed with the following equation:

$$f_A = \frac{X^n}{{K_d}^n + X^n}, \tag{6}$$

where K_d describes dissociation constant and n nonlinearity coefficient. Gene expression can be therefore expressed as $f_A \cdot \beta$, where β corresponds to maximal promoter activity. If we combine this model with protein degradation and basal expression of promoter, equation 3 is obtained. If protein X would have a complementary, i.e. repressible role, fractional occupancy would be expressed as:

$$f_R = \frac{1}{1 + \left(\frac{X}{K_d}\right)^n + X^n}. \tag{7}$$

Even though fractional occupancy is able to precisely capture the dynamics when dealing with different types of binding site occupancies, models obtained in these two simple examples obtained with TD modelling are the same as ODE based models.

2.6. Single Molecule Level Models

Biological systems neither posses deterministic or continuous nature which is mostly presumed by ODE and thermodynamic models. *Single molecule level* models on the other hand describe their dynamics on the molecular level. Concentrations of observed species are thus presented with whole numbers. Moreover, reactions which affect their abundances are determined stochastically. When we are dealing with large systems or observing a system over longer periods of time the differences in response of different approaches are negligible [17]. But if the molecular population is small or if the system is sensitive to noise effects, continuous deterministic models may lead us to wrong conclusions [19].

Similar as with deterministic models state of the system can be defined with a vector $\mathbf{x}(t) = [x_1(t), x_2(t),...,x_N(t)]^T$, where $x_i(t)$ is current abundance of chemical species S_i as non-negative whole number. Each of the chemical species from the set $\{S_1,S_2,...,S_N\}$ interacts with others through so called reaction channels $\{R_1,R_2,...,R_M\}$, which are defined by their *state change vectors* and their *propensities*. State change vector defines the species and their quantities that are produced and consumed by each reaction j in the form $\nu_j = (\nu_{1j},\nu_{2j},...,\nu_{Nj})$. Propensities $(a_j(\mathbf{x}(t)))$ can be derived from reaction rates and system volume and are used to calculate the probability for each reaction to occur in a time step dt. Dynamics of the system can be analysed with the establishment of so called *Chemical Master Equation* [19]:

$$\dot{p}(\mathbf{x}(t)) = -p(\mathbf{x}(t)) \sum_{j=1}^{M} a_j(\mathbf{x}(t)) + \sum_{j=1}^{M} p(\mathbf{x}(t) - \nu_j; t) a_j(\mathbf{x}(t) - \nu_j). \tag{8}$$

This equation completely determines the probability of each state, but can be solved analytically only for very small systems. Various numerical approximations are therefore used instead, such as *Stochastic simulation algorithm* (SSA) [35], which can be described as:

1. Initialize the system.
2. Calculate current propensities $a_j(\mathbf{x}(t))$ and their sums $a_0(\mathbf{x}(t))$.
3. Generate random values r_1 and r_2.
4. Determine next time step: $\tau = \frac{1}{a_0(\mathbf{x}(t))} \ln\left(\frac{1}{r_1}\right)$.
5. Determine next reaction R_j to occur according to equation $\sum_{j'=1}^{j-1} a_{j'}(\mathbf{x}(t)) \leq r_2 a_0(\mathbf{x}(t)) < \sum_{j'=1}^{j} a_{j'}(\mathbf{x}(t))$.
6. Calculate state change: $\mathbf{x}(t+\tau) = \mathbf{x}(t) + \nu_j$.
7. Increase time: $t \leftarrow t + \tau$.
8. Return to step 2 unless conditions for stopping the simulation are fulfilled.

Computational time increases drastically with the number of observed chemical species. One of the solutions is to use another approximative approach, namely τ leaping, which does not calculate time steps in each iteration, but uses a fixed value through the whole simulation. In order to obtain valid results, time step must be chosen carefully (no propensity function should change its value significantly within defined time step).

Single molecule level models can be extended with the incorporation of time delays in a similar way as ODE models are extended to DDE models. Here, time delays can be assigned to each product of observed chemical reactions. Dynamics of such models can be analysed with *Delayed SSA*, which can be presented with the following steps [36]:

1. Initialize the system. Clear the queue L, which will contain the chemical species representing queued products and their designated times of appearance.

2. Determine next time step τ and next reaction R_j to occur in the same way as in basic SSA.

3. Calculate state change: Let t be current time and $t_{\min}$ be the lowest value in queue L. If $t + \tau < t_{\min}$, calculate the state change in the same way as in SSA, with an exception of delayed products, which are inserted into L together with their time of appearance in the system. If $t + \tau \geq t_{\min}$ release all the elements of the queue, with their designated times of appearance lower than $t+\tau$ and accordingly update the state vector.

4. Increase time:
$$t \leftarrow \begin{cases} t + \tau, & \text{if } t + \tau < t_{\min} \\ t_{\min}, & \text{otherwise} \end{cases}$$

5. Return to step 2 unless conditions for stopping the simulation are fulfilled.

The effectiveness of described approaches is drastically reduced, when the number of observed chemical reactions and chemical species increases. However, several improvements could be used to decrease the computational complexity of the algorithms based on CME. *Multi Time Scale* modelling approach [37] considers the fact that chemical reactions occur in different time scales. The rates of transcription processes for example are usually significantly higher than the rates of DNA and RNA binding reactions, such as scaffolding, dimerization, linking etc. The common kinetic-propensity approaches presented above may already take into consideration these differences. Nested simulations for each time scale reaction set may on the other hand improve the effectiveness of simulating the state changes [37]. A simple application of this concept can be easily implemented by using SSA specifically for the reactions that occur in time scales of minutes or hours and a nested SSA for the reactions that occur in time scales of seconds.

2.7. Petri Nets

Petri Nets (PNs) are in their most basic form used for modelling and analysis of various concurrent, asynchronous and distributed systems. The mathematical background of PNs enables us to analyse the system we are modelling, while a PN graph gives us its graphical representation. With recent development of different PN extensions they are becoming a powerful tool for describing biological systems. PNs were at first applied to metabolic networks [38], but they are, however, becoming increasingly popular for modelling gene regulatory networks.

PN is a directed-bipartite graph with two different types of vertices: *places* and *transitions*. When modelling biological systems, places correspond to chemical species and transitions to the events occurring in the system, namely chemical reactions that govern dynamics of the system. They are connected by *arcs* (directed edges) which represent how different chemical species interact in the system. At any time, places can hold zero or a positive number of *tokens*. Depending on what reaction we are modelling, these tokens can represent species concentration or simply presence or absence of a certain chemical compound. Distribution and allocation of tokens over places represents current state of the system which is called a *marking* of the PN. Marking of a PN changes when a transition *fires*. Transition can be fired only if all required conditions for that transition are met, e.g. all the chemical species needed for a reaction are present. Let's presume we have two chemical species x_1 and x_2 that can be combined with a chemical reaction t_1 which produces x_3. We can represent the model of this reaction and its different states with a PN as shown in Fig. 4.

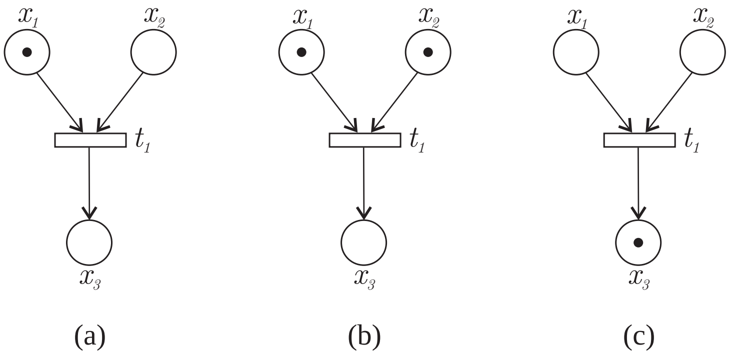

Figure 4. Example of a simple Petri Net of a chemical reaction $x_1 + x_2 \rightarrow x_3$. Fig. (a) presents a scenario where chemical reaction will not happen since one of the reactants is missing (x_2). Fig. (b) shows an enabled transition (chemical reaction can happen) and Fig. (c) the configuration after transition was fired.

We can construct a PN for any reaction or process which is a part of a larger biological system. By combining these basic parts, PNs presenting larger regulatory networks are constructed. Basic PNs support only strictly discrete modelling without the notion of time and are as such used as a framework for many different purely qualitative static modelling approaches [39]. However, with different extensions, PNs can be also used to construct dynamic continuous and discrete [40] models, while considering both deterministic [41] and stochastic [42] representations of the model. Additionally, possibilities to augment continuous deterministic PNs with fuzzy methods are currently being analysed [43]. The proposed solution aims to solve the problem of parameter sloppiness while maintaining a relatively good accuracy of established models at the same time.

Because PNs are a versatile tool for modelling biological systems, the size of the system we can efficiently analyse and model depends on the granularity and approach we aim for. While we can efficiently analyse large static qualitative models, different extensions increase the complexity of PN dynamics and can only be used to construct smaller models.

3. Parameter Estimation Techniques

A common problem of several computational modelling approaches is in their dependence on accurate kinetic data, such as kinetic rate constants or diffusion coefficients. It may be difficult, economically infeasible or even impossible to obtain these parameters experimentally in some cases. *Parameter estimation problem* often represents a serious obstacle when the modelling constraints require a high standard of reliability. Several computational techniques have been developed to overcome this problem. However, no standard has actually been defined, because of intrinsic uncertainty of the underlying biological systems. These methods have to face nonlinear constraints, which are implicit to such systems. The most relevant contributions in the field of parameter estimation have come from control theory of dynamical systems. Control theory is responsible for the development of several optimization and estimation methods used in automatic systems control. Many of these methods, such as *extended Kalman filter*, have already been successfully applied to computational biology [44, 45, 46].

Extended Kalman filtering approach presumes that an arbitrary nonlinear dynamic system may be approximated with a set of state change equations

$$\begin{aligned} \mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \theta) + \mathbf{w}_k, \\ \mathbf{y}_k &= h\left(\mathbf{x}_k\right) + \mathbf{v}_k \end{aligned} \tag{9}$$

where $\mathbf{x}$ is state vector of chemical species, i.e. it defines state of the system, $\mathbf{u}$ vector of system inputs, θ parameter vector which defines kinetic constants, $\mathbf{y}$ output vector of the system, $\mathbf{w}$ and $\mathbf{v}$ Gaussian noise vectors with zero mean and covariance matrices $\mathbf{R}$ and $\mathbf{Q}$ respectively; h is output and f transition function of the system, which completely defines its state change dynamics. Input vector $\mathbf{u}$ contains the parameters that define external influences on the system, such as temperature or pH variations. Output vector $\mathbf{y}$ usually contains experimentally obtained data. Function h describes these data, e.g. it can be interpreted as a response function that approximates the time course of a certain output protein concentrations.

Extended Kalman filter is able to estimate the state vector $\mathbf{x}$ on each discrete time step k, with its estimation vector $\hat{\mathbf{x}}_k$. In order to estimate unknown parameters at the same time state extension has to be performed:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x} \\ \theta \end{bmatrix}. \tag{10}$$

The estimation is obtained in a two stage computation of the predictor-corrector form (see Fig. 5). In first step predicted state vector $\hat{\mathbf{x}}_{k|k-1}$ and covariance matrix $P_{k|k-1}$, which contains predicted variance changes of previously estimated state vector $\hat{\mathbf{x}}_{k-1}$, are evaluated. These predictions are used to construct gain K and to update the state and covariance matrix estimation, $\hat{\mathbf{x}}_k$ and P_k respectively. State estimation $\hat{\mathbf{x}}_k$ is evaluated by adjusting current (predicted) state estimation $\mathbf{x}_{k|k-1}$ with the difference between predicted and estimated output of the system, i.e. $\mathbf{y}_k$ and $h(\hat{\mathbf{x}}_k - 1)$ respectively, amplified or muted by the obtained gain K.

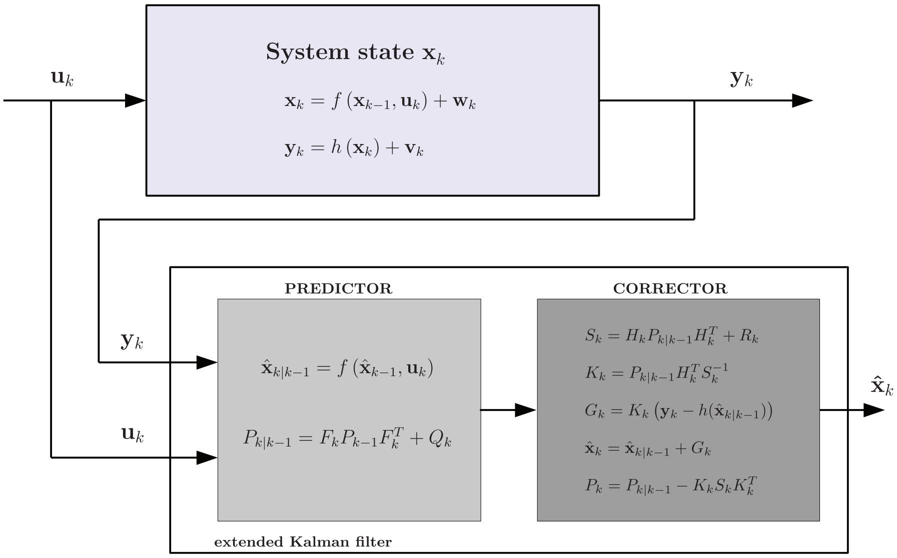
System state $\mathbf{x}_k$
$\mathbf{x}_k = f\left(\mathbf{x}_{k-1}, \mathbf{u}_k\right) + \mathbf{w}_k$
$\mathbf{y}_k = h\left(\mathbf{x}_k\right) + \mathbf{v}_k$
$\mathbf{u}_k$
$\mathbf{y}_k$
PREDICTOR
CORRECTOR
$\mathbf{y}_k$
$\mathbf{u}_k$
$\hat{\mathbf{x}}_{k|k-1} = f\left(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k\right)$
$P_{k|k-1} = F_k P_{k-1} F_k^T + Q_k$
$S_k = H_k P_{k|k-1} H_k^T + R_k$
$K_k = P_{k|k-1} H_k^T S_k^{-1}$
$G_k = K_k\left(\mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1})\right)$
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + G_k$
$P_k = P_{k|k-1} - K_k S_k K_k^T$
$\hat{\mathbf{x}}_k$
extended Kalman filter

Figure 5. A schematic description of extended Kalman filtering, where F_k is Jacobian of function f, evaluated on the previous *a priori* state estimates $\hat{\mathbf{x}}_{k|k-1}$, formally denoted as $J_f^x(\hat{\mathbf{x}}_{k|k-1})$ and similarly, H_k is Jacobian of h, formally denoted as $J_h^x(\hat{\mathbf{x}}_{k|k-1})$. We refer to [47] for a complete derivation of extended Kalman filter equations.

Initial state estimation has to be performed before the filtering. Initial states are usually set to the mean values of all initial concentrations of chemical species $\bar{\mathbf{x}}$. Initial covariance matrix P_0 is on the other hand set to be a diagonal positive definite matrix containing initial mean variance of the state vector $\mathbf{x}_0$ [44]:

$$\hat{\mathbf{x}}_0 = \bar{\mathbf{x}}_0 \tag{11}$$

$$\hat{P}_0 = E\left\{(\mathbf{x} - \bar{\mathbf{x}}_0)(\mathbf{x} - \bar{\mathbf{x}}_0)^T\right\} \tag{12}$$

The initial estimations are very important for the global convergence of the filter. Wrong estimation will cause the errors to be carried on during the entire filtering process, accumulating more and more noisy data. Once the initialization is performed, the evaluations of predictor and corrector equations can be performed in each time step (see Fig. 5). Computational complexity of extended Kalman filtering approach depends on the size of the state vector $\hat{\mathbf{x}}$ and on the filtering time, i.e. number of samples from which estimations are performed.

The main disadvantage of other state of the art parameter estimation methods is in their computational complexity, when applied to models with high numbers of unknown parameters. A model representation of GRN which implements a simple logic gate, such as AND or NOR, may hide several tens of unknown kinetic constants. Extended Kalman filtering approach seems to decrease computational complexity and increase the quality of estimations. Unfortunately, extended Kalman filter may sometimes diverge, which can result in several magnitudes of variations among numerical estimations. However, a validation of estimations based on statistical tests, such as χ^2, can be performed after the filtering [44] in order to confirm the reliability of estimates. Other approaches from control theory have also been successfully applied to parameter estimation problem, e.g. state estimation techniques by *state observers methods* [48, 49] and *particle filtering* [50]. We refer to [44, 8] for a comprehensive review of these approaches.

4. Advanced Model Analysis

Confidence in results obtained with established models can be increased with several validation techniques. Computational models are often based on hypothetical assumptions which are either well known or have to be confirmed. Computational approaches can be used for hypotheses confirmation, e.g. using statistical tests, such as χ^2 [44] or robustness criteria [51]. If confirmation is negative, the identification of erroneous model components and the review of the basic hypothesis of the model itself is vital. In worst case it is necessary to redesign the entire model including the principal hypothesis. If satisfactory accuracy of modelled dynamics is achieved further analytic approaches can be used in order to estimate the *performance*, *robustness* and *stability* of observed biological system.

4.1. Performance Evaluation of Biological Systems

Performance evaluation techniques are used to objectively evaluate the behaviour of biological systems by establishing various *objective functions*. In order to analyse specific performances, both modelling and experimental data can be included in the domain of these functions. A typical example of objective function, which is vastly used in reverse engineering, is *mean square error*:

$$E(Z,X) = \frac{1}{N}\sum_{i=1}^{N}(z_i - x_i)^2, \tag{13}$$

where $\{z_1,z_2,...,z_N\} \in Z$ are samples reflecting the desired dynamics of the system and $\{x_1,x_2,...,x_N\} \in X$ are samples obtained from modelling or experimental results. At glance function $E(Z,X)$ is a simple error measure. However, it can also be used as a naive approach to validate the model accuracy, e.g. by using the experimental results as Z and the modelling results as X in Eq. 13. We can thus quantitatively describe the similarities among the modelling and experimental results. Furthermore, mean square error function can be used to estimate unknown model parameters (see Section 3). The best model response may be obtained by the minimization of the function $E(Z,X)$ regarding the unknown parameters. Unfortunately, computational complexity of this approach makes it applicable only to problems with small numbers of unknown parameters. Improvements can be obtained with the use of special heuristics [8].

Different type of objective functions can be established with several metrics, such as *signal to noise ratio* or quantities which describe the results obtained with robustness, sensitivity or stability analysis (see Sections 4.2 and 4.3). In [13] various metrics were introduced and used to estimate the performance of modelled biological system from the information processing perspective. Characteristics that are similar to the ones used to describe the performance of digital electronic circuits, such as switching times and logical levels, were applied to biological systems. This work was extended in [52], where robustness and logical compatibility among different biological processing structures was also considered in order to automatize the construction of more complex information processing biological systems. While these metrics were only used on selected models of hypothetical biological processing structures, they could also be applied to data gathered from laboratory experiments.

4.2. Robustness and Sensitivity Analysis

Robustness is believed to be the key factor of adaptability in the evolutionary process of biological systems [53]. In cell biology, the robustness is the property of a biological system to remedy a substantial fluctuation in its homeostasis due to a sudden change in the conditions for its stability. Such changes can be provoked by external perturbations on the key parameters of the system. A robust system may respond with a counterbalance effects to these parameter changes, such as bacterial chemotaxis behaviour [28].

Although a general quantitative measure for robustness has still not been established, many efforts have come from various scientific disciplines, especially from the control theory. The use of bifurcation analysis, i.e. the Hopf bifurcation, was studied in [54] for evaluating the robustness of the Laub and Loomis model of cAMP oscillations in *Dictyostelium discoideum* cells. Similarly in [55] the same model was analysed, but with the prevalent use of μ-analysis. An interesting method for robustness analysis using *linear time logic* (LTL) was proposed in [56]. Despite its complexity, this methodology appears to offer a large spectrum of application, especially for synthetic gene networks.

Sensitivity analysis may also represent a metric to quantitatively evaluate the robustness of computational models [57]. This analysis aims to identify the parameters for which small input variations cause substantial variations in model response. Sensitivity analysis approaches can be divided in two categories, i.e. *local sensitivity analysis* and *global sensitivity analysis* approaches. Local sensitivity analysis refers to analytical methods that are capable to evaluate how much the variations in the model outputs can be apportioned to small variations in input parameter values [57]. On the other hand global sensitivity analysis aims to analyse large and even complex perturbations of parameter values with various numerical and statistical methods. State of the art sensitivity analysis approaches can be mainly applied to deterministic models only (for an application to stochastic models see [58]).

Local sensitivity can be estimated by evaluating first-order derivatives of the model output response relatively to input parameters. A quantitative measure can be mathematically represented by sensitivity coefficients of the form [57]:

$$S_i = \frac{\partial y_i}{\partial p} = \lim_{\Delta p \to 0} \frac{y_i\,(p + \Delta p) - y_i(p)}{\Delta p} \qquad (14)$$

Finite difference approximation, direct differential method and adjoint sensitivity analysis [57] evaluate sensitivity in terms of sensitivity coefficients as described in Eq. 14. In metabolic control analysis (MCA) [59] elasticity coefficients were developed for estimating the relationship between model output y_i and specific model parameter p.

Several global sensitivity analysis methods exist (see review in [57]). In order to analyse large parameter perturbations, effective sampling of all possible parameter values becomes crucial. Latin hypercube sampling is usually used in this context rather than Monte Carlo random sampling [57]. Global sensitivity analysis methods can be further divided in two subcategories, i.e. *variance based* and *variance non-based approaches*. Variance based methods, such as *Sobol sensitivity analysis* and *Fourier amplitude sensitivity test* (FAST), aim to estimate the global sensitivity as a relation between statistical variances of model outputs and chosen model parameters. These relations can assume a coefficient-like form such as in Sobol sensitivity analysis [57]:

$$S_{i_1 i_2 \ldots i_s} = \frac{D_{i_1 i_2 \ldots i_s}}{D}, \tag{15}$$

where $S_{i_1 i_2 \ldots i_s}$ are sensitivity coefficient, D is total variance of the system and $D_{i_1 i_2 \ldots i_s}$ are partial variances regarding the chosen parameter x_i. The main disadvantage of variance-based methods is their computational complexity.While the main advantage of non-variance based methods, such as *multi-parametric sensitivity analysis* (MPSA) and *Morris sensitivity analysis*, is their low computational complexity, they imply a certain grade of monotonicity in the model response. An additional disadvantage of the Morris method is in its unreliability when the model response exhibits negative values.

A closely related concepts to sensitivity and robustness are *reliability* and *scalability*. Robustness can be seen as a metric for evaluating reliability of a certain model. Reliable models might be used further for building scalable systems, for which the reliability of each component tends to be crucial for the entire structure. Hence a compatibility among basic modules may be required. Compatibility has already been analysed in [13] in the context of information processing biological structures.

4.3. Stability Analysis

The main goal of stability analysis is to obtain the insights into system's asymptotic behaviour, i.e. its behaviour after a long period of time ($t \to \infty$) without external perturbations, and dependencies of its asymptotic behaviour of given parameter set, also referred to as *bifurcation analysis*. Observed biological systems can in this context exhibit convergence towards *steady states*, which can be analysed with *steady state analysis*, or *stable oscillatory behaviour*, which can be analysed with *limit cycle analysis*. All these methods derive from the theory of *nonlinear dynamical systems* [60], which are usually described with a system of ordinary differential equations. Stability analyses of biological systems are therefore performed on their ODE models (see Section 2.4). We will presume that external inputs are fixed through the course of stability analysis and therefore omit the factor $\mathbf{u}$(t) from the Equation 1.

The main goal of steady state analysis is to investigate the existence and types of steady states the observed biological system may reflect [61]. Existence of a steady state $\mathbf{x}^* = (x_1^*,x_2^*,\ldots,x_n^*)$ can be conditioned with the equation

$$\frac{dx_i}{dt} = f_i(\mathbf{x}^*) = 0; \forall i \in 1,...,N. \tag{16}$$

Steady states can be divided regarding their stability in two types, i.e. *stable* and *unstable*. The neighbourhood will always converge into a stable steady state, but diverge from an unstable steady state. Stability can be determined with the construction of Jacobian matrix of the form

$$J(\mathbf{x}(\mathbf{t})) = \begin{bmatrix} \frac{\partial f_1(\mathbf{x}(\mathbf{t}))}{\partial x_1} & \frac{\partial f_1(\mathbf{x}(\mathbf{t}))}{\partial x_2} & \cdots & \frac{\partial f_1(\mathbf{x}(\mathbf{t}))}{\partial x_n} \\ \frac{\partial f_2(\mathbf{x}(\mathbf{t}))}{\partial x_1} & \frac{\partial f_2(\mathbf{x}(\mathbf{t}))}{\partial x_2} & \cdots & \frac{\partial f_2(\mathbf{x}(\mathbf{t}))}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(\mathbf{x}(\mathbf{t}))}{\partial x_1} & \frac{\partial f_n(\mathbf{x}(\mathbf{t}))}{\partial x_2} & \cdots & \frac{\partial f_n(\mathbf{x}(\mathbf{t}))}{\partial x_n} \end{bmatrix}. \tag{17}$$

Steady state can be referred to as stable, when all real parts of eigenvalues of the Jacobian matrix are negative.

Existence of oscillatory behaviour on the other hand depends on the existence of a *stable limit cycle*. Limit cycle is an isolated simple oriented closed curve trajectory, which does not contain singular points (i.e. steady stable states) [61]. If the system converges to a limit cycle with time, i.e. $t \to \infty$, limit cycle is *stable*. The easiest way to analyse the existence of a limit cycle is with the state space investigation. When dealing with biological systems, this space is strictly limited by minimum and maximum concentrations of observed chemical species and its exhaust investigation does therefore not issue large computational complexities.

The type of behaviour system exhibits is strictly dependant on the parameter values used in its ODE description. Adjusting these values can therefore drastically change system's asymptotic behaviour, e.g. from a stable state convergence to self-sustained oscillations. Transitions among different types of behaviour are called *bifurcations* and the set of parameter values at which the transitions occur *bifurcation points* [62]. Different types of transitions exist, regarding the characteristics of behaviour that arises and characteristics of behaviour that ceases with the transition through the bifurcation point [60], e.g. *saddle-node* bifurcations, *Hopf* bifurcations and *pitchfork* bifurcations.

Stability analysis can not only be used when analysing the asymptotic behaviour of biological systems, but also when evaluating their robustness, e.g. if the distance of parameter values from the bifurcation point is large, the probability that the system will in reality reflect predicted behaviour is much higher and the system is therefore more robust.

5. Computational Design of Biological Systems

The most common approaches to *de novo* engineering of biological systems with desired behaviour are *directed evolution* [63] and *rational design* [64]. While directed evolution is an experimental method that performs artificial evolution on an initial biological system and therefore mimics natural evolution, but in a much shorter time scale, rational design uses engineering approaches to build novel biological systems and is as such a cornerstone of synthetic biology. These approaches combine modularization, rationalization and modelling [64]. Probably most famous results of rational design approach are toggle switch [65] and repressilator [66] circuits. Rational design combined with computer modelling can be also regarded as *computer aided design* of biological systems in which computational models of basic biological components are connected to each other rationally into more complex modules and systems. If analysed behaviour of these networks is appropriate, experimental realization can be conducted. The analyses of their behaviour can be performed with the methods described in Section 4.

The other computational approach in the design of biological systems is based on the unsupervised design with the investigation of search space of all possible solutions and optimization of objective functions which define the correlation among the desired and reflected behaviour of biological system [12]. Until recently this approach was only used for the design of protein and amino acids nucleotide sequences [67]. With the development of characterisation of basic genetic regulatory elements, mutation effects on their genetic functionalities and increased accuracy of their appropriate models, automatic design approaches can also be applied to the regulatory networks [68]. Given an input, which defines the specified behaviour, computational tools, such as AutoBioCAD [68], are able to find the nucleotide sequence and computational model of regulatory network with appropriate dynamics. Initial solution, on which the evolution is performed, is constructed randomly or with rational design from the basic parts characterized within appropriate libraries. In order to achieve the desired behaviour these solution is evolved with the employment of mutation operators, i.e. modifications of its topology (addition, deletion and replacement of basic parts) and kinetic rates (e.g. with the promoter mutations). Search space is therefore comprised of all possible combinations of elementary structures and their mutations that are provided by available libraries [69]. Each intermediate solution is evaluated with the calculation of objective (fitness) function and the best ones are selected for the next iteration of evolution. These mutation and selection operators are applied in accordance with various metaheuristics such as simulated annealing [68] and genetic algorithm [12]. Even though other automatic design approaches have also been reported, they will not be presented here on account of their several limitations in comparison with the described approach. These limitations include either genetic and functional diversity [70, 71] or requirements for predefinition of network topology [72].

Computational design approaches provide us with the results which rely on computational models of elementary structures and their mutual interactions. Experimental realization of these solutions can reflect the behaviour unpredicted within underlying models. On the other hand these solutions can be a basis for further optimization and fine-tuning with various experimental methods such as directed evolution.

6. Conclusion

While the complexity and the size of controllable biological systems rises, computational approaches gain more and more important role in their design and analysis. Here, we reviewed a collection of such approaches, which we find the most important, even though many others exist. Novel techniques are being developed on the account of many limitations of existent ones, such as incompatibility of accuracy of the modelling results with the size and complexity of modelled system. New methodologies are being introduced to the field also from other engineering disciplines, e.g. with the use of Kalman filtering or fuzzy logic methods. In the near future we expect the field of computational biology to evolve further on with the collaboration of new researchers from various scientific disciplines, until the final goal is achieved, i.e. to establish computational methods, that would allow high accuracy in the modelling, analysis and design of both, engineered and natural biological systems of arbitrary size and complexity.

References

[1] A. Kremling and J. Saez-Rodriguez, “Systems biology - an engineering perspective,” *Journal of Biotechnology*, vol. 129, pp. 329–351, 2007.

[2] S. A. Benner and A. M. Sismour, “Synthetic biology,” *Nature Reviews Genetics*, vol. 6, pp. 533–543, 2005.

[3] M. Hecker, S. Lambeck, S. Toepfer, E. van Someren, and R. Guthke, “Gene regulatory network inference: Data integration in dynamic models - a review,” *BioSystems*, vol. 96, pp. 86–103, 2009.

[4] C. Sima, J. Hua, and S. Jung, “Inference of gene regulatory networks using time-series data: A survey,” *Current Genomics*, vol. 10, pp. 416–429, 2009.

[5] P. B. Madhamshettiwar, S. R. Maetschke, M. J. Davis, A. Reverter, and M. A. Ragan, “Gene regulatory network inference: evaluation and application to ovarian cancer allows the prioritization of drug targets,” *Genome Medicine*, vol. 4(41), pp. 1–15, 2012.

[6] C. A. Voigt, “Genetic parts to program bacteria,” *Current Opinion in Biotechnology*, vol. 17(5), pp. 548–557, 2006.

[7] J. Cao and H. Zhao, “Estimating dynamic models for gene regulation networks,” *Bioinformatics*, vol. 24(14), pp. 1619–1624, 2008.

[8] J. Sun, “Parameter estimation using metaheuristics in systems biology: A comprehensive review,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9(1), pp. 185–202, 2012.

[9] R. Hamed, S. Ahson, and R. Parveen, “Designing genetic regulatory networks using fuzzy Petri nets approach,” *International Journal of Automation and Computing*, vol. 7(3), pp. 403–412, 2010.

[10] J. Goutsias, “Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems,” *The Journal of Chemical Physics*, vol. 122, pp. 1–15, 2005.

[11] G. Batt, B. Yordanov, R. Weiss, and C. Belta, “Robustness analysis and tuning of synthetic gene networks,” *Bioinformatics*, vol. 23(18), pp. 2415–2422, 2012.

[12] G. Rodrigo, J. Carrera, T. E. Landrain, and A. Jaramillo, “Perspectives on the automatic design of regulatory systems for synthetic biology,” *FEBS Letters*, vol. 586, pp. 2037–2042, 2012.

[13] M. Moškon and M. Mraz, “Modelling and analysing the information processing capabilities of simple biological systems,” *Mathematical Modelling and Analysis*, vol. 17, pp. 467–484, 2012.

[14] V. Filkov, *Handbook of Computational Molecular Biology*. Chapman & Hall/CRC Press, 2005, ch. Identifying gene regulatory networks from gene expression data, pp. 27.1–27.29.

[15] G. Karlebach and R. Shamir, “Modelling and analysis of gene regulatory networks,” *Nature*, vol. 9, pp. 770–780, 2008.

[16] R. S. Wang, A. Saadatpour, and R. Albert, “Boolean modeling in systems biology: an overview of methodology and applications,” *Physical Biology*, vol. 9(5), pp. 1–14, 2012.

[17] M. Kaern, W. J. Blake, and J. Collins, “The engineering of gene regulatory networks,” *Annual Review of Biomedical Engineering*, vol. 5, pp. 179–206, 2003.

[18] M. Kaern, T. C. Elston, W. J. Blake, and J. J. Collins, “Stochasticity in gene expression: From theories to phenotypes,” *Nature Reviews Genetics*, vol. 6, pp. 451–464, 2005.

[19] H. El Samad, M. Khammash, L. Petzold, and D. Gillespie, “Stochastic modeling of gene regulatory networks,” *International Journal of Robust and Nonlinear Control*, vol. 15, pp. 691–711, 2005.

[20] D. J. Wilkinson, “Stochastic modelling for quantitative description of heterogeneous biological systems,” *Nature Reviews Genetics*, vol. 10(2), pp. 122–133, 2009.

[21] T. Schlitt and A. Brazma, “Current approaches to gene regulatory network modelling,” *BMC Bioinformatics*, vol. 8(6), 2007.

[22] H. de Jong, “Modeling and simulation of genetic regulatory systems: A literature review,” *Journal of Computational Biology*, vol. 9, pp. 67–103, 2002.

[23] G. Pavlopoulos, M. Secrier, C. Moschopoulos, T. Soldatos, S. Kossida, J. Aerts, R. Schneider, and P. Bagos, “Using graph theory to analyze biological networks,” *BioData Mining*, vol. 4(1), 2011.

[24] I. A. Maraziotis, A. Dragomir, and D. Thanos, “Gene regulatory networks modelling using a dynamic evolutionary hybrid,” *BMC Bioinformatics*, vol. 11(140), pp. 1–17, 2010.

[25] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, “Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks,” *Bioinformatics*, vol. 18(2), pp. 261–274, 2002.

[26] H. Lähdesmäki, S. Hautaniemi, I. Shmulevich, and O. Yli-Harja, “Relationships between probabilistic boolean networks and dynamic bayesian networks as models of gene regulatory networks,” *Signal Processing*, vol. 86(4), pp. 814–834, 2006.

[27] T. Tian and K. Burrage, "Stochastic models for regulatory networks of the genetic toggle switch," *Proceedings of the National Academy of Sciences*, vol. 103, pp. 8372–8377, 2006.

[28] U. Alon, *An Introduction to Systems Biology*. Chapman & Hall/CRC, 2007.

[29] M. Andrecut, J. D. Halley, D. A. Winkler, and S. H. S, "A general model for binary cell fate decision gene circuits with degeneracy: Indeterminacy and switch behavior in the absence of cooperativity," *PLoS ONE*, vol. 6(5), 2011.

[30] J. N. Weiss, "The hill equation revisited: uses and misuses," *FASEB Journal*, vol. 11, pp. 835–841, 1997.

[31] M. Barrio, K. Burrage, A. Leier, and T. Tian, "Oscillatory regulation of hes1: Discrete stochastic delay modelling and simulation," *PLoS Computational Biology*, vol. 2(9), 2006.

[32] H. M. Sauro, *Enzyme Kinetics for Systems Biology*. Future Skill Software (Ambrosius Publishing), 2012.

[33] A. Ay and D. N. Arnosti, "Mathematical modeling of gene expression: a guide for the perplexed biologist," *Critical Reviews in Biochemistry and Molecular Biology*, vol. 46(2), pp. 137–151, 2011.

[34] X. He, M. A. H. Samee, C. Blatti, and S. Sinha, "Thermodynamics-based models of transcriptional regulation by enhancers: The roles of synergistic activation, cooperative binding and short-range repression," *PLoS Computational Biology*, vol. 6(9), 2010.

[35] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *Journal of Computational Physics*, vol. 22, pp. 403–434, 1976.

[36] A. S. Ribeiro and J. Lloyd-Price, "SGN Sim, a stochastic genetic networks simulator," *Bioinformatics*, vol. 23, pp. 777–779, 2007.

[37] W. E, D. Liu, and E. Vanden-Eijnden, "Nested stochastic simulation algorithms for chemical kinetic systems with multiple time scales," *Journal of Computational Physics*, vol. 221(1), pp. 158–180, 2007.

[38] R. Hofestdt and S. Thelen, "Quantitative modeling of biochemical networks," *In Silico Biology*, vol. 1, pp. 39–53, 1998.

[39] C. Chaouiya, A. Naldi, E. Remy, and D. Thieffry, "Petri net representation of multi-valued logical regulatory graphs," *Natural Computing*, vol. 10(2), pp. 727–750, 2011.

[40] H. Matsuno, A. Doi, M. Nagasaki, and S. Miyano, "Hybrid Petri net representation of gene regulatory network." *Pacific Symposium On Biocomputing*, pp. 341–352, 2000.

[41] D. Gilbert and M. Heiner, “From Petri nets to differential equations - an integrative approach for biochemical network analysis,” in *Petri Nets and Other Models of Concurrency - ICATPN 2006*, ser. Lecture Notes in Computer Science, S. Donatelli and P. Thiagarajan, Eds. Springer Berlin Heidelberg, 2006, vol. 4024, pp. 181–200.

[42] P. J. E. Goss and J. Peccoud, “Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets,” *Proceedings of the National Academy of Sciences*, vol. 95(12), pp. 6750–6755, 1998.

[43] J. Bordon, M. Moškon, and M. Mraz, “Modeling gene regulatory networks using Petri nets,” in *Proceedings of 22nd International Electrotechnical and Computer Science Conference*, 2012.

[44] G. Lillacci and M. Khammash, “Parameter Estimation and Model Selection in Computational Biology,” *PLoS Computational Biology*, vol. 6(3), 2010.

[45] X. Sun, L. Jin, and M. Xiong, “Extended Kalman Filter for Estimation of Parameters in Nonlinear State-Space Models of Biochemical Networks,” *PLoS ONE*, vol. 3(11), 2008.

[46] X. Liu and M. Niranjan, “State and parameter estimation of the heat shock response system using Kalman and particle filters,” *Bioinformatics*, vol. 28(11), pp. 1501–1507, 2012.

[47] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.

[48] D. Dochain, “State and parameter estimation in chemical and biochemical processes : a tutorial,” *Journal of Process Control*, vol. 13, pp. 801–818, 2003.

[49] M. Quach, N. Brunel, and F. Alche, “Estimating parameters and hidden variables in non-linear state-space models based on ODEs for biological networks inference,” *Bioinformatics*, vol. 23(23), pp. 3209–3216, 2007.

[50] G. Lillacci and M. Khammash, “A distribution-matching method for parameter estimation and model selection in computational biology,” *International Journal of Robust and Nonlinear Control*, 2012.

[51] D. Bates and C. Cosentino, “Validation and invalidation of systems biology models using robustness analysis,” *IET Systems Biology Biology*, vol. 5, no. 4, pp. 229–244, 2011.

[52] M. Moškon, “Computer structures perspective on switching dynamics of simple biological systems,” Ph.D. dissertation, University of Ljubljana, Faculty of Computer and Information Science, 2012.

[53] H. Kitano, “Biological robustness,” *Nature Reviews Genetics*, vol. 5(11), pp. 826–837, 2004.

[54] R. Ghaemi and D. D. Vecchio, "Evaluating the robustness of a biochemical network model," in *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, pp. 615–620.

[55] J. Kim, D. G. Bates, I. Postlethwaite, L. Ma, and P. A. Iglesias, "Robustness analysis of biochemical network models," *IEE Proceedings Systems Biology*, vol. 153(3), pp. 96–104, 2006.

[56] A. Rizk, G. Batt, F. Fages, and S. Soliman, "A general computational method for robustness analysis with applications to synthetic gene networks," *Bioinformatics*, vol. 25, no. ISMB 2009, pp. i169–i178, 2009.

[57] Z. Zi, "Sensitivity analysis approaches applied to systems biology models," *IET Systems Biology*, vol. 5(6), pp. 336–346, 2011.

[58] A. Degasperi and S. Gilmore, "Sensitivity Analysis of Stochastic Models of Bistable Biochemical Reactions," *Lecture Notes in Computer Science*, no. 5016, pp. 1–20, 2008.

[59] C. Reder, "Metabolic control theory: A structural approach," *Journal of Theoretical Biology*, vol. 135, pp. 175–201, 1988.

[60] S. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, ser. Studies in nonlinearity. Perseus Books Group, 2008.

[61] L. Edelstein-Keshet, *Mathematical Models in Biology*. Siam, Philadelphia, 2005.

[62] P. Fall, E. S. Marland, J. M. Wagner, and J. J. Tyson, *Computational Cell Biology (Interdisciplinary Applied Mathematics)*. Springer, 2002.

[63] F. H. Arnold, "Design by directed evolution," *Accounts of Chemical Research*, vol. 31, pp. 125–131, 1998.

[64] A. S. Khalil and J. J. Collins, "Synthetic biology: Applications come of age," *Nature Reviews Genetics*, vol. 11(5), pp. 367–379, 2010.

[65] T. S. Gardner, C. R. Cantor, and J. J. Collins, "Construction of a genetic toggle switch in Escherichia coli," *Nature*, vol. 403, pp. 339–342, 2000.

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

[67] G. Rodrigo, T. E. Landrain, and A. Jaramillo, "De novo automated deisgn of small RNA circuits for engineering synthetic riboregulation in living cells," *Proceedings of the National Academy of Sciences*, 2012.

[68] G. Rodrigo and A. Jaramillo, "AutoBioCAD: Full biodesign automation of genetic circuits," *ACS Synthetic Biology*, vol. 0, no. 0, p. null, 0.

[69] G. Rodrigo, J. Carrera, and A. Jaramillo, “Computational design of synthetic regulatory networks from a genetic library to characterize the designability of dynamical behaviors,” *Nucleic Acids Research*, vol. 39, 2011.

[70] M. A. Marchisio and J. Stelling, “Automatic design of digital synthetic gene circuits,” *PLoS Computational Biology*, vol. 7(2), 2011.

[71] J. Beal, R. Weiss, D. Densmore, A. Adler, E. Appleton, J. Babb, S. Bhatia, N. Davidsohn, T. Haddock, J. Loyall, R. Schantz, V. Vasilev, and F. Yaman, “An end-to-end workflow for engineering of biological networks from high-level specifications,” *ACS Synthetic Biology*, vol. 1, no. 8, pp. 317–331, 2012.

[72] L. Huynh, J. Kececioglu, M. Koppe, and I. Tagkopoulos, “Automatic design of synthetic gene circuits through mixed integer non-linear programming,” *PLoS ONE*, vol. 7(4), 2012.

MA