# GeNGe: Systematic Generation of Gene Regulatory Networks

## Supplementary Material

Hendrik Hache*, Christoph Wierling, Hans Lehrach, and Ralf Herwig
Max Planck Institute for Molecular Genetics, Ihnestr. 63-73, 14194 Berlin, Germany
*hache@molgen.mpg.de

## Contents

| Section | Title | Page |
|---------|-------|------|
| 1 | Reverse Engineering Methods | 2 |
| 2 | Tools for Forward Modeling of Gene Regulatory Networks | 2 |
| 3 | GEne Network GEnerator |
| 3.1 | Network Level | 4 |
| 3.1.1 | Motif Networks | 4 |
| 3.1.2 | Scale-free Networks | 5 |
| 3.1.3 | Random Networks | 5 |
| 3.2 | Kinetic Level | 5 |
| 3.3 | Simulation Level | 7 |
| 4 | Transcription Function | 8 |
| 4.1 | Introduction | 8 |
| 4.2 | Used Kinetics |
| 4.2.1 | Kinetic 1 | 12 |
| 4.2.2 | Kinetic 2 | 12 |
| 4.2.3 | Kinetic 3 | 12 |
| 4.2.4 | Kinetic 4 | 12 |
| 4.2.5 | Kinetic 5 | 13 |
| 5 | Examples | 13 |
| 5.1 | Saturation Effect and Transcription Activation Threshold | 13 |
| 5.2 | Oscillation | 14 |
| 5.3 | Bistability | 16 |
| 6 | Time Scaling | 18 |
1 Reverse Engineering Methods

In recent years, several reverse engineering methods were proposed which are based on different mathematical models, such as Boolean networks (Liang et al., 1998), linear models (D’haeseleer et al., 1999), differential equations (Chen et al., 1999), association networks (Basso et al., 2005; Schäfer and Strimmer, 2005), static Bayesian networks (Friedman et al., 2000), neural networks (Hache et al., 2007), state space models (Rangel et al., 2004; Beal et al., 2005), and dynamic Bayesian networks (Friedman et al., 1998; Yu et al., 2004; Werhli et al., 2006). A historical perspective of different methods applied until 2002 is given by van Someren et al. (2002). de Jong (2002) and more recently Gardner and Faith (2005) discuss further details and mathematical aspects. The validation of the performances of these methods is a key aspect for showing the benefit of such algorithms. This must be done under a controlled framework starting with artificial data. For that purpose, the web-application GEne Network GEnerator (GeNGe, http://genge.molgen.mpg.de) can be used for the automatic generation of many different expression profiles based on gene regulatory networks by performing various in silico experiments.

2 Tools for Forward Modeling of Gene Regulatory Networks

There are some tools previously published that provide also forward gene regulatory network modeling approaches, such as SynTReN (den Bulcke et al., 2006), RMBNToolbox (Aho et al., 2007), RENCO (Roy et al., 2008), and SynBioSS (Hill et al., 2008). A general focus among others are quite similar, namely the generation of synthetic datasets, which can be, e.g., used for a systematic validation of network inference methods. Nevertheless, the functionalities can vary across the tools substantially, mainly regards to analysis features and the possibility of performing various in silico experiments. In the following we introduce them shortly.

SynTReN (Synthetic Transcriptional Regulatory Networks) was developed by den Bulcke et al. (2006). It is implemented in Java and is running on a local machine. In a single window the parameters for the network and data generation can be specified. SynTReN generates a topology by selecting subnetworks from known biological networks (Escherichia coli or Saccharomyces cerevisiae). These networks show similar topological properties compared to the source networks, such as average indegree and average path length. The gene regulation is modeled by Michaelis-Menten and Hill kinetics with automatically chosen parameters. Other kinetics cannot be selected and the simulation parameters cannot be adjusted. The model system is solved to calculate only normalized steady-state levels. Therefore, simulations of large systems are possible.
RMBNToolbox (Random Models for Biochemical Networks) was developed by Aho et al. (2007). The focus of this Matlab package is the (random) generation and simulation of biochemical systems. The toolbox offers a possibility to check the generated model on consistency and stability. Kinetics are randomly chosen from a kinetic law library. Nevertheless, the toolbox can also be used for modeling gene regulation on a simplified level. Transcription factor concentration is assumed to be equal to the concentration of the produced transcript and the transcription kinetics are based on the rate laws suggested by Mendes et al. (2003). Time course data are then calculated within Matlab.

RENCO (REgulatory Network generator with COmbinatorial control) was developed by Roy et al. (2008). It is a command line tool for the generation of networks (scale-free, exponential degree, or user-defined) and the setup of gene regulatory models, including an mRNA and protein layer, based on kinetics for the combinatorial control of transcription (Schilstra and Bolouri, 2002). The tool lacks an own visualization interface, but the model can be exported in the SBML format and imported to other tools, where the simulation and visualization can be performed.

SynBioSS (Synthetic Biology Software Suite) developed by Hill et al. (2008) is a recently proposed tool for detailed modeling and stochastic simulation of gene regulatory models. The software suite, which has to be installed on a local computer, consists of a desktop simulator user interface. A list of reactions can be uploaded or can be input by the user successively. The tool provides a large repository of reaction kinetics and parameters, stored in a Wiki database. Stochastic simulations are performed locally. Furthermore, it supports parallel execution of multiple trajectories, which are required for stochastic simulation to obtain a population distribution.

Another application is PyBioS, which is developed by Wierling et al. (2007). The focus of this web tool is the highly detailed description and simulation of small and large biochemical reaction systems. It is not restricted to gene regulation. A model can be constructed from various databases, such as KEGG (Kanehisa et al., 2008), Reactome (Matthews et al., 2009), or ConsensusPathDB (Kamburov et al., 2009), or by hand. Kinetics are assigned to each reaction automatically or has to be chosen from a kinetic repository. Furthermore, the kinetic parameters can be changed individually. An ordinary differential equation (ODE) system is generated automatically and solved numerically. Simulation results are visualized either as time course plots or within the network graph, which facilitates the interpretation of the results.

We developed GeNGe to combine and extent features of such tools described above in a single framework. A goal in the development of GeNGe is that with a minimum of user-activity models of gene regulatory networks are generated and time course data under different perturbation conditions are calculated. The tool does not need to be installed locally since it is available via the web. Networks can be easily generated and models are set up with a specified kinetic schema, selected form a set of supported kinetic laws. Various in silico experiments can be performed, e.g., local perturbations, i.e. knockdowns of a certain degree, of
a single gene or multiple genes. This can be done for an arbitrary number of networks in parallel to generate as many datasets as needed. Nevertheless, it is possible to create a network by successively adding nodes and edges by hand, selecting a kinetic schema, and analyse subsequently the systems behavior under different conditions.

3 GEne Network GEnerator

The workflow of GEne Network GEnerator (GeNGe) is divided into three levels, the network level, the kinetic level, and the simulation level. In the first level, networks are added to a network repository. In the second level, the kinetic laws for the description of transcription, translation, and degradation are set. In the third level, the parameters and initial values are specified and the simulations are performed. In the following, each level is described in more details.

3.1 Network Level

Within the network level networks are generated that built the basis for modeling of gene regulatory networks (GRNs). Various network types are supported by GeNGe. An upload of user-defined networks given as adjacency matrices or tables of regulatory interactions is possible. Furthermore, a set of networks are pre-defined, e.g., artificial networks showing different dynamic characteristics, network motifs, or biological GRNs. Networks can be constructed or manipulated by adding, deleting, or changing edges in the network. In addition, different artificial types of networks, such as motif networks, scale-free networks, and random networks can be generated automatically in GeNGe.

3.1.1 Motif Networks

Biological networks show different properties compared to random networks, such as in- and out-degree distributions. Furthermore, several small network motifs, i.e. a subnetwork showing certain characteristics, occur more often in GRNs than expected by chance. Different types of such motifs have been experimentally identified by Lee et al. (2002): single-input motif (SIM), feed-forward loop (FFL), multi-component loop (MCL), multi-input motif (MIM), regular chain (RC), and auto-regulation (AR). These motifs can be used in GeNGe as basic building blocks for the construction of GRNs. During network generation motifs are assembled in the current network according to an adjustable motif distribution.

The network generation in GeNGe is an iterative process starting with an empty network. A motif is randomly selected according to the motif distribution. The motif-specific properties, such as the number of regulators and targets, are as selected randomly as well. Let N be the number of nodes in the selected motif. This
motif is assembled into the current network by the following procedure. First, N nodes are chosen, either from the nodes in the current network (K nodes) or nodes which are not already in the network (N-K nodes). Afterwards, each of the motif nodes are assigned to one of these N nodes. A larger K, i.e., a larger number of nodes which are already in the network, results finally in a more connected and wired network. By changing the parameter ‘Connectivity between motifs’ it is possible to influence the ratio K/(N-K) and have a bias to a more connected graph or to more separated motifs.

The process of assembling network motifs is repeated until all nodes are included in the network.

3.1.2 Scale-free Networks

It is assumed that many real world networks are scale-free networks, i.e. they have power law degree distributions in in- and out-degree. Bollobás et al. (2003) introduced a procedure to generate directed scale-free graphs. Similar to undirected scale-free graphs (see e.g. Albert and Barabási, 2002) a network grows under the assumption of preferential attachment depending on in- and out-degree. The resulted network shows a power-law distribution of in- and out-degree.

3.1.3 Random Networks

A random network growth starts with an unconnected graph of n nodes and adds edges between them at random. Each possible edge has the same probability and is independent of the others. This probability can be chosen arbitrarily in the interval [0,1]. Such a random network is also called Erdős-Rényi network (Erdős and Rényi, 1959). See also Barabási and Oltvai (2004) for more information.

3.2 Kinetic Level

Given a specific network structure, GeNGe provides a mathematical model for the regulatory interactions that covers the transcriptional as well as the translational layer. These layers include instances for mRNAs and proteins as well as for polymerase and ribosome. Furthermore, an RNase and a proteasome instance catalyze the mRNA and protein degradation, respectively. The kinetics of the concentration of a mRNA and protein, associated to a certain gene, are described by

\[ \frac{d[\text{mRNA}]}{dt} = k_1[\text{Polymerase}]\phi_n(c_1, \ldots, c_n) - k_2[\text{RNase}]\delta([\text{mRNA}]), \quad (1) \]

\[ \frac{d[\text{Protein}]}{dt} = k_3[\text{Ribosome}][\text{mRNA}] - k_4[\text{Proteasome}]\delta([\text{Protein}]). \quad (2) \]

Here, \( k_1 \) and \( k_3 \) are the maximal transcription rate of the mRNA and the maximal translation rate of the corresponding protein, respectively. \( k_2 \) and \( k_4 \) are
the degradation rates. Translation is described with a linear kinetic. Usually, degradation is modeled with a linear kinetic. However, it is observed that protein degradation can also be also described by a Michaelis-Menten kinetic (Grilly et al., 2007). Therefore, GeNGe supports a linear ($\delta(x) = x$) and a Michaelis-Menten like ($\delta(x) = x/(K_M + x)$) mRNA and protein degradation, where $x$ is the mRNA or protein concentration. An additional parameter $K_M$, specific for each mRNA and protein is introduced for the Michaelis-Menten kinetic, representing the substrate concentration, i.e., mRNA or protein concentration, at which the reaction rate reaches half of its maximum value. The degradation kinetic schema for mRNAs and proteins can be chosen independently.

We partially nondimensionalized the Eq. (1) and (2) by rescaling the concentrations

$$[\text{mRNA}] = M \cdot [\text{mRNA}'], \quad [\text{Protein}] = P \cdot [\text{Protein}]',$$  

where $M$ and $P$ are assumed to be normalization concentrations. The concentrations $[\cdot]'$ are now dimensionless. Following the approach by Elowitz and Leibler (2000), the normalization factor $M$ of the mRNAs can be interpreted as the translation efficiency that is the average concentration of proteins produced per mRNA molecule. The protein concentration normalization factor, $P$, can be considered as the concentration to half-maximally activate or repress the gene transcription, while other transcription factor concentrations are considered as constant. If the protein does not act as a transcription factor, the normalization is an artificial scaling factor.

The kinetic reads with the scaling factors

$$\frac{d[\text{mRNA}']}{dt} = k'_1[\text{Polymerase}]\phi_n(P \cdot c'_1, \ldots, P \cdot c'_n)$$
$$- k'_2[\text{RNase}]\delta(M \cdot [\text{mRNA}']),$$

$$\frac{d[\text{Protein}']}{dt} = k'_3[\text{Ribosome}][\text{mRNA}]'$$
$$- k'_4[\text{Proteasome}]\delta(P \cdot [\text{Protein}']),$$

with the rescaled kinetic constants

$$k'_1 = \frac{k_1}{M}, \quad k'_2 = \frac{k_2}{M},$$

$$k'_3 = \frac{M}{P} k_3, \quad k'_4 = \frac{k_4}{P}.$$  

Each normalization factor of the mRNAs and proteins can be specified within GeNGe independently. Eq. (4) and (5) are numerically solved by GeNGe. The time series output are rescaled to the input dimension.

Various non-linear transcription functions $\phi_n(c_1, \ldots, c_n)$ can be chosen for transcriptional regulation, which are dependent of $n$ concentrations $\{c_i\}$ associated
3.3 Simulation Level

with the respective transcription factors. See section 4 for more details and the provided transcription kinetics based on the bio-logic by Schilstra and Nehaniv (2008). It is assumed that the transcription factors bind independently and have a specific regulation strength on the transcription of the target gene. For more complex dynamics, GeNGe supports joint bindings of sets of transcription factors to the DNA. Each set acts independently of the other sets. Further, it has an individual regulation strength. In such a single set, a transcription factor can occur more than once to model Hill-like kinetics.

The kinetic functions are set for all genes by choosing a certain kinetic schema. Nevertheless, all kinetic parameters are specific for each gene, hence, they are independent across genes. An additional index for the parameters indicating the gene are omitted here due to simplicity.

In the current version of GeNGe units of the variables and constants are set as follows

- the functions $\phi_n$ and $\delta$ are dimensionless
- unnormalized concentrations of mRNAs and proteins in [nM],
- the normalization constants $M$ and $P$ in [nM],
- the kinetic constants $k_1, k_2, k_4$ in [min$^{-1}$],
- the kinetic constants $k_3$ in [nM$^{-1}$min$^{-1}$],
- the Michaelis constants $K_M$ in [nM].

3.3 Simulation Level

All kinetic parameters and initial values can be specified in the last level before the simulations will be performed. The unnormalized initial values for each component in the model can be chosen arbitrarily. To simulate global perturbations (for example network noise) it is also possible to choose the initial values randomly from a Gaussian distribution centered at the corresponding steady-state concentrations. Furthermore, each kinetic parameter can be changed individually. Specific perturbations of a gene or of a set of genes, e.g., knock-downs, are realized by reducing the corresponding transcriptional rate $k_1$ according to the specified perturbation level (partial knock-downs). All other parameters and initial values for control and knock-down run are the same. For a 100% knock-down the transcriptional rate factor is set zero.

Based on the network topology, the kinetics, and the parameters an ordinary differential equation (ODE) system of the network is set-up and exported to an ODE solver by web services API provided by PyBioS (Wierling et al., 2007).

The procedures can be repeated with different settings and used in an iterative way. All simulation results, including the network structure as a matrix, the mathematical model as SBML and time series of each in silico experiment can be downloaded.
4 Transcription Function

4.1 Introduction

We used the bio-logic described by Schilstra and Nehaniv (2008) for transcription kinetics of GRNs. Starting with an example, the transcription regulation function \( \phi_n \) for one gene with three transcription factors (TF) with the concentrations \( c_1, c_2, \) and \( c_3 \) is given by

\[
\phi_3(c_1, c_2, c_3) = \frac{D}{N} \quad \text{with}
\]

\[
D = w_{(0,0,0)} + w_{(1,0,0)}x_1 + w_{(0,1,0)}x_2 + w_{(0,0,1)}x_3 + \\
+ w_{(1,1,0)}r_{(1,1,0)}x_1x_2 + w_{(1,0,1)}r_{(1,0,1)}x_1x_3 + \\
+ w_{(0,1,1)}r_{(0,1,1)}x_2x_3 + w_{(1,1,1)}r_{(1,1,1)}x_1x_2x_3
\]

\[
N = w_{(0,0,0)} + w_{(1,0,0)}x_1 + w_{(0,1,0)}x_2 + w_{(0,0,1)}x_3 + \\
+ w_{(1,1,0)}x_1x_2 + w_{(1,0,1)}x_1x_3 + w_{(0,1,1)}x_2x_3 + \\
+ w_{(1,1,1)}x_1x_2x_3.
\]

The definitions of the parameters is given below. The general definition of the regulation function dependent of an arbitrary number of TFs is

\[
\phi_n(c_1, \ldots, c_n) = \frac{\sum v w_v \prod_{i=1}^n x_i^{v_i}}{\sum w_v \prod_{i=1}^n x_i^{v_i}} \quad \text{with} \quad x_i = \frac{c_i}{K_i}. \quad (9)
\]

The transcription is stimulated by certain combinatorial effects of TFs with different strengths. Every configuration of \( n \) TFs is assigned a binary vector \( v = (v_1, \ldots, v_n) \), where \( v_i \in \{0, 1\} \) denotes occurrence of the \( i \)th TF in this configuration. There are \( 2^n \) possible configurations of TFs (including the one with no TFs, \( v = (0, \ldots, 0) \), i.e., unbounded DNA). \( \sum_v \) is the sum over all such configurations. \( \{w_v\} \) are modulation coefficients which describe the contribution of a TF configuration \( v \) on the transcriptional rate, with \( w_v \geq 0 \). \( w_v = 0 \) means that the configuration \( v \) has no influence on the transcription.

A measure of cooperativity in the binding of a certain configuration \( v \) of regulators is given by the parameter \( r_v \). For the unbounded DNA (\( v = (0, \ldots, 0) \) and single TFs the cooperative factors are \( r_v = 1 \). For non-cooperative transcription factor binding we get \( r_v = 1 \) which we assume in the models. The \( x_i \) is the normalized TF concentrations \( c_i \) by a normalization factor \( K_i \), which is the equilibrium dissociation constant of the complex \([\text{DNA} \cdot \text{TF}_i]\). Every gene can have different numbers of transcription factors and modulation coefficients.

To describe joint bindings of TFs, i.e., binding of multiple TFs jointly to the promoter of a gene, the transcription function Eq. (9) has to be modified. The TFs in such a set do not effect individually the transcription rate of the target
4.1 Introduction

Figure 1: Illustration of joint interaction sets. (a) Graph of a four node example. The nodes represent the gene and the corresponding protein, which can act as a transcription factor. A green edge indicates a positive regulatory effect of the transcription factor on the transcription of the connected gene (activation) and a red edge with a bar at the end indicates a negative effect (inhibition). Some transcription factors (Gene2 and Gene3) have only a regulatory effect, if they bind jointly. These are combined to interactions sets, indicated in the graph by black edges to a joint node. The kind of regulatory effect of an interaction set, i.e., activation or inhibition, is represented by a green or red edge, respectively, from the joint node to the target gene. Single transcription factors (Gene1) are considered as interaction sets with one element. (b) Interaction sets bind to the promoter of the target gene and, thus, can regulate the transcription of this gene. It is assumed, that each interaction set binds independently from other interaction sets.

gene. However, if they bind together they contribute to the transcription rate as an inducer or repressor with a certain regulation strength.

Instead of considering single TFs we introduce \( \nu \) sets of joint interactions \( I_1, \ldots, I_\nu \) with \( \nu \leq n \) (see Fig. 1). Each set can contain one or more TFs and each TF is in exactly one set. Therefore, all sets are disjoint and the conjunction of all sets is equal to the set of all TFs. In the modified transcription function \( \phi' \) all configurations of interaction sets are considered instead of configurations of single TFs. Omitting the prim for indication of modified definitions (\( \phi' \rightarrow \phi \)), the transcription function reads

\[
\phi_n(c_1, \ldots, c_n) = \frac{\sum_{\nu} w_{\nu} r_{\nu} \prod_{i=1}^{\nu} \prod_{x \in I_i} x^{v_{ij}}}{\sum_{\nu} r_{\nu} \prod_{i=1}^{\nu} \prod_{x \in I_i} x^{v_{ii}}}. \tag{10}
\]

Here, \( \nu \) is a redefined binary vector of length \( \nu \) assigned to a configuration, where \( v_i \in \{0, 1\} \) denotes occurrence of the \( i \)th interaction set in this configuration. \( w_{\nu} \) is the modulation coefficient and \( r_{\nu} \) is the cooperativity factor of configuration \( \nu \). The normalized concentrations were redefined. The normalization constant of a certain TF is dependent of the interaction set \( I_k \) to which the TF belongs to and is equal to the dissociation constant of the TFs-DNA complex \([\text{DNA } I_k]\), i.e., binding of all TFs in this particular interaction set to the DNA.
With the substitution
\[ \bar{x}_i = \bar{x}(I_i) = \prod_{x \in I_i} x \]  
(11)
one obtains for Eq. (10)
\[ \phi_n(c_1, \ldots, c_n) = \frac{\sum \nu \prod_{i=1}^{\nu} \bar{x}_i^{v_i}}{\sum \nu \prod_{i=1}^{\nu} \bar{x}_i^{v_i}}. \]  
(12)

Note that the function is still implicitly dependent of all TF concentrations \( \{c_i\} \).

As an example (see Fig. 1, the transcription function of a gene with three transcription factors TF1, TF2, TF3, where TF1 is an inhibitor and TF2 and TF3 are in one interaction set, is given by
\[ \phi_3(c_1, c_2, c_3) = \frac{D}{N} \]  
with
\[ D = w_{(0,0,0)} + w_{(1,0,0)}\bar{x}_1 + w_{(0,1,1)}r_{(0,1,1)}\bar{x}_2\bar{x}_3 + w_{(1,1,1)}r_{(1,1,1)}\bar{x}_1\bar{x}_2\bar{x}_3 \]
\[ N = w_{(0,0,0)} + w_{(1,0,0)}\bar{x}_1 + w_{(0,1,1)}\bar{x}_2\bar{x}_3 + w_{(1,1,1)}\bar{x}_1\bar{x}_2\bar{x}_3. \]

We simplified the transcription function Eq. (12) by considering interaction set specific regulation strengths \( \{a_i\} \) independent of all other interaction sets, but still specific for the target gene. \( a_i > 0 \) means activation of the target gene by a factor of \( 2^{a_i} \) and \( a_i < 0 \) means inhibition by a factor \( 2^{-a_i} \). A modulation coefficient of a configuration is composed by a product of each interaction set specific regulation strength
\[ w_\nu = \prod_{i=1}^{\nu} (2^{a_i})^{v_i}. \]  
(14)

Further, we assume that the interaction sets bind independently from each other, i.e., \( r_\nu = 1 \ \forall \nu \). With this assumption, Eq. (14), and \( w_{\nu=(0,\ldots,0)} = 1 \) we obtain for Eq. (12)
\[ \phi_n(c_1, \ldots, c_n) = \frac{\sum \nu \prod_{i=1}^{\nu} [2^{a_i}\bar{x}_1]^{v_i}}{\sum \nu \prod_{i=1}^{\nu} \bar{x}_i^{v_i}} = \frac{\prod_{i=1}^{\nu} [1 + 2^{a_i}\bar{x}_1]}{\prod_{i=1}^{\nu} [1 + \bar{x}_1]}. \]  
(16)
4.2 Used Kinetics

The step from Eq. (15) to Eq. (16) can be proved by mathematical induction for every natural number \( \nu \), considering that \( \sum \nu \) is the sum over all binary vectors of length \( \nu \). The last equation can also be written as

\[
\phi_n(c_1, \ldots, c_n) = \prod_{i=1}^{\nu} \left[ 1 + \left( 2^{a_i} - 1 \right) \frac{\bar{x}_i}{1 + \bar{x}_i} \right]. \tag{17}
\]

With \( w_{\nu=(0, \ldots, 0)} = 0 \), i.e., no basal expression of regulated genes, one obtains

\[
\phi_n(c_1, \ldots, c_n) = \frac{\sum \prod_{i=1}^{\nu} \left[ 2^{a_i} \bar{x}_i \right]^{\nu_i} - 1}{\sum \prod_{i=1}^{\nu} \bar{x}_i^{\nu_i}} \tag{18}
\]

\[
= \prod_{i=1}^{\nu} \left[ 1 + 2^{a_i} \bar{x}_i \right] - 1 \frac{\prod_{i=1}^{\nu} \bar{x}_i}{\prod_{i=1}^{\nu} \left[ 1 + \bar{x}_i \right]} . \tag{19}
\]

As stated above, the step from Eq. (18) to Eq. (19) can be proved by mathematical induction for every natural number \( \nu \). One obtains

\[
\phi_n(c_1, \ldots, c_n) = \prod_{i=1}^{\nu} \left[ 1 + (2^{a_i} - 1) \frac{\bar{x}_i}{1 + \bar{x}_i} \right] - \prod_{i=1}^{\nu} \frac{1}{1 + \bar{x}_i} \tag{20}
\]

In comparison to Eq. (9) the simplified Eq. (17) and (20) are much more convenient for modeling purposes. Summarizing, the simplifications include the assumptions of interaction sets (Eq. (11)), independent binding of such interaction sets \( (r_{\nu} = 1 \ \forall \nu) \), and specific regulation strengths of interaction sets. Further, the regulation strength of multiple interaction sets is the product of individual strengths (Eq. (14)).

As an example (see Fig. 1), the simplified transcription function of a gene with three transcription factors TF\(_1\), TF\(_2\), TF\(_3\), where TF\(_1\) is an inhibitor and TF\(_2\) and TF\(_3\) are in one interaction set is with Eq. (17)

\[
\phi_3(c_1, c_2, c_3) = \left[ 1 + (2^{a_1} - 1) \frac{\bar{x}_1}{1 + \bar{x}_1} \right] \left[ 1 + (2^{a_2} - 1) \frac{\bar{x}_2 \bar{x}_3}{1 + \bar{x}_2 \bar{x}_3} \right] . \tag{21}
\]

The regulation strengths are \( a_1 < 0 \) (Gene1) and \( a_2 > 0 \) (interaction set). Note that the normalized TF concentration \{\( \bar{x}_i \)\} are used.

4.2 Used Kinetics

In the following, we introduce the transcription kinetics which are supported by GeNGe. The kinetics are derived from Eq. (17) and (20).
4.2 Used Kinetics

4.2.1 Kinetic 1

Under the assumption that input genes, i.e., genes with no regulator, have a basal expression and regulated genes have no such basal expression, except genes with only inhibitors, the regulation factor writes

\[ \phi_{v}^{(1)}(c_1, \ldots, c_n) = \begin{cases} 
\varphi_1 - \varphi_2 & \text{for } v \neq 0 \text{ and } \arg\max_i (a_i) > 0, \\
\varphi_1 & \text{for } v \neq 0 \text{ and } \arg\max_i (a_i) < 0, \\
1 & \text{for } v = 0,
\end{cases} \]  
(22)

with the terms

\[ \varphi_1 = \prod_{i=1}^{v} \left[ 1 + (2^{a_i} - 1) \frac{\bar{x}_i}{1 + \bar{x}_i} \right], \]  
(23)

\[ \varphi_2 = \prod_{i=1}^{v} \frac{1}{1 + \bar{x}_i}. \]  
(24)

The term \( \varphi_2 \) in equation (22) implements the assumption, that regulated genes do not have an constant production rate.

4.2.2 Kinetic 2

Under the assumption that all genes, except genes with only activators as regulators, have a basal expression, the regulation factor writes

\[ \phi_{v}^{(2)}(c_1, \ldots, c_n) = \begin{cases} 
\varphi_1 - \varphi_2 & \text{for } v \neq 0 \text{ and } \arg\min_i (a_i) > 0, \\
\varphi_1 & \text{for } v \neq 0 \text{ and } \arg\min_i (a_i) < 0, \\
1 & \text{for } v = 0.
\end{cases} \]  
(25)

4.2.3 Kinetic 3

Under a slightly different assumption that input genes have a basal expression and all others genes do not have such a constant expression, the regulation factor writes

\[ \phi_{v}^{(3)}(c_1, \ldots, c_n) = \begin{cases} 
\varphi_1 - \varphi_2 & \text{for } v \neq 0, \\
1 & \text{for } v = 0.
\end{cases} \]  
(26)

4.2.4 Kinetic 4

A basal expression for all genes results in the regulation factor

\[ \phi_{v}^{(4)}(c_1, \ldots, c_n) = \begin{cases} 
\varphi_1 & \text{for } v \neq 0, \\
1 & \text{for } v = 0.
\end{cases} \]  
(27)
4.2.5 Kinetic 5

Setting the regulation strength for activators to \( a_i^A = 1 \) and for inhibitors to \( a_i^I = -\infty \), i.e., very strong effect inhibition effect, gives the regulation factor

\[
\phi_v^{(5)}(c_1, \ldots, c_n) = \prod_A \left[ 1 + \frac{x_i}{1 + \tilde{x}_i} \right] \prod_I \left[ 1 - \frac{x_i}{1 + \tilde{x}_i} \right],
\]

(28)

where the first product is over all activators and the latter over all inhibitors. This kinetic law term is already described by Mendes et al. (2003) and is used often for describing transcriptional regulation.

5 Examples

With GeNGe it is possible to model various transcriptional dynamics. Some examples are given in the following. The graphs show the topology of the gene regulatory network. A node represents the gene and the corresponding protein, which can act as a transcription factor. Green edges from a node1 to another node2 represents an activating transcriptional regulation of the protein (node1) on the gene (node2), a red edge with a bar represents an inhibitory effect of the protein (node1) on the transcription of the gene (node2). Nodes from a joint interaction set are connected by an black arrows to a joint node. From there an arrow points to the target node, the gene, which is regulated by the joint interaction set. The color indicates the kind of interaction (green for activation, red for inhibition.)

5.1 Saturation Effect and Transcription Activation Threshold

Pre-defined model in GeNGe: Two Node Example

Graph:

Mathematical Model:

Transcription functions:
\[ \phi^{\text{gene.000}} = 1 \]
\[ \phi^{\text{gene.001}}([\text{Protein}_{\text{gene.000}}]) = 1 + (2^2 - 1) \frac{[\text{Protein}_{\text{gene.000}}]^2}{N^2 + [\text{Protein}_{\text{gene.000}}]^2} \]

Degradation functions: linear
Translation functions: linear

Info: It is a very simple example to show, that with the used kinetic a saturation of the target gene gene.001 is reached while the concentration of the protein, corresponding to gene.000 increases permanently. Furthermore, by changing the normalization parameter \( N \) of the joint interaction set of gene gene.001 it is possible to delay the transcription of this gene. This parameter can be considered as a threshold parameter.

Time courses:

Normalization Factor \( N = 1 \) Normalization Factor \( N = 10 \) Normalization Factor \( N = 30 \)

5.2 Oscillation

Pre-defined model in GeNGe: Simple Oscillator

Graph:
Mathematical Model:

Transcription Functions:

\[
\phi^{\text{CI}}([\text{Protein}_{\text{tetR}}]) = 1 + \frac{1}{1 + [\text{Protein}_{\text{tetR}}]^2} \tag{31}
\]

\[
\phi^{\text{lacI}}([\text{Protein}_{\text{cl}}]) = 1 + \frac{1}{1 + [\text{Protein}_{\text{cl}}]^2} \tag{32}
\]

\[
\phi^{\text{tetR}}([\text{Protein}_{\text{lacI}}]) = 1 + \frac{1}{1 + [\text{Protein}_{\text{lacI}}]^2} \tag{33}
\]

Degradation functions: linear

Translation functions: linear

Info: This model of a simple oscillator is adapted from the repressilator by Elowitz and Leibler (2000), who designed and constructed a synthetic network in *Escherichia coli*.

Time courses:
blue line: control
red line: 80% knockdown of lacI
5.3 Bistability

Pre-defined model in GeNGe: Toggle Switch 1

Graph:

Mathematical Model:

Transcription Functions:
\[ \phi_{\text{Input}} = 0 \]  
\[ \phi_{\text{gene1}}([\text{Protein}_{\text{Input}}], [\text{Protein}_{\text{gene2}}]) = \left[ 1 + \frac{[\text{Protein}_{\text{Input}}]}{1 + [\text{Protein}_{\text{input}}]^2} \right] \cdot \left[ 1 + (2^{-5} - 1) \frac{[\text{Protein}_{\text{gene2}}]^2}{1 + [\text{Protein}_{\text{gene2}}]^2} \right] \]  
\[ \phi_{\text{gene2}}([\text{Protein}_{\text{gene1}}]) = 1 + (2^{-5} - 1) \frac{[\text{Protein}_{\text{gene1}}]^2}{1 + [\text{Protein}_{\text{gene1}}]^2} \]  

**Degradations functions:** linear  
**Translation functions:** linear

**Info:** This model reproduces a toggle switch. Gardner et al. (2000) constructed in *Escherichia coli* such a genetic toggle switch. In our model, the input signal by *Input* pushes the system to a steady state (*gene1* high and *gene2* low), where it remains, although the input is not any more present (left column of time course plots). If there is no input signal from *Input* (the initial values of the mRNA and protein are set zero), the system goes to a different steady state (*gene1* low and *gene2* high); see right column. The initial values of the mRNAs and proteins of *gene1* and *gene2* are the same in both situations. Consider the different scaling of the y-axes.

**Time courses:**

---

**Input on to off**

Image 1

**Input off**

Image 2
For the determination of the time scaling of the network generation, i.e. construction of the network topology, we generated ten networks of each of the network types motif network, random network, and scale-free network with different sizes, i.e. number of nodes, varying from 100 to 500 and averaged the values. Each node corresponds to one mRNA and one protein. For the network generation we used the default parameter settings. The time scaling (see Fig. 2) shows a slight quadratic behavior.

For the determination of the time scaling of the data generation, i.e. generating the mathematical model and solving the ODE system, we used from the above generated networks two networks of each of the network types and performed six time course simulations, respectively. We averaged over all simulations for networks with the same size. Once a mathematical model is generated, the ODE system is used for subsequent simulations. Therefore, we distinguish between model generation, i.e. set up of the ODE system and data generation, i.e. solving the mathematical system. The mathematical system contains twice as much...
Figure 3: Time scaling for model and data generation in GeNGe. Model generation is the set up of the ODE system, based on the network topology. During data generation the ODE is solved numerically for a specific parameter set.

equations as nodes, since for each node exists an mRNA and a protein. For the simulations, we used also the default parameters settings. The time scaling (see Fig. 3) shows for the model and data generation a quadratic behavior, respectively.

References

Aho, T., Smolander, O.-P., Niemi, J. and Yli-Harja, O. (2007) Rmbntoolbox: random models for biochemical networks. BMC Syst Biol, 1, 22.

Albert, R. and Barabási, A.-L. (2002) Statistical mechanics of complex networks. Rev. Mod. Phys., 74, 47–97.

Barabási, A.-L. and Oltvai, Z. N. (2004) Network biology: understanding the cell’s functional organization. Nat Rev Genet, 5, 101–113.

Basso, K., Margolin, A. A., Stolovitzky, G., Klein, U., Dalla-Favera, R. and Califano, A. (2005) Reverse engineering of regulatory networks in human B cells. Nat Genet, 37, 382–390.

Beal, M. J., Falciani, F., Ghahramani, Z., Rangel, C. and Wild, D. L. (2005) A bayesian approach to reconstructing genetic regulatory networks with hidden factors. Bioinformatics, 21, 349–356.

Bollobás, B., Borgs, C., Chayes, J. and Riordan, O. (2003) Directed scale-free graphs. In SODA ’03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, pp. 132–139. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
Chen, T., He, H. L. and Church, G. M. (1999) Modeling gene expression with differential equations. *Pac Symp Biocomput*, pp. 29–40.

de Jong, H. (2002) Modeling and simulation of genetic regulatory systems: a literature review. *J Comput Biol*, 9, 67–103.

den Bulcke, T. V., Leemput, K. V., Naudts, B., van Remortel, P., Ma, H., Verschoren, A., Moor, B. D. and Marchal, K. (2006) Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, 7, 43.

D’haeseleer, P., Wen, X., Fuhrman, S. and Somogyi, R. (1999) Linear modeling of mRNA expression levels during CNS development and injury. *Pac Symp Biocomput*, pp. 41–52.

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

Erdős, P. and Rényi, A. (1959) On random graphs. i. pp. 290–297.

Friedman, N., Linial, M., Nachman, I. and Pe’er, D. (2000) Using Bayesian Networks to Analyze Expression Data. *J Comput Biol*, 7, 601–620.

Friedman, N., Murphy, K. and Russell, S. (1998) Learning the Structure of Dynamic Probabilistic Networks. In Cooper, G. and Moral, S. (eds.), *UAI ’98: Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 139–147. Morgan Kaufmann, Madison, Wisconsin, USA.

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

Gardner, T. S. and Faith, J. J. (2005) Reverse-engineering transcription control networks. *Physics of Life Reviews*, 2, 65–88.

Grilly, C., Stricker, J., Pang, W. L., Bennett, M. R. and Hasty, J. (2007) A synthetic gene network for tuning protein degradation in saccharomyces cerevisiae. *Mol Syst Biol*, 3, 127.

Hache, H., Wierling, C., Lehrach, H. and Herwig, R. (2007) Reconstruction and validation of gene regulatory networks with neural networks. In *FOSBE ’07: Proceedings of the 2nd Foundations of Systems Biology in Engineering Conference*, pp. 319–324.

Hill, A. D., Tomshine, J. R., Weeding, E. M. B., Sotiropoulos, V. and Kaznessis, Y. N. (2008) Synbioss: the synthetic biology modeling suite. *Bioinformatics*, 24, 2551–2553.

Kamburov, A., Wierling, C., Lehrach, H. and Herwig, R. (2009) Consensuspathdb–a database for integrating human functional interaction networks. *Nucleic Acids Res*, 37, D623–D628.
Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T. and Yamanishi, Y. (2008) Kegg for linking genomes to life and the environment. *Nucleic Acids Res*, 36, D480–D484.

Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., Gerber, G. K., Hannett, N. M., Harbison, C. T., Thompson, C. M., Simon, I., Zeitlinger, J., Jennings, E. G., Murray, H. L., Gordon, D. B., Ren, B., Wyrick, J. J., Tagne, J.-B., Volkert, T. L., Fraenkel, E., Gifford, D. K. and Young, R. A. (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, 298, 799–804.

Liang, S., Fuhrman, S. and Somogyi, R. (1998) Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pac Symp Biocomput*, pp. 18–29.

Matthews, L., Gopinath, G., Gillespie, M., Caudy, M., Croft, D., de Bono, B., Gapaati, P., Hemish, J., Hermjakob, H., Jassal, B., Kanapin, A., Lewis, S., Mahajan, S., May, B., Schmidt, E., Vastrik, I., Wu, G., Birney, E., Stein, L. and D'Eustachio, P. (2009) Reactome knowledgebase of human biological pathways and processes. *Nucleic Acids Res*, 37, D619–D622.

Mendes, P., Sha, W. and Ye, K. (2003) Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, 19 Suppl 2, II122–II129.

Rangel, C., Angus, J., Ghahramani, Z., Lioumi, M., Sotheran, E., Gaiba, A., Wild, D. L. and Falciani, F. (2004) Modeling t-cell activation using gene expression profiling and state-space models. *Bioinformatics*, 20, 1361–1372.

Roy, S., Werner-Washburne, M. and Lane, T. (2008) A system for generating transcription regulatory networks with combinatorial control of transcription. *Bioinformatics*, 24, 1318–1320.

Schilstra, M. J. and Bolouri, H. (2002) The Logic of Gene Regulation. In *ICSB ’02: Proceedings of the 3rd International Conference on Systems Biology*, pp. 197–198. Karolinska Institutet, Stockholm, Sweden.

Schilstra, M. J. and Nehaniv, C. L. (2008) Bio-logic: gene expression and the laws of combinatorial logic. *Artif Life*, 14, 121–133.

Schäfer, J. and Strimmer, K. (2005) An empirical bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21, 754–764.

van Someren, E. P., Wessels, L. F. A., Backer, E. and Reinders, M. J. T. (2002) Genetic network modeling. *Pharmacogenomics*, 3, 507–525.

Werhli, A. V., Grzegorczyk, M. and Husmeier, D. (2006) Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical Gaussian models and Bayesian networks. *Bioinformatics*, 22, 2523–2531.

Wierling, C., Herwig, R. and Lehrach, H. (2007) Resources, standards and tools for systems biology. *Brief Funct Genomic Proteomic*, 6, 240–251.
Yu, J., Smith, V. A., Wang, P. P., Hartemink, A. J. and Jarvis, E. D. (2004) Advances to Bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, 20, 3594–603.