

# NETBUILDER' – LOGICAL MODELLING AND SIMULATING OF GENETIC REGULATORY NETWORKS

Katja Wegner<sup>1</sup>, Mark Robinson<sup>2</sup>, Maria J. Schilstra<sup>1</sup>

<sup>1</sup> Biocomputation, STRI, University of Hertfordshire, College Lane, UK-AL10 9AB Hatfield

<sup>2</sup> Department of Biochemistry and Molecular Biology, 422 Biochemistry Building, Michigan State University, East Lansing, MI 48824-1319

Genetic regulatory networks are important for understanding the control of processes in living cells. Since these networks are often complex and there is still a huge lack of knowledge, modelling and simulating techniques can be used to gain new insights. In this paper we will present how a genetic regulatory network can be modelled and simulated as a Petri Net by using NetBuilder'. NetBuilder' is a new software tool for constructing and studying multi-cellular genetic regulatory networks. It has an easy to use graphical user interface that enables users to create a layered Petri Net model. The constructed Petri Net model is automatically translated into a mathematical model and can be simulated straight away. NetBuilder' was developed to help researchers without any knowledge about modelling and simulation techniques to get started.

## 1. Introduction

Genetic regulatory networks (GRNs) control many biological processes by sampling the internal dynamics of a cell along with any external signals from its environment and surrounding cells [1]. For example, signal transduction pathways can transmit external signals to the nucleus where they are integrated and processed via a GRN. Many important biological processes are controlled by GRNs including those underlying many human genetic disorders [2].

Because these networks are typically very complex and detailed knowledge about them is still scarce, modelling and simulation techniques play an important role in our growing understanding of their functioning. *In-silico* techniques enable researchers to study GRNs in a flexible, cheap and fast way compared to *in-vitro* and *in-vivo* experiments. An extended overview about modelling and simulation methods for GRNs can be found in [3].

A new software tool, NetBuilder', was developed to facilitate the modelling and simulating of GRNs. Modelling and simulation tools often require a formal background which many researchers cannot be expected to have mastered. NetBuilder' provides of an easy to use graphical user interface for creating a model which is then automatically translated into a mathematical model. This mathematical model can subsequently be simulated using either default or user defined parameters. This design allows NetBuilder' to encapsulate and hide much of the complexities of the modelling process from the user. More advanced users are also able to define arbitrary rate equations by using the integrated equation editor.

NetBuilder' uses Petri Net notation to represent the network components. This notation provides an inherently intuitive way to model biochemical processes, and conveniently allows models to be abstracted at different levels. Petri Nets have been successfully used in the past to model several biological processes, e.g. sucrose breakdown pathway in the potato tuber [4].

## 2. NetBuilder'

NetBuilder' consists of a model designer, a simulation engine and a data handler, which manages data exchange and plotting functionality. These components and an example regulatory network will be explained in more detail in the following subsections. A typical user scenario will include the following steps:

1. Creation of the model.
2. Setting of the initial state and simulation of the model.
3. Visualisation and analysis of the resulting time series data.
4. Adjustment of model parameters and iteration from step 2 until the model shows the wanted behaviour.

### 2.1. Example – The Repressilator

The Repressilator is a synthetic regulatory circuit that was successfully introduced into *E.coli* to produce a functioning oscillatory network [5]. The system consists of three transcriptional repressor systems: each consisting of a gene, its mRNA product, and the encoded protein product. These three components are associated with three dynamic processes: transcription, translation and degradation. The protein product of each systems represses the transcription of one of the other systems (figure 1).

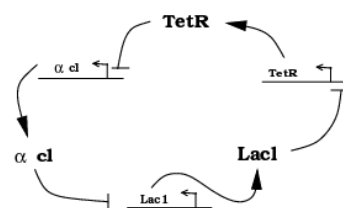


Figure 1. Schematic representation of the Repressilator.

The resulting network describes a negative feedback loop which can generate oscillatory behaviour under

a range of operational parameters.

This system is an excellent system for illustrating the fundamentals of modelling a gene regulatory network in NetBuilder' without many of the complexities and unknowns typically associated with a biologically derived regulatory network. The original study of the Repressilator network [5] explored the theoretical behaviour of the network using both deterministic and stochastic models. Only deterministic simulations of the model will be demonstrated here.

## 2.2. Modelling

A Petri Net is a bipartite graph with two types of nodes: places and transitions. Places represent molecular species; transitions represent processes such as gene transcription or degradation reactions. Places are containers for tokens (individual molecules). When a transition fires (i.e. a reaction takes place), input and output arcs transfer these tokens, thereby changing the state of the network. Arcs can only connect only two nodes of different types and thereby define the role of the connected place:

- An **input arc** connects a reactant place to a transition. Firing of the transition decreases the tokens of the reactant place by a stoichiometric amount (defined as weight of the arc).
- An **output arc** connects a transition to a product place. Firing of the transition increases the tokens of the product place by stoichiometric amount.
- A **modifier arc** represents a regulatory influence (excitatory or inhibitory) on a transition without changing the value of the connected place.

Transitions without input arcs act as sources: they generate but do not consume tokens. Transitions without output arcs act as sinks: they consume but do not generate tokens. Transitions without input or modifier arcs fire at a constant rate. A network can be separated into layers. Layers are containers for places, transitions, and other layers and can be used to create compartmental models.

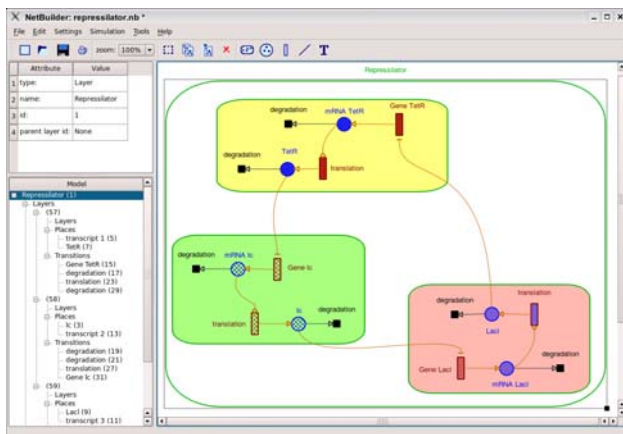


Figure 2. Screen shot of the NetBuilder' model design view.

A screen shot of NetBuilder' is shown in figure 2. The graphical user interface is separated into three parts. On the right hand side there is the drawing area for the construction of a model. Objects can be copied, deleted and resized and exhibit typical drag and drop functionality. Additional information about the model is displayed on the left hand side. Clicking on an object causes its attributes (e.g. type, name, id) to be displayed in the upper left part and its position in the model hierarchy to be displayed in the lower left part.

A place is, by default, represented by a circle, a transition by a rectangle, and a arcs by a directed arrow. The graphical properties of these objects are highly configurable. If an object is labelled, a text object is attached to the object which although it can be placed freely will always move in tandem with the associated object.

The drawing area in the screen shot (Figure 2) shows a Petri Net model of the Repressilator:

- The three transcription processes, one for each gene as well as the three translation processes are represented by source transitions.
- The six degradation processes, one for each mRNAs and protein, are represented by sink transitions.
- The three species of mRNA and three species of protein product are represented by places.

The three layers (yellow, green, and pink rectangular rounded rectangles) in figure 2 are containers for functionally related places and transitions. Output arcs connect the transcription processes to the places representing the associated mRNA, which, in turn, act as activators for the translation processes. The outputs of the translation transitions are the three proteins (TetR,  $\alpha$ lc or LacI), and each of these act as an inhibitor for one of the transcription processes. Proteins and mRNAs are constantly degraded, and this is achieved by connecting an input arc to a degradation transition. After the Petri Net model of the Repressilator has been constructed, its dynamics can be simulated.

## 2.3. Simulation

In addition to the style parameters, places, transitions and arcs have simulation specific attributes. Each place has a value, representing the (integer) number of tokens it contains, or another (continuous) value proportional to concentration or molar fraction, whereas each transition is associated with a rate constant.

The feature that makes NetBuilder' particularly suitable for the modelling of complex GRNs, and

distinguishes it from otherwise comparable modelling and simulation tools, is its arc grouping capability. Modifier arcs that impinge on one transition can be grouped into multiple functional sets, to synergistically combine their effects. An example is shown in figure 3, where three arcs are connected to a transition, and grouped into two sets.

The "exploded" transition view shows that the arcs connecting places A and C together form a synergistic **"and"** group. This group then combines additively (via an **"or"** operation) with the arc connecting B. Suppose A, B, and C represent transcription factors that activate (A, B) or repress (C) the transcription of a gene represented by the transition, the arc grouping shown in here indicates that transcription will only occur if A is present and B is absent, or if C is present (independently of the presence or absence of the other two). Various other operations (representing inter alia competitive interaction, [6]) are available to the user, each of which can have an unlimited number of inputs.

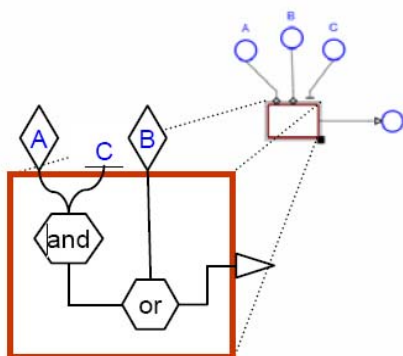


Figure 3. Transition with three modifier arcs grouped into two sets. The standard representation of the transition and its connections to places A, B (activators) and C (repressor) is in top right hand side quadrant; the bottom left hand quadrant of the figure shows an "exploded" view of the transition, in which the functional grouping of arcs is further specified.

Once the parameter values and the appropriate arc groupings and combinatorial operations have been specified, and the initial state has been set, the model is converted into a standard chemical kinetics model, consisting of a state vector (containing the place values), a rate vector (containing the instantaneous rate (or flux) for each transition), a stoichiometry matrix (constructed from the arc weights), and a list of automatically constructed functions (mass action, modified by the activator and inhibitor functions) to compute the instantaneous rates from the current state and the parameter values. This mathematical representation allows the user to generate time curves (by numerical integration), using stochastic (Gillespie algorithm [7], Gibson and Bruck [8] variant), or deterministic algorithms (using a fixed time step and simple explicit Euler integration, or the variable time step explicit and implicit methods available in the LSODE library [9]). NetBuilder' also allows the user to divide the set

of transitions into (arbitrary) stochastic and continuous subsets, and perform hybrid stochastic-continuous simulations (Gillespie first-reaction algorithm interwoven with numerical integration algorithms in the LSODE library, MJS, unpublished).

After running a deterministic simulation, the results are graphically displayed as a line plot. The default view is a time-concentration diagram in which the time steps are along the x-axis and the number of tokens of a place along the y-axis. It is also possible to create scatter-plots that show two time series against each other. These plots are useful to detect linear correlations.

The time-concentration plot of the simulation with the default parameter set (initial number of tokens = 0.0, rate constants and arc weights = 1.0) shows (figure 4) the system reaching steady state after an initial heavily damped oscillatory period.

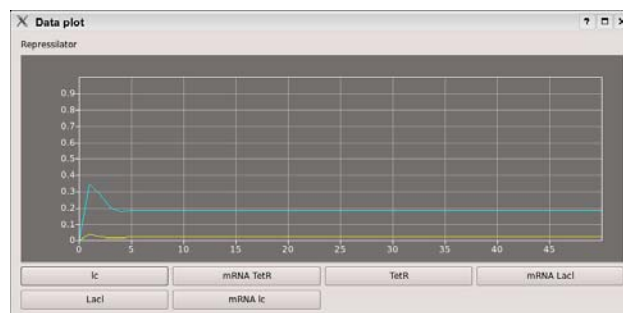


Figure 4. Deterministic simulation results of the Repressator model with the default parameter set. The buttons below the plot can be used to remove a curve from the plot or add a curve to the plot.

The three overlapping higher curves (blue) represent the identical concentrations of the three proteins and the overlapping lower curves (yellow) the concentrations of the three mRNAs. These results are expected if all rate constants of the transitions, all arc weights and all initial values of the places are equal.

## 2.4 Adjusting model parameters

The system is predicted to exhibit undamped oscillations under certain conditions [5], and to achieve this behaviour we need to take into account existing knowledge about parameters and initial values. To this aim, we increase the rate constant of the transcription processes from 1.0 to 30.0 and the multiplier parameter values of the incoming inhibition arcs to 40.0, translation processes to 20.0 and decrease the mRNA degradation to 0.5 and the protein degradation to 0.1 (the values derived in [5]). We also set the initial values of the mRNA LacI to 5.0 and mRNA  $\alpha$ lc to 1.0. These attributes can be easily edited in the initial state view that shows all simulation attributes in one dialog. The results of the simulation with these parameters are shown in figure 5.

With this set of parameters we do indeed observe oscillatory behavior. Notice that the concentrations of mRNA and protein achieved with this parameter set are significantly higher than those obtained with the defaults. This is a direct result of the greatly increased production/degradation ratios.

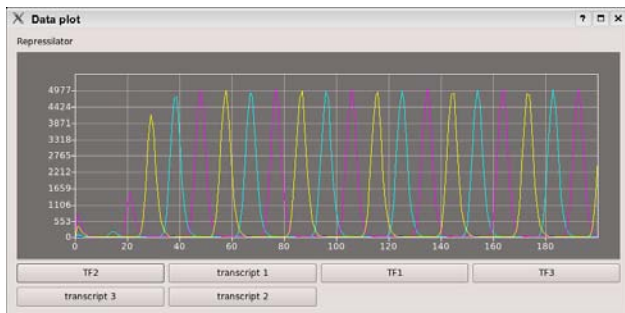


Figure 5. Simulation results obtained under conditions for which the Repressilator exhibits oscillatory behaviour.

## 2.5 Data Exchange

NetBuilder' has its own XML-based file format for saving the model and simulation settings. It also imports and exports the Systems Biology Markup Language (SBML [12]) which is supported by many software tools and enables users to exchange models between different software tools. Therefore, the user can create their models and run simulations in NetBuilder' and use analysis methods in any software tool that supports SBML. We also added the export of the Petri Net Markup Language (PNML [13]), so that any Petri Net tool which imports PNML can be used to analyse Petri net specific properties, such as boundedness, place invariants, or conflicting transitions. The drawing of the model can also be saved as an image (JPEG), whereas the simulation results may be saved as a table in an ASCII file in which each column contains the trajectory of a place and each row contains the values of the selected places at a particular time.

## 3. Conclusion

We have explained how a genetic regulatory network is modelled as a Petri Net and simulated in NetBuilder', using the Repressilator model as an example. As the Petri Net model is automatically translated into the appropriate mathematical formulation, users need to know very little about the mathematical background, and can quickly begin to experiment with their models by changing parameters. Thus, users can get a feeling for the essentials of modelling and simulation, before moving on to more complicated systems.

NetBuilder' is a new modelling and simulation software that is specialised on genetic regulatory networks using Petri Nets. There are many computational tools for the study of biological network dynamics, but we are not aware of any tool that is aimed specifically at the modelling and simulation of higher order regulatory interactions.

In the future, we intend to offer more analysis tools. A decomposition method for analyzing the complexity of GRNs and other regulatory networks is currently under development, as is an evolutionary algorithm for evolving artificial GRNs exhibiting specific behaviour [10]. NetBuilder' is freely available for different operating systems and can be downloaded from [14].

## Acknowledgements

The authors thank Ralph Gauges for his support and his provision of the tool Motato that was reused to build certain elements of NetBuilder's graphical user interface. This work was supported by a Welcome Trust Project Grant (ref. 072930/Z/03/Z).

## Literaturverzeichnis

- [1] Wiles J, Geard N, Watson J, Willadsen K, Mattick J, Bradley D, et al. There's more to a model than code: understanding and formalizing in silico modeling experience. In: GECCO '05: USA: ACM Press; 2005. 281-288.
- [2] Kahlem P, Birney E. Dry work in a wet world: computation in systems biology. *Molecular Systems Biology*. 2(2006), 40-44.
- [3] de Jong H, Modeling and simulation of genetic regulatory systems: a literature review, *J Comput Biol*, 9(2002), Nr. 1, 67-103.
- [4] Koch I, Junker BH, Heiner M. Application of Petri net theory for modelling and validation of sucrose breakdown pathway in the potato tuber. *Bioinformatics*, 21(2005), Nr.7,1219-1226.
- [5] Elowitz MB, Leibler SA. Synthetic gene oscillatory network of transcriptional regulators. *Nature*, 403(2000), 335-338.
- [6] Schilstra MJ, Nehaniv CL. Bio-Logic: Gene Expression and the Laws of Combinatorial Logic. *Artificial Life*. 14(2007), Nr. 1, 121-33.
- [7] Gillespie DT. Exact stochastic simulation of coupled chemical reactions. *J Phys Chem*. 81(1977), Nr. 25, 2340-2361.
- [8] Gibson MA, Bruck J. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J Phys Chem A*. 104(2000), 1876-1889.
- [9] Radhakrishnan K, Hindmars AC. Description and Use of LSODE, the Livermore Solver for Ordinary Differential Equations; 1993. NASA Reference Publication 1327. Lawrence Livermore National Laboratory Report UCRL-ID-113855.
- [10] Knabe JF, Nehaniv CL, Schilstra MJ. Genetic Regulatory Network models of Biological Clocks: Evolutionary history matters. *Artificial Life*, 14(2008), Nr. 1, 135-48.
- [12] Hucka M, et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*. 19(2003) Nr. 4, 524-531.
- [13] Billington J, Christensen S, van Hee KM, Kindler E, Kummer O, Petrucci L, et al. The Petri Net Markup Language: Concepts, Technology, and Tools. In: ICATPN, 2003, 483-505.
- [14] NetBuilder' web page, 2008. <http://strc.herts.ac.uk/bio/maria/Apostrophe>