Monika Heiner (Ed.)

Biological Processes & Petri Nets

5th International Workshop BioPPN 2014
Tunis, 23 June 2014
Proceedings

CEUR Workshop Proceedings
Volume 1159

Editor:

Monika Heiner
Brandenburg University of Technology Cottbus-Senftenberg
Computer Science Institute
Data Structures and Software Dependability
03013 Cottbus, Germany
monika.heiner@b-tu.de

Online available as CEUR Workshop Proceedings (ISSN 1613-0073), Volume 1159
http://CEUR-WS.org/Vol-1159/

BibTeX entry:

```
@proceedings{bioppn2014,
  editor    = {Monika Heiner},
  title     = {Proceedings of the 5th International Workshop on
               Biological Processes \& Petri Nets (BioPPN 2014),
               satellite event of Petri Nets 2014, Tunis, Tunisia, June 23, 2014},
  booktitle = {Biological Processes \& Petri Nets},
  location  = {Tunis, Tunisia},
  publisher = {CEUR-WS.org},
  series    = {CEUR Workshop Proceedings},
  volume    = {Vol-1159},
  year      = {2014},
  url       = {http://CEUR-WS.org/Vol-1159/}
}
```

Preface

This volume contains the peer-reviewed papers presented at BioPPN 2014 – the 5th International Workshop on *Biological Processes & Petri Nets* held on June 23, 2014 in Tunis as satellite event of PETRI NETS 2014.

The workshop had been organised to provide a platform for researchers aiming at fundamental research and real life applications of Petri nets in Systems and Synthetic Biology. Systems and Synthetic Biology are full of challenges and open issues, with adequate modelling and analysis techniques being one of them. The need for appropriate mathematical and computational modelling tools is widely acknowledged.

Petri nets offer a family of related models, which can be used as a kind of umbrella formalism – models may share the network structure, but vary in their kinetic details (quantitative information). This undoubtedly contributes to bridging the gap between different formalisms, and helps to unify diversity. Thus, Petri nets have proved their usefulness for the modelling, analysis, and simulation of a diversity of biological networks, covering qualitative, stochastic, continuous and hybrid models. The deployment of Petri nets to study biological applications has not only generated original models, but has also motivated research of formal foundations.

There were six submissions to the BioPPN workshop. One of the papers had been originally submitted to the PNSE workshop, and reviewed by the PNSE programme committee. Following the recommendation of these reviews, this paper has then been moved to the BioPPN workshop. In summary, each submission was reviewed by at least three, and on the average four, program committee members. The list of reviewers comprises 17 professionals of the field coming from 12 different countries and writing in total 28 reviews, most of them of substantial length.

The programme committees decided finally to accept five papers. The five accepted peer-reviewed papers (with an acceptance rate of 83%) involve 16 authors coming from seven different countries.

The program also includes one invited talk on *Mathematical models on cancer progression* given by Marco Beccuti from Università degli Studi di Torino. In summary, the workshop proceedings enclose theoretical contributions as well as biological applications, demonstrating the interdisciplinary nature of the topic.

For more details see the workshops website http://www-dssz.informatik.tu-cottbus.de/BME/BioPPN2014.

We acknowledge substantial support by the EasyChair management system, see http://www.easychair.org, during the reviewing process and the production of these proceedings.

June 20, 2014 Monika Heiner
Cottbus

This page is intentionally left blank.

Table of Contents

Program Committee

Gianfranco Balbo	University of Torino, Computer Science Department, Italy
Rainer Breitling	University of Manchester, Manchester Institute of Biotechnology, UK
Claudine Chaouiya	Insituto Gulbenkian de Ciência, Oeiras, Network Modelling Group, Portugal
Ming Chen	Zhejiang University, College of Life Sciences, Department of Bioinformatics, China
David Gilbert	Brunel University, Centre for Systems and Synthetic Biology, UK
Simon Hardy	Université Laval, Institut universitaire en santé mentale de Québec, Canada
Monika Heiner	Brandenburg University of Technology Cottbus-Senftenberg, Computer Science Institute, Germany
Mostafa Herajy	Port Said University, Mathematics and Computer Science Department, Egypt
Peter Kemper	College of William and Mary, Department of Computer Science, USA
Sriram Krishnamachari	Indraprastha Institute of Information Technology (IIIT), India
Chen Li	Zhejiang University, School of Medicine, Center for Genetic & Genomic Medicine, China
Fei Liu	Harbin Institute of Technology, Control and Simulation Center, China
Wolfgang Marwan	Otto von Guericke University Magdeburg & Magdeburg Centre for Systems Biology, Germany
Hiroshi Matsuno	Yamaguchi University, Graduate School of Science and Engineering, Japan
P.S. Thiagarajan	National University of Singapore, School of Computing, Department of Computer Science, Singapore

Mathematical models on cancer progression

Marco Beccuti

Università degli Studi di Torino,
Dipartimento di Informatica, Italy
beccuti@di.unito.it

Abstract. The Cancer Stem Cell (CSC) involvement into tumor progression, tumor recurrence, and therapy resistance is one of the most studied subject of current cancer research [10,4,8,6]. Nevertheless, due to the complex dynamics characterizing the CSC tumor, a comprehensive theory has not been established yet. To this end, some advises can be obtained combining mathematical modeling and experimental data [5,9,2,7]. Indeed, mathematical modeling is a powerful instrument which may drive the comprehension of a biological system, providing a clear description of its essential dynamics.

The aim of this talk is hence to show how the CSC tumor growth could be described/studied through the application of mathematical models. In details two different modeling approaches are presented: the former one consists in a multilevel/multiscale model [1], which details both molecular and cellular aspects. By means of this framework we were able to reproduce the tumor growth trend observed in mice, highlighting the strong connection existing between cellular events and cell population dynamics. We were also able to reproduce molecular vaccinations, correctly miming the in vivo vaccinations in animals. However, this detailed approach can engender difficulties in the parameter estimation process when only few kinetic information is available.

The second contribution [3] was designed really to address this complexity issue. We defined a new compartmental mathematical framework only focusing on the cell subpopulation dynamics. Indeed, the aim of this work was to describe CSC tumor progression trying to identify its essential mechanisms at population level. Through a quantitative and qualitative analysis of our model was hence possible to define rules controlling the breast cancer progression.

Lastly, we point out that the CSC theory is applicable to several other human cancers. Therefore, being our two model based on the key dynamics of the CSC theory, they can be further adapted for the study of many other tumor cases too.

References

1. F Cordero M Beccuti C Fornari S Lanzardo L Conti F Cavallo G Balbo RA Calogero. Multi-level model for the investigation of oncoantigen-driven vaccination effect. *BMC Bioinformatics*, 14, 2013.
2. MD Johnston CM Edwards WF Bodmer PK Maini SJ Chapman. Mathematical modeling of cell population dynamics in the colonic crypt and in colorectal cancer. *PNAS*, 104, 2007.

M. Heiner (Ed.): BioPPN 2014, a satellite event of PETRI NETS 2014,
CEUR Workshop Proceedings Vol. 1159, 2014.

3. C Fornari M Beccuti S Lanzardo L Conti G Balbo F Cavallo RA Calogero F Cordero. A mathematical-biological joint effort to investigate the tumor-initiating ability of cancer stem cells. *PloS Computational Biology*, under review.
4. JE Visvader GJ Lindeman. Cancer stem cells in solid tumours: accumulating evidence and unresolved questions. *Nature Review Cancer*, 10, 2008.
5. R Molina-Pena and M Álvarez. A simple mathematical model based on the cancer stem cell hypothesis suggests kinetic commonalities in solid tumor growth. *Plos One*, 7, 2012.
6. R Pardal MF Clarke SJ Morrison. Applying the principles of stem-cell biology to cancer. *Nature Review Cancer*, 3, 2003.
7. F Michor Y Iwasa MA Nowak. Dynamics of cancer progression. *Nature Reviews Cancer*, 4, 2004.
8. R Bjerkvig BB Tysnes KS Aboody J Najbauer AJA Terzis. The origin of the cancer stem cells: current controversis and new insights. *Nature Review Cancer*, 5, 2005.
9. A Marciniak-Czochra T Stiehl W Wagner. Characterization of stem cells using mathematical models of multistage cell lineages. *Math Comp Model*, 2010.
10. MF Clarke JE Dick PB Dirks CJ Eaves CH Jamieson DL Jones J Visvader IL Weissman GM Wahl. Cancer stem cells' perspectives on current status and future directions: Aacr workshop on cancer stem cells. *Cancer Res*, 66, 2006.

Overcoming unknown kinetic data for quantitative modelling of biological systems using fuzzy logic and Petri nets

Jure Bordon, Miha Moškon, Miha Mraz

University of Ljubljana,
Faculty of Computer and Information science,
Slovenia,
jure.bordon@fri.uni-lj.si

Abstract. Biological system modelling is used to guide experimental work, therefore reducing the time and cost of *in vivo* implementation of newly designed systems. We introduce an improved modelling method, based on fuzzy logic and Petri nets. By using fuzzy logic to linguistically describe a biological process, we avoid the necessity to use kinetic rates, which are often unknown. We introduce a new set of transition functions to enable the use of our method with existing Continuous Petri nets. With this we achieve the extension of usability and applicability of current Continuous Petri nets definition even for biological systems for which exact kinetic data are unknown. We demonstrate the contribution of our approach by using it to model the translation in a simple transcription-translation system. We compare the results obtained to the results of exiting ODE approaches.

Keywords: modelling biological systems, missing kinetic data, ODE, fuzzy logic, Petri nets

1 Introduction

Advances in synthetic biology are consistently opening new possibilities for the design and construction of complex biological systems. Because *in vivo* design is costly and time-consuming, various modelling methods can be used to check whether the desired behaviour of the system is achievable *in silico* first [1–3]. Furthermore, modelling enables us to test in what way small or substantial changes to the design of our system affects its behaviour and potentially change the design before implementing it *in vivo*. Which modelling method to use depends on the size of the system, the desired accuracy of simulation results and whether accurate kinetic rates, which determine system's dynamics, are known [4]. We usually describe a biological system as a set of chemical species, which are connected by interactions (chemical reactions). Once we define the desired behaviour of our system by choosing the right chemical species and interactions among them, the first step is to build a model. While existing deterministic and stochastic quantitative approaches [5–9] can produce a detailed prediction of system behaviour

M. Heiner (Ed.): BioPPN 2014, a satellite event of PETRI NETS 2014,
CEUR Workshop Proceedings Vol. 1159, 2014.

and therefore reduce the time and cost of such design, they heavily rely on kinetic rates. In synthetic biology biological systems are usually newly designed and the exact details of interactions (kinetic rates) are often unknown [10]. Consequently, existing quantitative approaches can only be used to build a model of a limited set of biological systems [11]. We can use parameter estimation techniques to extract kinetic rates from experimental data. However, due to the complexity of interactions, we often need to establish strict limitations on parameter values in order to get biologically relevant and realistic parameters [12]. The diagram on Figure 1 presents the role of modelling in designing a new biological system. With existing methods the first step of the design process presented on the diagram is only possible when we are building a model with well characterized parts (left side), while our approach can be used for modelling biological systems in the same way even if accurate kinetic data is unknown (right side). Existing methods are often used within the framework of Petri nets, a formalism that has been extended to suit the needs for continuous deterministic and stochastic approaches [13].

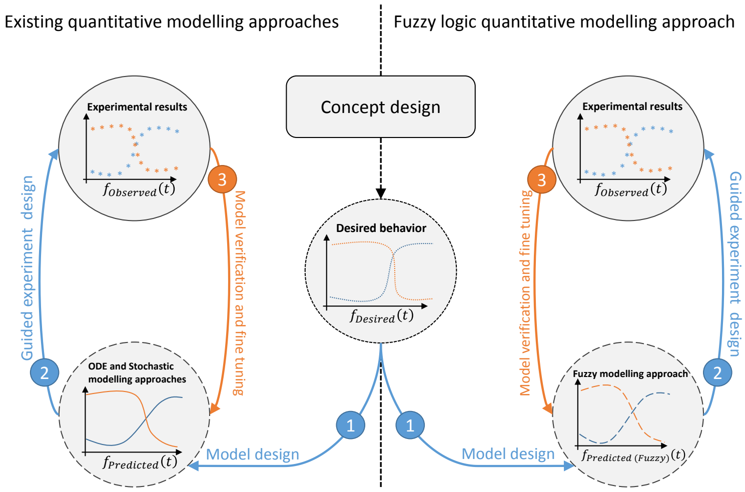


Fig. 1. Sequence of steps which can significantly reduce the time and cost of *in vivo* implementation. Because accurate kinetic data is needed for the first step, existing approaches are often not usable (left side). Proposed modelling method uses the same paradigm for model building (right side), but can be used even when accurate kinetic rates are unknown.

Similarly to quantitative Petri nets, fuzzy logic Petri nets have been established as a very promising modelling approach for qualitative analysis of biological systems. Fuzzy logic uses linguistic terms and rules for system behaviour description, allowing intuitive design and model construction. It has been applied to several research areas such as: extracting activator/repressor relationship from micro-array data [14, 15], searching for basic motifs in unknown gene regulatory networks (positive/negative feedback loops, degradation, ...) [16] and qualitative description of gene regulation [17]. Additionally, in [18] authors show that fuzzy logic can serve as an alternative but more intuitive approach for modelling biological systems. In their work they apply fuzzy logic and Petri nets to quantitative modelling of biological systems and successfully demonstrate that Hill, Michaelis-Menten and mass-action functions can be approximated by fuzzy logic systems if kinetic data is available. In this paper we propose an improved modelling method that builds on established fuzzy logic and Petri nets approaches but further extend its uses to allow us to obtain quantitative results even when kinetic data is unknown. We inherit existing continuous Petri net definition and extend it to include necessary transition functions for our fuzzy approach. In addition, we can use the proposed method only for parts of the system where kinetic data is unknown, while preserving the accuracy of ODEs in other parts. Because the proposed method is based on linguistic description of the processes, we can use rough estimations and general knowledge about the process to obtain quantitative results. Rough estimations can be obtained by observing existing systems with similar chemical species [19, 20]. Even though we use these estimations and consequently obtain less accurate simulation results, they are still comparable to results of existing methods, are biologically relevant and can be used to guide experimental work.

This paper is organized as follows: in Section 2 we present the basics of fuzzy logic modelling and how fuzzy logic is used in the Petri net framework. In Section 3 we demonstrate the proposed method by constructing a model of translation in a simple transcription-translation system, in Section 4 simulation results obtained with ODE and proposed method are compared and in Section 5 we summarize what the main contribution of the method is and give some directions for future research.

2 Petri Nets as a Framework for Fuzzy Logic

2.1 Fuzzy Logic as a Modelling Approach

Fuzzy logic uses linguistic terms and rules to describe current system state and how the state of the system changes over time [21, 22]. Numerical (crisp) values, which are used for presenting chemical species' concentrations, are converted to fuzzy values. Fuzzy values are given by linguistic terms, presented as membership degree to *fuzzy sets*, such as *Low*, *Medium* and *High*. Conversion from crisp to fuzzy value is performed with *fuzzification* rules, which include the definitions and number of fuzzy sets and the shapes and positions of their membership functions. While a membership function can have arbitrary shape and position, the most commonly used functions are simple triangular [23]. In order to simulate system change and obtain fuzzy value of output variables, IF-THEN rules are

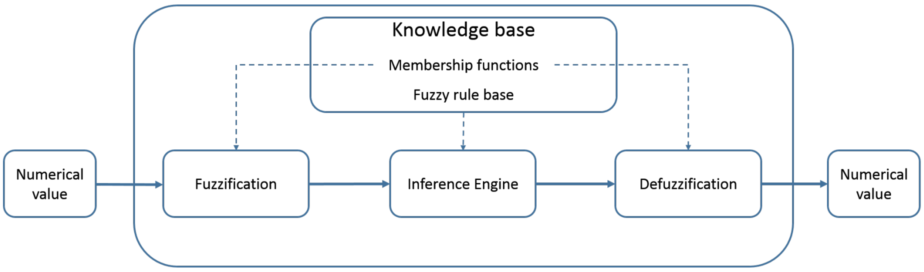


Fig. 2. Fuzzy logic modelling. Input variables are first converted to fuzzy variables by fuzzification. Once we have input fuzzy values, IF-THEN rules are applied to obtain output fuzzy values. Output variable is then defuzzified to obtain the crisp value. This sequence of steps can easily be translated to a Petri net.

applied to input fuzzy variables. Example of such rule is `IF` x `is High THEN` y `is Low`, where x is the input variable and y is the output variable. Since biological processes often have more than one input, we will need to use rules that combine the effect of input variables with operators AND and OR. An example of such rule is `IF` x_1 `is High AND` x_2 `is Low THEN` y `is Low`, where x_1 and x_2 are input variables and y is the output variable. Final step of fuzzy logic reasoning is obtaining crisp value of output variable, which is termed *defuzzification* and is performed by applying center-of-gravity (COG) method. Figure 2 shows these three steps as a sequence of actions: fuzzification, applying IF-THEN rules and defuzzification. Fuzzy logic can be used to intuitively model biological processes. IF-THEN rules are used to describe underlying dynamics where input variables are presented by current concentrations of chemical species and output variables define changes of concentrations. If we augment this description with rough estimation of reaction speed and therefore introduce the component of time, we can obtain quantitative results. In addition, the sequence of three steps can be efficiently used within the Petri net framework [24, 25].

2.2 Fuzzy Logic and Petri Nets

By using Petri net formalism it is possible to intuitively build the Petri net graph of the model. Once the Petri net is constructed using different modelling methods is easy. We only need to change the underlying transition function and firing rules. Continuous Petri nets use real numbers in places (marking values), meaning that transitions also no longer consume and produce whole tokens, but instead change the marking of an input or output place by a real value. New marking values in places are calculated by adding the contribution from input transitions and subtracting the value that is consumed due to output transitions. This allows a continuous flow throughout the Petri net, which can be used to present a system of ODEs [13]. Similar approach is used with the proposed fuzzy logic modelling method. Input and output of fuzzy part is identical to that of existing continuous Petri net [26]. However, when using fuzzy logic, we first need to fuzzify the input variable (additional transition function) and calculate the membership to each defined fuzzy set. By applying the defined IF-THEN rules (one transition for each rule), we obtain the fuzzy value of output variable and then defuzzify (center-of-gravity transition function) it to obtain the crisp value. We use this crisp value to change the marking of a place the same way we do in continuous Petri nets, by adding or subtracting a real value. We will use existing continuous Petri net definition from [26]. We will add a new set of functions that are needed for fuzzy logic. This set of functions will include fuzzification functions, functions for applying IF-THEN rules and defuzzification function. Existing definition $PN_{Continuous} = \langle P, T, f, v, m_0 \rangle$ is extended by a set of functions $v_{fuzzy} = (f_{fuzzification}, f_{defuzzification}, f_{IF-THEN})$. Functions in $f_{fuzzification}$ define how we obtain fuzzy value from an input crisp value. An example of such function is a triangular membership function for a fuzzy set A:

$$\mu_A(x) = \begin{cases} \frac{x-a}{b-a} & a \leq x \leq b, \\ \frac{c-x}{c-b} & b \leq x \leq c, \\ 0 & otherwise, \end{cases} \tag{1}$$

where x is the crisp value of the input variable and parameters a, b, c the x-coordinates of triangle vertices that define the shape of membership function. Function $f_{defuzzification}$ gives us the opposite rule and defines how we obtain crisp value from fuzzy value by applying the center-of-gravity method (COG).

$$y = \frac{\sum_{i=1}^{n} \overline{y_i} \cdot \mu[i]}{\sum_{i=1}^{n} \mu[i]}, \tag{2}$$

where y is the crisp value, $\overline{y_i}$ x-coordinate at which membership function of fuzzy set i has the highest possible degree of membership (parameter b from Eqn. 1) and $\mu[i]$ current degree of membership for fuzzy set i. Output fuzzy value is obtained by applying IF-THEN rules to the input variables. With basic (one input and one output) IF-THEN rules $f_{IF-THEN}$ is simple. If we have an input variable x, an output variable y and a rule `IF x is Low THEN y is High`, x membership degree to its fuzzy set `Low` is assigned to y membership degree to its fuzzy set `High`. This process is then repeated for all rules to obtain fuzzy value of y. However, biological processes usually have more than one input chemical species, therefore we need to use rules with more than one input variable. When applying IF-THEN rules with more than one input variables we usually define the rules using operator `AND`, which acts as a function $Min(\mu_1[i], \mu_2[i], ..., \mu_n[i])$, where $\mu_j[i]$ is a membership degree of variable j to its fuzzy set i. If we have two input variables x_1, x_2, an output variable y and a rule `IF` x_1 `is Low AND` x_2 `is High THEN` y `is High`, y degree of membership to its fuzzy set `High` would be assigned as a lower of the two values: x_1 degree of membership to its fuzzy set `Low` and x_2 membership degree to its fuzzy set `High`, which we can also note as $\mu_y[High] = Min(\mu_{x_1}[Low], \mu_{x_2}[High])$. Once we define the set of these three types of functions (fuzzification, defuzzification, IF-THEN rules), we have all the tools needed to construct a fuzzy Petri net model of a biological process.

3 Simple Transcription-Translation System: Modelling Translation With Fuzzy Logic and Petri Nets (case study)

We present model construction using proposed method on a simple transcription-translation system introduced in [18] to verify a qualitative modelling technique by qualitatively comparing its results with the results of an ODE approach. This system consists of 5 chemical species: DNA, mRNA, Transcription resource (TsR), Translation resource (TlR) and protein (GFP). The dynamics of the system are governed by transcription (TsR consumption, mRNA production), translation (GFP production) and the decay of mRNA and TlR as shown on Figure 3.

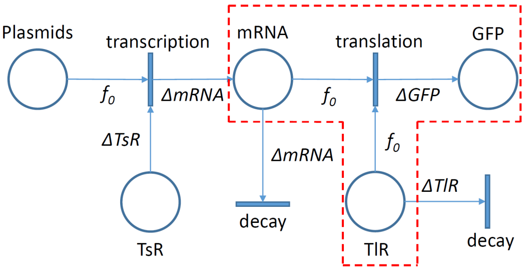


Fig. 3. Petri net of a simple transcription-translation system. We use fuzzy approach on the marked part of the Petri net (translation). Our simulations will observe how concentration of GFP changes over time, if we insert DNA at different time points. The limiting factors for system stability are limited amount of transcription and translation resources: TsR consumption and TlR degradation.

We will adopt the ODE model of this system from [18]. It is defined as the following set of differential equations:

$$\frac{d[mRNA]}{dt} = \frac{k_{ts} \cdot [TsR] \cdot [DNA]}{m_{ts} + [DNA]} - \delta_{mRNA} \cdot [mRNA], \tag{3}$$

$$\frac{d[TsR]}{dt} = -\frac{k_{TsR} \cdot [TsR] \cdot [DNA]}{m_{ts} + [DNA]}, \tag{4}$$

$$\frac{d[GFP]}{dt} = \frac{k_{tl} \cdot [TlR] \cdot [mRNA]}{m_{tl} + [mRNA]} - k_{mat} \cdot [GFP], \tag{5}$$

$$\frac{d[TlR]}{dt} = -\frac{\delta_{TlR} \cdot [TlR]}{m_{TlR} + [TlR]}. \tag{6}$$

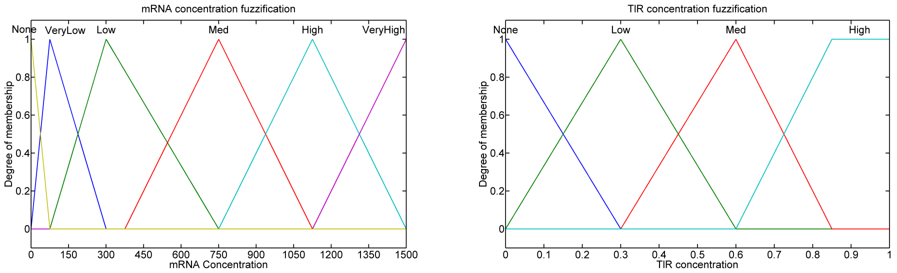


Fig. 4. Membership functions for input variable fuzzy sets: mRNA (left) is described by 6 fuzzy sets - None, VeryLow, Low, Medium, High and VeryHigh - and TlR (right) by 4 fuzzy sets - None, Low, Medium and High.

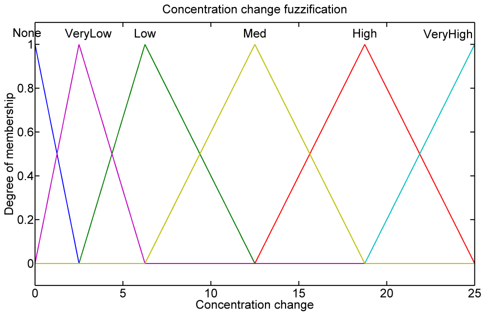


Fig. 5. Membership functions for output variable (concentration change) fuzzy sets - None, VeryLow, Low, Medium, High, VeryHigh.

The ODE model from [18] assumes that concentration of TlR and mRNA do not change as the result of translation. mRNA concentration increases as a result of transcription and only decreases as a result of degradation. Additionally, TlR concentration also only decreases as a result of degradation. To verify the proposed method, we will assume that k_{tl} and/or m_{tl} from Eqn. (5) are unknown when constructing the fuzzy logic model. We evaluate the fuzzy logic approach by constructing a fuzzy Petri net model of translation, replace the translation part of Eqn. (5) with our fuzzy description as shown on Figure 3 and compare the simulation results to the initial ODE model. First step in constructing a fuzzy logic model is to define membership functions for fuzzification and defuzzification of our input variables (concentration of mRNA and TlR) and output variable (concentration change of GFP). Membership functions we use for both input and output variable fuzzy sets are shown on Figures 4 and 5.

According to [27] we assume that mRNA concentration is the strongest factor of maximum translation speed (maximum change in concentration). TlR therefore reaches highest possible contribution before reaching its maximum concentration, while on the other hand even small amounts of mRNA should result in GFP concentration change.

When defining membership functions for output variables, we need to take into account the rough estimation of translation speed. Our rough estimation is based on data from different biological systems, using different chemical species. Considering translation rate, maximum concentration of mRNA and type of chemical species from [18, 19, 27, 28], our rough estimation is that the maximum change in concentration of a protein as a result of translation is $25nM/min$. How input variables affect output variable is defined by the IF-THEN rules shown in Table 1.

Table 1. The set of rules that defines how input variables affect output variable. If either of the input variables is None change in concentration will also be None. In all other cases, increasing both input variable concentration will increase the change in concentration of GFP, reaching highest change when both inputs are at their highest values.

TlR \ *mRNA*	None	VeryLow	Low	Med	High	VeryHigh
None	None	None	None	None	None	None
Low	None	VeryLow	Low	Low	Low	Med
Med	None	VeryLow	Low	Med	Med	High
High	None	VeryLow	Low	Med	High	VeryHigh

IF-THEN rules are defined so they reflect the descriptive knowledge we have about translation. The more there is of either mRNA or TlR, the higher the change in concentration of GFP should be; if one of the inputs is low, change in concentration changes accordingly; if any of the inputs is missing, there should not be any concentration change, etc. Once we obtain the fuzzy value of our concentration change by applying IF-THEN rules, we need to defuzzify it in order to get a crisp value, which we can then use in calculating the new concentration of the GFP. Fuzzy output is translated into a crisp value according to the Eqn. (2). This crisp value is then used just as it would have been if it was a result of a step in numerical solving of system of ODEs. We constructed the Petri net for our fuzzy description of translation as a series of three steps - fuzzification, applying IF-THEN rules and defuzzification (Figure 6). We can use this PN to replace the translation transition from Figure 3, if parameter values for Eqn. (5) are unknown.

Using this constructed Petri net, we will observe how concentration of GFP changes over time and when it reaches its maximum value if we add the DNA at different time points and compare the simulation results to the ODE model.

4 Results

Both ODE and Fuzzy logic models were built in MATLAB Simulink. Petri nets serve as a powerful framework for both approaches, however computing underlying numerical solutions can be done by an external engine like MATLAB. We

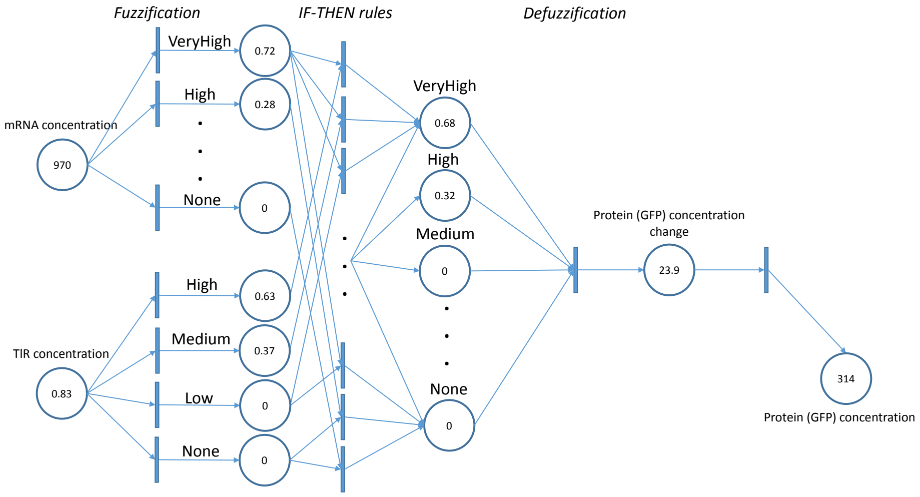


Fig. 6. Translation model using fuzzy logic and Petri nets. Inputs and output of this model are the same as with ODE: mRNA concentration, TlR concentration and change in concentration of the protein (GFP). This Petri net can be inserted into 3 to get the full model of the system. For reactions other than translation ODEs are used (Eqns. (3),(4),(6) and degradation part of Eqn. (5).

used MATLAB Simulink built-in ode4 (Runge-Kutta) solver and set the simulation time to 1000 minutes with a 0.1 minute fixed time step. Initial concentrations of both TsR and TlR were set to 1 nM, all others were set to 0 nM. During the simulation we inserted 3.4 nM of DNA at 6 different time points (six different simulations with same initial concentrations): 0 minutes, 37 minutes, 73 minutes, 112 minutes, 153 minutes and 187 minutes (these concentrations and time points were chosen according to [18] in order to make comparison of simulation results relevant). To avoid discontinuity of ODE solving, the input and output of the fuzzy component is evaluated for every step of the simulation. This slightly increases computation time of the simulation. Figure 7 shows simulation results of two different models.

Simulation results from both models show that the plateau of protein concentration is reached at the same time (at about 200 minutes) which is the result of translation resource degrading to 0, stopping translation entirely. Since we did not include protein degradation, its concentration stays unchanged for the remaining time of simulation. We see that even though we described translation with fuzzy approach we still get comparable quantitative results. The error introduced due to using rough estimation of translation speed instead of exact translation rate is noticeable. However, we did not use any exact parameter values for translation with the proposed method and still managed to obtain quantitatively and biologically relevant results, which are comparable to those obtained with (strict) ODE approach. In addition, because we only changed how

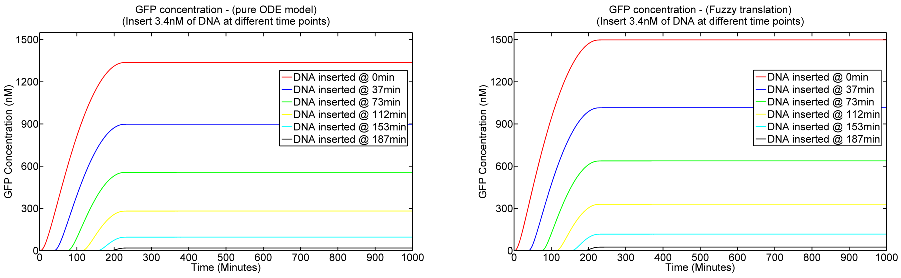


Fig. 7. We observed how GFP concentration changes if we add 3.4 nM of DNA at different points when using strict ODE model (left) and when using the proposed fuzzy approach for modelling translation (right).

we model translation, trajectories for other processes stay unchanged. Simulation results indicate that fuzzy logic is a viable modelling approach even when kinetic data is unknown. By exploiting information we have about the system for similar models and biological systems, we can successfully build a quantitative model even when accurate parameters are unknown. By using our approach with Petri nets, we can easily change the underlying description of a process for which kinetic data is unknown while preserving accuracy of ODEs for the parts of system where it is possible.

5 Summary

We presented the Fuzzy logic approach for modelling biological processes, which avoids using exact kinetic data. Proposed method uses a rough estimation of process dynamic to obtain quantitative simulation results. This estimation is extracted from existing base of knowledge about modelling biological processes by inspecting similar systems and chemical species. With introducing this method to Petri nets we managed to further extend their usability and applicability to continuous approaches, even when kinetic data is unknown. We showed its uses on a simple transcription-translation system by substituting the ODE translation description with the proposed fuzzy approach, achieving quantitatively and biologically relevant results, without using exact kinetic data. Adding additional functions for fuzzification, application of IF-THEN rules and defuzzification increases the complexity of the Petri net model. However, these functions are very simple and can be evaluated the same way that ODEs are. In addition, these three stages of fuzzy logic are repeated for every process for which we use the proposed approach and while we need to manually define fuzzy sets, membership functions and IF-THEN rules, once those are defined we could generate the Fuzzy Petri net automatically. The number of transitions and edges for fuzzification and defuzzification stages are defined by the number of fuzzy sets, while the functions for these transitions are defined by the shape of membership functions. Number of edge and transition functions in IF-THEN rule stage are defined by

IF-THEN rules (e.g. IF x_1 is High AND x_2 is Low THEN y is Low would generate a transition with two input edges - from places $x1_{High}$ and $x2_{Low}$ - and one output edge - to place y_{Low}; the function in the transition would be Min(Input 1, Input 2)). Moreover, we could use hierarchical Petri net structure, where top level would resemble the Petri net shown on Figure 3, while the fuzzification, IF-THEN rules and defuzzification stages (Figure 6) would be presented as a lower level Fuzzy Petri net that describes all three stages as one transition (in our case translation). Our future research also includes using this approach on a more complex system and observe how inaccuracy of our rough estimation changes the overall trajectory of concentrations. We would also like to consider using experimental data for fine tuning our estimations, which would bring the accuracy of simulation results even closer to those of existing methods.

Acknowledgement

Results presented here are in scope of PhD thesis that is being prepared by Jure Bordon, University of Ljubljana, Faculty of Computer and Information science.

References

1. Kitney, R., Freemont, P.: Synthetic biology - the state of play. FEBS Lett **586**(15) (July 2012) 2029–2036
2. Chen, Y.Y., Galloway, K.E., Smolke, C.D.: Synthetic biology: advancing biological frontiers by building synthetic systems. Genome biology **13**(2) (2012) 240
3. Chen, L., Wang, R.: Designing gene regulatory networks with specified functions. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-I: REGULAR PAPERS **53** (2006) 2444–2450
4. Alon, U.: An Introduction to Systems Biology: Design Principles of Biological Circuits. Chapman and Hall/CRC mathematical & computational biology series. Chapman & Hall/CRC (2007)
5. de Jong, H.: Modeling and simulation of genetic regulatory systems: A literature review. Journal of Computational Biology **9** (2002) 67–103
6. Weiss, J.N.: The hill equation revisited: uses and misuses. FASEB Journal **11** (1997) 835–841
7. Gillespie, D.T.: Stochastic simulation of chemical kinetics. Annu. Rev. Phys. Chem. **58** (2007) 35–55
8. Andrianantoandro, E., Basu, S., Karig, D.K., Weiss, R.: Synthetic biology: new engineering rules for an emerging discipline. Molecular Systems Biology **May** (2006) 1–14
9. Cheng, A., Lu, T.K.: Synthetic biology: An emerging engineering discipline. Annu. Rev. Biomed. Eng. **14** (2012) 155–178
10. Sun, J., Garibaldi, J.M., Hodgman, C.: Parameter estimation using metaheuristics in systems biology: a comprehensive review. Computational Biology and Bioinformatics, IEEE/ACM Transactions on **9**(1) (2012) 185–202
11. Heath, A.P., Kavraki, L.E.: Computational challenges in systems biology. Computer Science Review **3**(1) (2009) 1–17

12. Lillacci, G., Khammash, M.: Parameter estimation and model selection in computational biology. PLoS computational biology **6**(3) (2010) e1000696
13. Heiner, M., Gilbert, D., Donaldson, R.: Petri nets for systems and synthetic biology. Formal methods for computational systems biology (2008) 215–264
14. Hamed, R.I., Ahson, S., Parveen, R.: A new approach for modelling gene regulatory networks using fuzzy Petri nets. Journal of Integrative Bioinformatics **7**(1) (2010) 113
15. Maraziotis, I.A., Dragomir, A., Thanos, D.: Gene regulatory networks modelling using a dynamic evolutionary hybrid. BMC Bioinformatics **11(140)** (2010) 1–17
16. Küffner, R., Petri, T., Windhager, L., Zimmer, R.: Petri nets with fuzzy logic (pnfl): reverse engineering and parametrization. PLoS One **5**(9) (2010) e12807
17. Gendrault, Y., Madec, M., Wlotzko, V., Lallement, C., Haiech, J.: Fuzzy logic, an intermediate description level for design and simulation in synthetic biology. In: Biomedical Circuits and Systems Conference (BioCAS), 2013 IEEE, IEEE (2013) 370–373
18. Windhager, L.: Modeling of dynamic systems with Petri nets and fuzzy logic. PhD thesis, Ludwig-Maximilians-Universität München (April 2013)
19. Tigges, M., Marquez-Lago, T.T., Stelling, J., Fussenegger, M.: A tunable synthetic mammalian oscillator. Nature **457** (2009) 309–312
20. Fung, E., Wong, W.W., Suen, J.K., Bulter, T., Lee, S.g., Liao, J.C.: A synthetic gene–metabolic oscillator. Nature **435**(7038) (2005) 118–122
21. Zadeh, L.: Fuzzy logic and approximate reasoning. Synthese **30**(3) (1975) 407–428
22. Zadeh, L.: Fuzzy logic= computing with words. Fuzzy Systems, IEEE Transactions on **4**(2) (1996) 103–111
23. Klir, G.J., Yuan, B.: Fuzzy sets and fuzzy logic. Prentice Hall New Jersey (1995)
24. Pedrycz, W., Gomide, F.: A generalized Fuzzy Petri net model. Fuzzy Systems, IEEE Transactions on **2**(4) (1994) 295–301
25. Hamed, R.: Intelligent method of Petri net formal computational modeling of biological networks. In: Computer Science and Electronic Engineering Conference (CEEC), 2013 5th. (Sept 2013) 162–167
26. Gilbert, D., Heiner, M.: From Petri Nets to Differential Equations - An Integrative Approach for Biochemical Network Analysis. In Donatelli, S., Thiagarajan, P., eds.: Petri Nets and Other Models of Concurrency - ICATPN 2006. Volume 4024 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2006) 181–200
27. Atkinson, M.R., Savageau, M.A., Myers, J.T., Ninfa, A.J.: Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in¡ i¿ escherichia coli¡/i¿. Cell **113**(5) (2003) 597–607
28. Michael B. Elowitz, S.L.: A synthetic oscillatory network of transcriptional regulators. Nature **403** (2000) 335–338

A multi-scale extensive Petri net model of the bacterial–macrophage interaction

Rafael V. Carvalho, Jetty Kleijn, Fons J. Verbeek

Leiden Institute of Advanced Computer Science, Leiden University, Niels Bohrweg 1,
2333 CA Leiden, The Netherlands
{r.carvalho; h.c.m.kleijn; f.j.verbeek}@liacs.leidenuniv.nl

Abstract. ***Mycobacterium tuberculosis*** **is considered one of the most efficient intracellular pathogens responsible for chronic infection, resulting in over 1.3 million of deaths a year. Exploring the host-cell signalling pathways, the bacteria evade host immune responses and enhance the infection inside the macrophage. Understanding how the bacteria interact with the immune system is an important step in the development of new therapies for** ***mycobacterium*** **pathogen. The aim of this paper is to present a prototype draft of a Petri Net model that highlights the interference strategies used by mycobacteria to achieve intracellular survival. The hierarchical model presents an overview of the important host-cell signalling pathways that occur at multiple (molecular, intracellular and intercellular) scales.**

Keywords: mycobacterial infection, host-cell signalling pathways, extended Petri Net, multi-scale modelling

1 Introduction

Tuberculosis (TB) is the second greatest killer disease worldwide due to a single infectious agent: *mycobacterium tuberculosis* (Mtb) [1]. Effective vaccination against tuberculosis is a challenge; a better understanding of the host-pathogen relationship provides an important key for new treatments. The host innate immune response is the first line of defence against invading microbes. It recognises the pathogen in the first stage of infection and initiates an appropriate immune response. Therefore it has been the subject of much scientific research involving mycobacterial infection [2–7] .

The complex interactions between bacteria and the immune cell involve various structures and processes that control, activate and inhibit proteins and signalling pathways in a dynamical system that determines the outcome of an infection [8]. A systematic approach to modelling these interactions should help to comprehend the events that occur between the host and pathogen [9]. Different methods have been used to model the mycobacterial infection process: Gammack *et al.* [10] provided a mathematical model based on Ordinary Differential Equation (ODE) to investigate the early and initial immune response to Mtb. Such work has inspired Segovia-Juarez *et al.* [11] to implement the ODEs that regulate the interaction between host and path-

M. Heiner (Ed.): BioPPN 2014, a satellite event of PETRI NETS 2014,
CEUR Workshop Proceedings Vol. 1159, 2014.

ogen using an agent-based approach, and Warrender *et al.* [9] use the CyCells simulator tool to simulate the interactions in Early *mycobacterium* infection.

Mathematical models, like those based on differential equations, are difficult to obtain and analyse when the number of interdependent variables grows and when the relationship depends on qualitative events. The computational models used for this problem offer an additional avenue for exploring the infection dynamics through the visualization of a specific behaviour simulation. However in both cases the interactions between bacteria and the immune cells and their structures are not intuitively described. The interactions are embedded in programing code and/or described in rules which are not straightforward to interact and comprehend their relationship.

A graphical representation of the interactions and influences among the various molecular and cellular components that involve the bacteria and host immune cells that also captures the dynamics of the system should be very useful. The framework of Petri nets represents a well-established technique in computer science for modelling distributed systems [12] and they have been successfully used to model biological behaviour. Heiner *et al.* [13] propose a methodology of incremental modelling using Petri Nets. They develop and analyse a qualitative model of the apoptotic pathway. In our previous work [14] we have developed a qualitative model of the mycobacterial infection process and the innate immune response. We modelled the cell dynamics level, characterized by the steps that are involved in the *Mycobactrium marinum* infection and granuloma formation in zebrafish.

In this paper, we extend our model and focus on interactions between the bacteria and the host immune cells - specifically the macrophages - in a multi-scale model. We identify and connect the important pathways involved in the host-pathogenic interactions that act over different scales (molecular, intracellular, and intercellular) during the innate immune response. The model captures the quintessential functional processes of the macrophage upon exposure to mycobacteria, their interconnections, subsequent signals and activation of the immune response. It provides a visualization of the signalling pathways that the host immune cell utilizes to terminate the infection as well as the way the pathogen exploits the pathways of the macrophages to enhance its intracellular survival persistence. This Petri net model makes it possible to perform "what-if" situations as part of the experimentation, simulating possible pathway disruptions and the consequences to the infection process. In this paper, we demonstrate the power of the Petri net formalism in modelling signalling and metabolic pathways that are involved in the host-pathogen interaction in a multi-scale model. We apply three different dynamics in the animation mode to mimic the alternatives that might occur once a bacterium is phagocytosed by a macrophage and the persistence of infection. As a next step we plan to consider a qualitative validation of the model so as to confirm consistency and correctness of its biological interpretation.

2 Mycobacteria Interaction With Macrophage

Macrophages play rather contradictory roles in infection and disease as they are likely the first host immune cells to respond to invading mycobacteria, and yet aid in subsequent dissemination of the bacteria [15]. The successful parasitation of macrophages by mycobacteria involves the inhibition of several host-cell processes, which allows the bacteria to survive inside the host cells. The host processes that are inhibited by the pathogenic bacteria include fusion of *Phagosomes* with *Lysosomes*, antigen presentation, apoptosis and the stimulation of bactericidal response [16].

Mycobacterial cells release a mixture of lipids and glycolipids that interfere on the macrophage response towards elimination and enabling bacterial survival [17]. Mannosylated Lipoarabinomannam (ManLAM) is one of the major modulators of phagosome maturation [18]. It prevents fusion of mycobacterial phagosome with the late endosome and lysosome by inhibiting the Calmoduling-Ca^{2+} phosphotidol-inositol-3 kinase [19]. Ca^{2+} also has influence in the apoptotic pathways since it increases the permeability of mitochondrial membranes releasing pro-apoptotic elements to facilitate apoptosis [16]. ManLAM also influences the apoptosis by phosphorylating the apoptotic protein Bad leaving the anti-apoptotic protein Bcl-2 free which inhibits caspase activity and functions as an anti-apoptotic regulator [20].

Macrophages and T cells produce many cytokines that promote or inhibit protective response to the mycobacterial infection. An important family of cytokines are the interleukin-10 (IL-10) that regulates the pro-inflammatory (PICs) and anti-inflammatory (AICs) cytokines. The bacteria can limit macrophage apoptosis by inducing the production of IL-10 which blocks the synthesis of Tumor-Necrosis Factor (TNF), a stimulator of apoptosis in infected macrophage [21, 22]. It is likely that bacteria prevent apoptosis in the early phase of infection to allow them to replicate efficiently. However, they induce or are unable to prevent cell death in the later phase, which might facilitate their systemic dissemination through uptake into immune cells [16].

2.1 Cell-cell Host Pathogen Interaction

The modulation of host signalling mechanism is a dynamic process requiring mycobacterial components that trigger or inhibit the host response such as the fusion of Phagosomes with Lysosomes, antigen presentation, apoptosis and stimulation of bactericidal responses due to the activation of pathways that leads to the bacterial survival. The immune cells can identify the pathogen through Pattern Recognition Receptors (PRRs), which are found on the cell surface, on the endosomes and on cytoplasm. It triggers a cascade of events that leads to proinflammatory and antimicrobial response through the phagosome maturation pathway. Van der Vaart *et al.* reviewed the PRRs that identify invading microbes, as well as the innate immune effector mechanisms that they activate in zebrafish embryos [23]. The maturation of the phasosome forms the late-phagosome which fuses with the lysosome forming the phagolysosome which can digest the pathogen and leads to the bacterial death [24–26]. The mycobacteria are using several strategies to avoid the maturation of the phagosome and the key contributor is mannosylated lipoarabinomannan (ManLAM), a glycolipid of the mycobacteria cell wall. ManLAM is involved in the inhibition the phagosome maturation by inhibition of calcium (Ca^{2+}) concentration rise in macrophage and also the

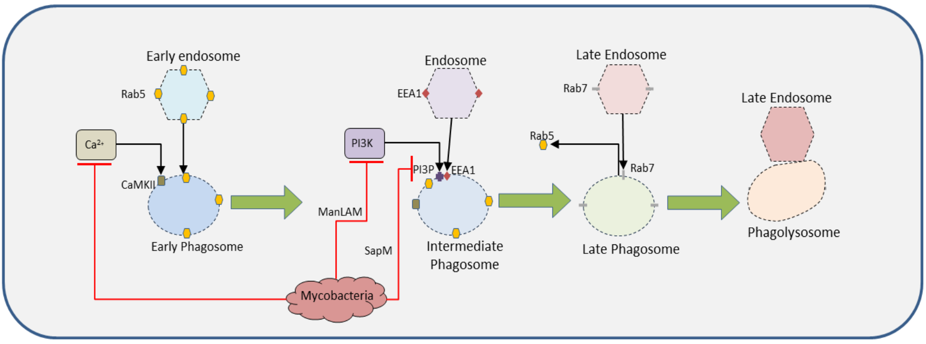


Fig. 1: Schematic overview of the phagosomal maturation pathway blocked by pathogenic mycobacteria according to Koul et al. [16]. **Nascent phagosome acquires Rab5 recruiting PI3K which generate PI3P. Pathogenic mycobacteria block the rise in cellular Ca^{2+}, recruitment of PI3K to the phagosomes and degrading PI3P through Sap-M**

Calmoduling-Ca^{2+} phosphotidol-inositol-3 kinase (PI3K) which is responsible to lead the maturation of the phagosome and drives the fusion with the lysosome [19, 27, 28]. To accomplish complete arrest and prevent the phagosome maturation, a second mycobacterial macromolecule, SapM, is released degrading the existing Phosphatidylinositol 3-phosphate (PI3P), a phospholipid found in the cell membrane involved on the phagosomal maturation [29]. A schematic representation of the phagosomal maturation arresting by the pathogenic mycobacteria is given in **Fig. 1**.

When the immune cell is not able to kill the bacteria through the phagolysosome, the macrophage activates the apoptosis thereby programming its own death and signalling to others defences mechanisms. Once the maturation fails, the apoptotic programme is mainly activated by the extrinsic apoptosis pathway, which is initiated by binding of ligands to death receptors; and the intrinsic pathway, which involves translocation of cytochrome-C from mitochondria to the cytosol. The activation of the caspase cascade and degradation of genomic DNA are characteristics of apoptotic cell death [16]. Mycobacteria alter host apoptotic pathways interfering on the intrinsic death pathway preventing the increasing in cytosolic Ca^{2+} concentration and also inhibit caspase activity and functions by stimulating the phosphorylation of the apoptotic protein Bad [28, 30]. It also limits macrophage apoptosis by inducing the production of cytokines such as interleukin-10 (IL-10) which interferes in one of the apoptosis stimulators of the macrophage in the extrinsic apoptosis pathway, the tumour-necrosis factor-α (TNF- α) [22, 31]. Mycobacteria take advantage of blocking these defence mechanisms of macrophages, phagocytosis and apoptosis, to proliferate inside the cell till a necrosis breakdown and dissemination of infection through the others immune cells that aggregate at that particular infected macrophage to take over the infection. The apoptotic pathway is depicted in **Fig. 2**.

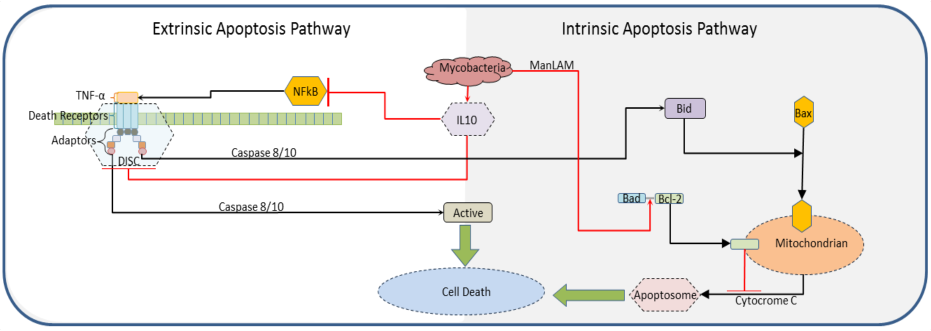


Fig. 2: Apoptotic pathway inhibition by pathogenic mycobacteria according to Koul et al. [16]**. Pathogen mycobacteria interfere in intrinsic apoptotic pathway suppressing Ca^{2+} and releasing Bcl-2. In the extrinsic apoptotic pathway it inhibits binding ligands and DISC formation.**

2.2 Molecular Host Pathogen Interaction

At the molecular level the most important interactions occur in the phagosomal maturation pathway and also in the apoptosis pathway. In both cases, the mycobacteria interfere in different ways to guarantee their survival and proliferation. Ca^{2+} is a key messenger that is released from intracellular storage; an increase in cytosolic Ca^{2+} concentration promotes the phagosomal maturation process by regulating calmodulin and the multifunctional serine/threonine protein kinase CaMKII [28]. CaMKII is important to PI3K activation and recruitment of early endosomal antigen 1 (EEA1) to the phagosomal membrane that is extremely important in the process of phagosomal maturation. PI3K is also essential for the production of the lipid regulator phosphatidylinositol 3-phosphate (PI3P) which form a ligand together with EEA1 leading to an intermediate phagosome which maturates to the late the endosome after EEA1 dissociation and acidic expression due to accumulation of the proton-ATPase [32, 33]. Through releasing ManLAM, the mycobacteria inhibit the rise of the Ca^{2+} concentration in macrophages and also the PI3K activation, preventing the generation of PI3P degrading the existing PI3P by the action of SapM.

Despite the fact that phagosomes fail to fuse with the lysosomes to degrade the bacteria, pathogen-derived material is released in the host cell lysosomes and the cell surface of the infected macrophage which can induce the apoptosis process [34]. Mycobacteria influence the host apoptosis through several mechanisms that interfere in the intrinsic and extrinsic apoptosis pathways. The cytosolic Ca^{2+} facilitates apoptosis by increasing the permeability of mitochondrial membranes that promote the release of pro-apoptotic elements such as cytochrome-C. In the cytosol, cytochrome-C associates with procaspase-9 and apoptosis protease forming a signaling complex called the apoptome which activates the induction of apoptosis [35]. ManLAM interfere in the intrinsic apoptosis pathway not only inhibiting the concentration of Ca^{2+} but also stimulating the phosphorylation of the apoptotic protein Bad that leave BCL-2 free that also prevents the release of cytochrome c.

The extrinsic apoptosis pathway is induced by Toll-like receptors (TLRs) who identify the virulence mycobacterial pathogen and trigger the synthesis of tumor-necrosis factor-α (TNF-α) - a stimulator of apoptosis – trough the TLR signaling pathway. To do so, an important adaptor factor protein, the Myleoid differentiation factor 88 (MYD88) recruits a family of kinases (IRAK) that will form "myddosome" signaling complex that activate nuclear factor kB (NF-kB) to transcript target gene to synthesize TNF-α. The tumor necrosis factor binds with death receptors leading to a cascade of events that will release caspase 8 and 10 and the formation of a death-inducing signal complex (DISC) resulting on the formation of apoptotic vesicles [35, 36]. Pathogen mycobacteria interfere in this process by inducing the production of immunosuppressive cytokine interleukin-10 (IL-10), which inhibit the phosphorylation of NF-kB, therefore the synthesis of TNF-α. It also inhibits the DISC formation and the extrinsic apoptotic pathway failure.

3 Petri Net Model of the Bacterium–Macrophage Interaction

We construct a Petri Net model of the process triggered in the macrophage in response to mycobacterial infection, based on an extensive literature survey and extending our previous model [14]. The model captures the interactions between the immune cell and the pathogen once a bacterium is phagocyte. The model is hierarchical and has three different levels of representations to mimic the signal processing that activates/inhibits the pathways related to the macrophage response to the bacteria. The first level models the overall actions from the system started after the phagocytosis and it represents the cell-cell interaction between the macrophage and the bacteria. The second level representing the intracellular interaction models two important signalling pathways: the Phagosome Maturation which is responsible for the degradation of the infection through antimicrobial components; and the Apoptotic Pathway which is the macrophage mechanism responsible to resolve the infection in response to virulence factors. It represents an alternative way to the phagolysosome. The third level represents the molecule-molecule interactions that occur on the Phagosome Maturation and Apoptotic pathways.

To model the host-pathogen interaction we use an Extended Petri Net implemented in the Snoopy tool [37] with a maximal concurrency semantics. All formal definitions can be found in [38]. The pathways described in section 2 represent a complex process involving various host-bacterial factors in a heavy cross-talk interaction. To get a consistent view of the entire interaction process, we express the most important reactions simplifying the pathways at different levels of abstraction. We define each biochemical compound or receptor as a place. The relations between biochemical substances are represented basically by transitions with corresponding arcs modeling biochemical reactions, inhibitions/degradations (using inhibitor arcs) or signaling/catalytic atomic events (using read arcs). To hierarchically connect the subnets we use coarse transitions and coarse places structuring all the levels as a tree as shown in **Fig. 3**. The top level (the root) models an overall view of the system starting by interactions that occur in the cellular wall and its consequences. It is connected to the subnets (mid-level) through coarse transitions which link to the molecular level modeled in coarse places (the leaves of the tree).

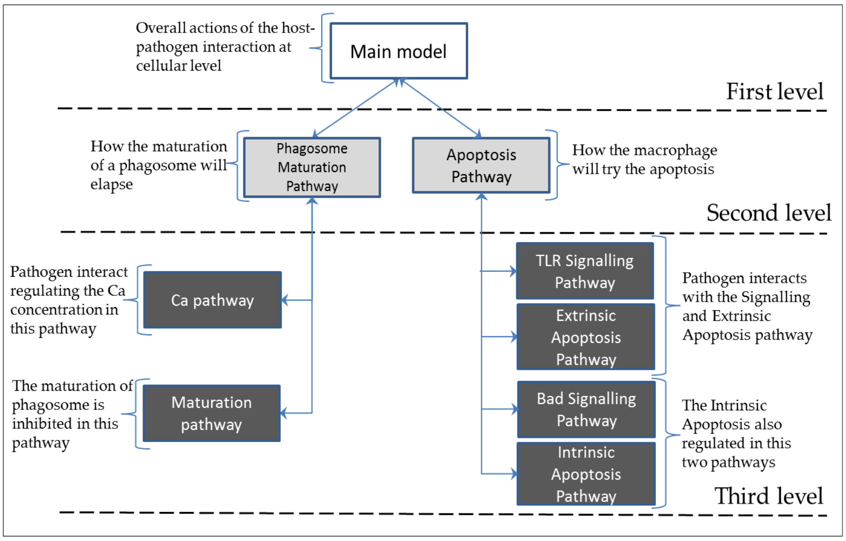

Overall actions of the host-pathogen interaction at cellular level
Main model
First level
How the maturation of a phagosome will elapse
Phagosome Maturation Pathway
Apoptosis Pathway
How the macrophage will try the apoptosis
Second level
Pathogen interact regulating the Ca concentration in this pathway
Ca pathway
The maturation of phagosome is inhibited in this pathway
Maturation pathway
TLR Signalling Pathway
Extrinsic Apoptosis Pathway
Pathogen interacts with the Signalling and Extrinsic Apoptosis pathway
Bad Signalling Pathway
Intrinsic Apoptosis Pathway
The Intrinsic Apoptosis also regulated in this two pathways
Third level

Fig. 3: Hierarchical structure of the net. The three levels are implemented in independent interconnected subnets

3.1 Model Definition

We start the modeling with the interaction between the bacteria inside the macrophage once it is in the host. The first level of our Petri net model is given in **Fig. 4**. The input place ***Infected_macrophage*** represents this situation. The sequence of interaction events happens once there is a bacterium infecting the macrophage (a token is present at the input place) detected by three reading arcs to trigger the interactions. The macrophage uses the PRRs to detect the presence of the pathogen and starts the phagosome maturation process, the bacteria starts its protein secretion system and counter attack by releasing SapM to degrade existing PI3P in the cytosol and ManLAM to interfere in the maturation of the phagosome which is modeled in a lower level by the coarse transition ***Phagosome_Maturation_Pathway***; and in the apoptosis process which is modeled in a lower level in the coarse transition ***Apoptosis_Pathway***. The presence of ManLAM triggers the macrophage production of the cytokine IL10 and also interferes in both pathways. ***Phagosome_Maturation_Pathway*** interacts with ***Apoptosis_Pathway*** releasing calcium and bactericidal material that was not degraded by the maturation.

In our model there are three different scenarios: The phagosome maturation occurs in the ***Phagosome_Maturation_Pathway*** leading to a late phagosome that will fuse with lysosome digesting the bacteria and turning the macrophage healthy. The second scenario can occur if the maturation fails but the apoptosis process in the ***Apopto-***

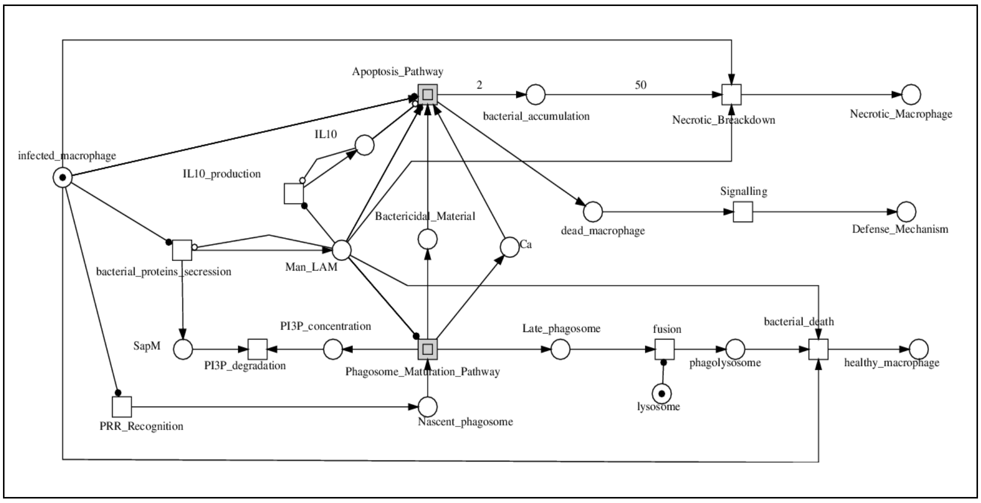


Fig. 4: Petri Net of host-pathogen interaction at the top (root) level. The coarse transitions: *Phagosome_Maturation_Pathway* and the *Apoptosis_Pathway* contain the second level of the model and are represented here by a double square.

sis_Pathway, leading to a dead macrophage, which will signal for another defense mechanism. The third scenario occurs when both pathways are failing at the molecular level, in that case the bacteria proliferate and accumulate in the macrophage till a necrosis breakdown, releasing all the pathogenic material to the surrounding cells. To represent the proliferation and accumulation of bacteria, we use weighted arcs that double the amount of bacteria (accumulated in the place: ***Bacterial_accumulation***). The breakdown of the macrophage occurs when it reaches a threshold of 50 bacteria (a weighted arc fires the transition ***Necrotic_breackdown***). Here we should note that the weighted arcs (with weights 2 and 50) are examples to express the idea of bacterial proliferation.

Following the hierarchical tree, we have at the second level: ***Phagosome_Maturation_Pathway*** and ***Apoptosis_Pathway***, two subnets which basically connect the cellular interaction (top level) with the molecular interactions at the biochemical pathways implemented in the coarse places (the leaves of the tree). **Fig. 5** depicts these subnets. At this level we have the signaling started in the cell wall (top level) that will trigger the production/interaction between molecules. For example the production/releasing of calcium is triggered by the PRRs and this process occur at ***Ca_pathway***; the PIP3 concentration and bactericidal material that are not degraded at the ***maturation_pathway*** and interact with the top level. We also have the interaction between the cytokine IL10 from the top level with the pro-inflammatory cytokines that will interfere in the TNF-α in the ***Extrinsic_Apopthosis_Pathway*** and ManLAM interfering in the BCL2 activation, which will act in the ***Intrinsic_Apopthosis_Pathway***.

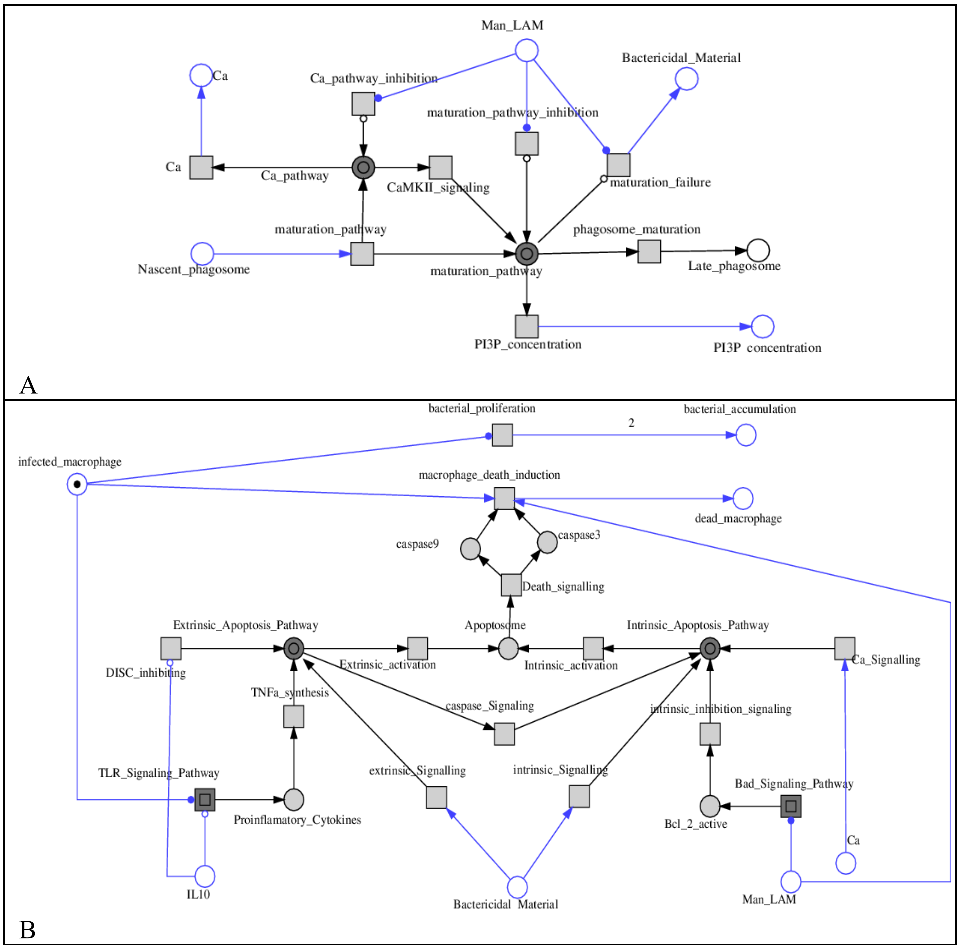


Fig. 5: Second level. A) The *Maturation_Pathway* Subnet which connects the cellular level with the molecular interactions in the production of calcium (*Ca_pathway*) and phagosomal maturation (*Maturation_pathway*) implemented in another subnet level (coarse places). B) *Apoptosis_Pathway* Subnet which connects the cellular level with the molecular interactions in the pro inflammatory cytokines, Bcl2 activation and their influences in the extrinsic and intrinsic apoptosis pathway.

At the lower level of the hierarchical model, we have six related but independent subnets. They represent the important molecular pathways responsible for: the production of calcium/CMKII in ***ca_pathway*** and phagosome maturation and PI3P concentration in ***maturation_pathway,*** both implemented in coarse places; activation of pro-inflammatory cytokines in ***TLR_Signaling_Pathway*** and Bcl2 activation in ***Bad_Signalling_Pathway***, both implemented in coarse transitions; and also the interactions that occur in the intrinsic and extrinsic apoptosis pathways implemented in the coarse places ***Extrinsic_Apopthosis_Pathway*** and ***Intrinsic_Apopthosis_Pathway*** respecivly. **Fig. 6** depicts all subnets in the leaves of the proposed hierarchical model.

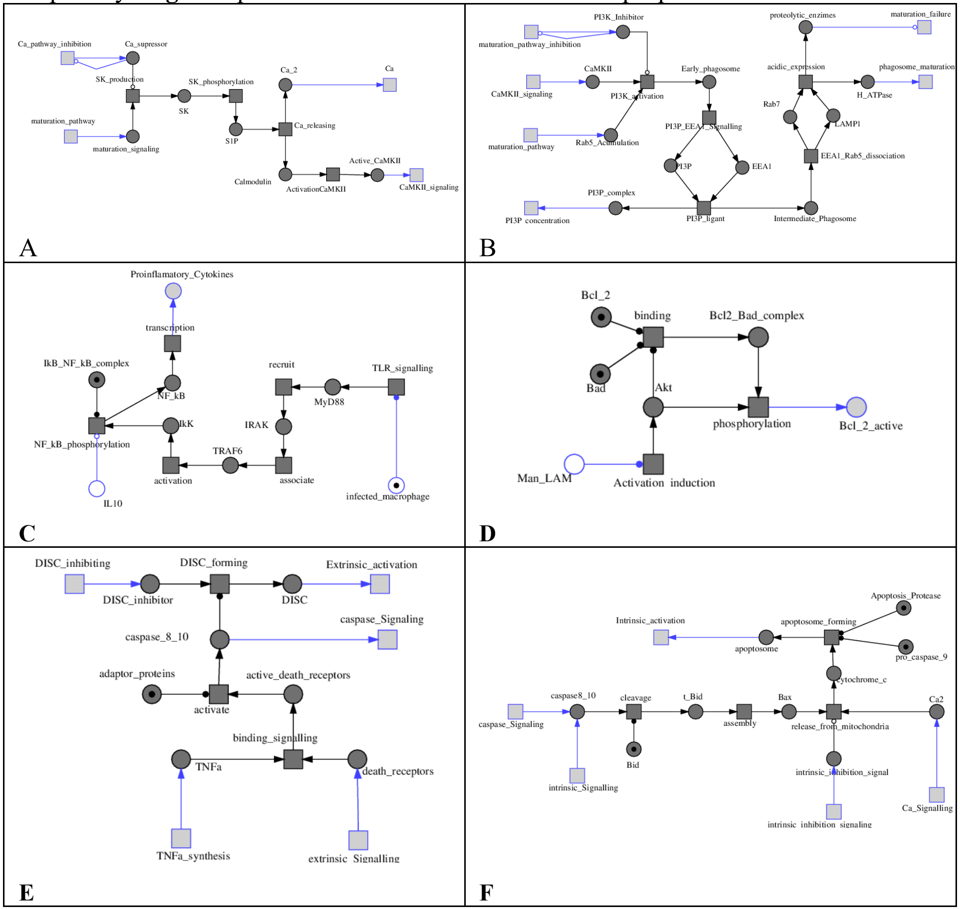


Fig. 6: Subnets that implement the molecular pathway interaction. A) Ca^{2+} and CaMKII production/inhibition model. B) Phagosome maturation pathway (activation and inhibition) and PI3P concentration model. C) TLR signalling and pro-inflammatory cytokines activation/inhibition model. D) Bad/BCL2 complex phosphorylation model. E) Extrinsic apoptosis pathway model. F) Intrinsic apoptosis pathway model

3.2 Animation and Validation

Animation and validation are important tools to provide a consistent model of biological behavior. They allow experimenting with different situations and possibilities of the model as well as checking for integrity and correspondence to the real world. Simulation and analysis for qualitative and quantitative behavior prediction are other steps necessary to certify a useful model. For the model presented here we also performed some animation and validation. For this we employed the Animation mode available in the Snoopy software. This feature allows animating the token flow of the net through all the subnets, visualizing the causality of the model and its behavior. Three different animations for the scenarios, previously discussed, were performed to experience the events that can occur in the model. For inspection and perusal, the animations can be found at http://bio-imaging.liacs.nl/galleries/epn-infection/.

To validate the model, it is necessary to define validation criteria for a consistency check. To do so, we have to consider that our model is based on a heuristic procedure of collecting information from the literature, perhaps with different interpretations, modeled from the process perspective (top level) down to the molecule perspective (leaves). We built a large model composed of sub-models and to provide a complete analysis, it is necessary to verify each component individually and the system as a whole, which increases the complexity of the validation even with computational support. Basic qualitative behavior properties can be checked using the Charlie analyzer tool [39]. Heiner *et al.* have used as example, p-invariants and t-invariants to analyze case studies in biochemical pathways in [40]. We started to analyze the structural and behavior properties of our model based on results from the Charlie tool which should then be biological meaningful. As a first result, we found that our model is not structurally bounded and not reversible. This implies that indeed the net allows for the proliferation of the bacteria and the infection process is not reversible.

4 Conclusion

In mycobacterial infection, the dynamics of the interactions between the host and bacteria forms a complex system involving numerous activations, inhibitory and control structures that determine the outcome of the infection. A systems approach is essential to comprehend the significance of the multiple events that occur simultaneously among the various molecular and cellular components of the host and pathogen.

Here, we seek to model the interaction of the macrophage upon exposure to pathogen mycobacteria, capturing important functional process and their interconnections including signaling and activation/inhibition of the immune responses on different levels of abstraction. The Petri net formalism has proved to be a useful modeling approach to describe and interconnect different abstract levels into a large and extensive model [13, 41] In our previous work [14] we have developed a Colored Petri net model to explore the early mycobacterial infection and the immune response, modeling the steps that regulates the infection process. In this paper we focus on the lower-scale processes occurring in the cell and descend to molecular interactions relevant to the infection process. Therefore we use an Extended Petri net for the different pathways in subnets, interconnecting them in a hierarchical structured model. The model provides a visualization of the processes occurring at multiple scales using levels that can be operated independently. It also describes the interconnections and signals that influence the host pathogen interaction.

This results in an Extended Petri net model implementation in the Snoopy tool [37]. The model expresses, at different levels of abstraction, the details that are involved in the macrophage-mycobacterium interaction. Information about the proteins released by the bacteria, their interference in the immune response and the pathways involved in this process are observed in our model. It is possible to visualize the dynamics of the molecular and cellular interaction as well as analyze different scenarios performing "what-if" simulation as part of the experimentation in the animation mode. The model represents the information about host-pathogen interaction available in the literature but the scalability of our model allows extension to a more complete system.

As part of the modelling process, we started to use the Charlie analyzer [39] to check properties of the model and its consistency. As a next step, an extensive analysis of more structural and behavior properties is necessary to validate the model. We also intend to extend to a quantitative model where, with support of experimental data rather than the examples we used until now, we can use analysis techniques for a prediction of qualitative as well as quantitative behavior. This can contribute, for example, in the prediction of results from new experiments and generation of further hypotheses about the innate immune system response to mycobacterial infection. Another challenge is to combine the models implemented in different classes of Petri nets in one system. One solution is to adapt each model in a Hybrid Petri Net, or abstract the models in a Nets-within-Nets approach where the communication of the tokens occurs via predefined interfaces which are dynamically bounded [42].

In summary we have presented in this paper a model that explores the interaction between mycobacterial pathogen and macrophage, modeling the dynamics in three different level of abstraction while interconnecting them in a hierarchical structure. We have checked the structural behavior of our model through an analysis tool. The interplay of hierarchical levels and qualitative/quantitative information has the potential to develop a powerful tool for the research in tuberculosis disease.

5 Acknowledgement

We are grateful to the anonymous reviewers for their constructive criticism which helped us to improve the presentation of this work in progress. This work was partially supported by Erasmus mundus and CNPq.

6 References

1. WHO Report, http://www.who.int/tb/publications/global_report/en/ (2012).

2. Davis, J.M., Clay, H., Lewis, J.L., Ghori, N., Herbomel, P., Ramakrishnan, L.: Real-time visualization of mycobacterium-macrophage interactions leading to initiation of granuloma formation in zebrafish embryos. Immunity. 17, 693–702 (2002).

3. Clay, H., Davis, J.M., Beery, D., Huttenlocher, A., Lyons, S.E., Ramakrishnan, L.: Dichotomous role of the macrophage in early Mycobacterium marinum infection of the zebrafish. Cell Host Microbe. 2, 29–39 (2007).

4. Lesley, R., Ramakrishnan, L.: Insights into early mycobacterial pathogenesis from the zebrafish. Curr. Opin. Microbiol. 11, 277–83 (2008).

5. Stein, C.: Innate immune genes in the zebrafish, Danio rerio. PhD Thesis. Mathematisch-Naturwissenschaftliche Fakultät, Universität zu Köln. Germany. http://kups.ub.uni-koeln.de/volltexte/2011/3308/, (2011).

6. Lalvani, A., Behr, M. a, Sridhar, S.: Innate immunity to TB: a druggable balancing act. Cell. 148, 389–91 (2012).

7. Stanley, S. a, Cox, J.S.: Host-pathogen interactions during Mycobacterium tuberculosis infections. Curr. Top. Microbiol. Immunol. 374, 211–41 (2013).

8. Casadevall, A., Pirofski, L.: Host-pathogen interactions: the attributes of virulence. Journal of Infection Disease. 184, 337–44 (2001).

9. Warrender, C., Forrest, S., Koster, F.: Modeling intercellular interactions in early Mycobacterium infection. Bull. Math. Biol. 68, 2233–61 (2006).

10. Gammack, D., Doering, C.R., Kirschner, D.E.: Macrophage response to Mycobacterium tuberculosis infection. J. Math. Biol. 48, 218–42 (2004).

11. Segovia-Juarez, J.L., Ganguli, S., Kirschner, D.: Identifying control mechanisms of granuloma formation during M. tuberculosis infection using an agent-based model. J. Theor. Biol. 231, 357–76 (2004).

12. Desel, J., Reisig, W., Rozenberg, G.: Lectures on Concurrency and Petri Nets. Springer Berlin Heidelberg, Berlin, Heidelberg (2004).

13. Heiner, M., Koch, I., Will, J.: Model validation of biological pathways using Petri nets--demonstrated for apoptosis. Biosystems. 75, 15–28 (2004).

14. Carvalho, R. V., Kleijn, J., Meijer, A.H., Verbeek, F.J.: Modeling innate immune response to early mycobacterium infection. Comput. Math. Methods Med. 790482 (2012).

15. Flynn, J.L., Chan, J.: What's good for the host is good for the bug. Trends Microbiol. 13, 98–102 (2005).

16. Koul, A., Herget, T., Klebl, B., Ullrich, A.: Interplay between mycobacteria and host signalling pathways. Nat. Rev. Microbiol. 2, 189–202 (2004).

17. Rosenberger, C.M., Finlay, B.B.: Phagocyte sabotage: disruption of macrophage signalling by bacterial pathogens. Nat. Rev. Mol. Cell Biol. 4, 385–96 (2003).

18. Stoop, E.J.M., Mishra, A.K., Driessen, N.N., van Stempvoort, G., Bouchier, P., Verboom, T., van Leeuwen, L.M., Sparrius, M., Raadsen, S.A., van Zon, M., van der Wel, N.N., Besra, G.S., Geurtsen, J., Bitter, W., Appelmelk, B.J., van der Sar, A.M.: Mannan core branching of lipo(arabino)mannan is required for mycobacterial virulence in the context of innate immunity. Cell. Microbiol. (2013).

19. Fratti, R.A., Chua, J., Vergne, I., Deretic, V.: Mycobacterium tuberculosis glycosylated phosphatidylinositol causes phagosome maturation arrest. Proc. Natl. Acad. Sci. U. S. A. 100, 5437–42 (2003).

20. Abarca-Rojano, E., Rosas-Medina, P., Zamudio-Cortéz, P., Mondragón-Flores, R., Sánchez-García, F.J.: Mycobacterium tuberculosis virulence correlates with mitochondrial cytochrome c release in infected macrophages. Scand. J. Immunol. 58, 419–27 (2003).

21. Schluger, N.W., Rom, W.N.: The host immune response to tuberculosis. Am. J. Respir. Crit. Care Med. 157, 679–91 (1998).

22. Moore, K.W., de Waal Malefyt, R., Coffman, R.L., O'Garra, A.: Interleukin-10 and the interleukin-10 receptor. Annu. Rev. Immunol. 19, 683–765 (2001).

23. Van der Vaart, M., Spaink, H.P., Meijer, A.H.: Pathogen Recognition and Activation of the Innate Immune Response in Zebrafish. Advances in hematology. 159807. (2012).

24. Rohde, K., Yates, R.M., Purdy, G.E., Russell, D.G.: Mycobacterium tuberculosis and the environment within the phagosome. Immunol. Rev. 219, 37–54 (2007).

25. Bouley, D.M., Ghori, N., Mercer, K.L., Falkow, S., Ramakrishnan, L.: Dynamic nature of host-pathogen interactions in Mycobacterium marinum granulomas. Infect. Immun. 69, 7820–31 (2001).

26. Booth, J.W., Trimble, W.S., Grinstein, S.: Membrane dynamics in phagocytosis. Semin. Immunol. 13, 357–64 (2001).

27. Vergne, I., Chua, J., Deretic, V.: Tuberculosis toxin blocking phagosome maturation inhibits a novel Ca2+/calmodulin-PI3K hVPS34 cascade. J. Exp. Med. 198, 653–9 (2003).

28. Malik, Z.A., Denning, G.M., Kusner, D.J.: Inhibition of Ca2+ Signaling by Mycobacterium tuberculosisIs Associated with Reduced Phagosome-Lysosome Fusion and Increased Survival within Human Macrophages. J. Exp. Med. 191, 287–302 (2000).

29. Puri, R.V., Reddy, P.V., Tyagi, A.K.: Secreted Acid Phosphatase (SapM) of Mycobacterium tuberculosis Is Indispensable for Arresting Phagosomal Maturation and Growth of the Pathogen in Guinea Pig Tissues. PLoS One. 8, e70514 (2013).

30. Rojas, M., García, L.F., Nigou, J., Puzo, G., Olivier, M.: Mannosylated lipoarabinomannan antagonizes Mycobacterium tuberculosis-induced macrophage apoptosis by altering Ca+2-dependent cell signaling. J. Infect. Dis. 182, 240–51 (2000).

31. Clay, H., Volkman, H.E., Ramakrishnan, L.: Tumor necrosis factor signaling mediates resistance to mycobacteria by inhibiting bacterial growth and macrophage death. Immunity. 29, 283–94 (2008).

32. Fratti, R.A., Backer, J.M., Gruenberg, J., Corvera, S., Deretic, V.: Role of phosphatidylinositol 3-kinase and Rab5 effectors in phagosomal biogenesis and mycobacterial phagosome maturation arrest. J. Cell Biol. 154, 631–44 (2001).

33. Sundaramurthy, V., Pieters, J.: Interactions of pathogenic mycobacteria with host macrophages. Microbes Infect. 9, 1671–9 (2007).

34. Russell, D.G., Mwandumba, H.C., Rhoades, E.E.: Mycobacterium and the coat of many lipids. J. Cell Biol. 158, 421–6 (2002).

35. Lee, J., Hartman, M., Kornfeld, H.: Macrophage apoptosis in tuberculosis. Yonsei Med. J. 50, 1–11 (2009).

36. KEGG PATHWAY: Tuberculosis - Mycobacterium marinum M, http://www.genome.jp/kegg-bin/show_pathway?mmi05152 (2013).

37. Heiner, M., Herajy, M., Liu, F., Rohr, C., Schwarick, M.: Snoopy – A Unifying Petri Net Tool. Petri Nets 2012. pp. 398–407. Springer, LNCS, Hamburg (2012).

38. Blätke, M., Heiner, M., Marwan, W.: Tutorial Petri Nets in Systems Biology. Technical Report, Otto-von-Guericke University, Magdeburg. (2011).

39. Wegener, J., Schwarick, M., Heiner, M.: A plugin system for Charlie. Proc. CS&P. 531–554 (2011).

40. Heiner, M., Koch, I.: Petri net based model validation in systems biology. Applications and Theory of Petri Nets 2004. pp. 216–237. Springer (2004).

41. Marwan, W., Rohr, C., Heiner, M.: Petri nets in Snoopy: A unifying framework for the graphical display, computational modelling, and simulation of bacterial regulatory networks. Bact. Mol. Networks. 1–31 (2012).

42. Heitmann, F., Köhler-Bußmeier, M.: P-and T-Systems in the Nets-within-Nets-Formalism. Appl. Theory Petri Nets. 368–387 (2012).

Systemic approach for toxicity analysis

Cinzia Di Giusto, Hanna Klaudel, and Franck Delaplace

Université d'Evry - Val d'Essonne, Laboratoire IBISC, Evry, France

Abstract. A high-level Petri net framework is introduced for the toxic risk assessment in biological and bio-synthetic systems. Unlike empirical techniques mostly used in toxicology or toxicogenomics, we propose a systemic approach consisting of a series of behavioral rules (reactions) that depend on abstract discrete "expression" levels of involved agents (species). We introduce a finite state high-level Petri net model allowing exhaustive verification (model-checking) of properties related to equilibrium alteration or appearing of hazardous behaviors. The approach is applied to the study of the impact of the aspartame assimilation into the blood glucose regulation process.

1 Introduction

Toxicology [23] studies the adverse effects of the exposures to chemicals at various levels of living entities: organism, tissue, cell or intracellular molecular systems. During the last decade, the accumulation of genomic and post-genomic data together with the introduction of new technologies for gene analysis has opened the way to *toxicogenomics*. Toxicogenomics combines toxicology with "Omics" technologies[1] to study the *mode-of-action* of toxicants or environmental stressors on biological systems. The mode-of-action is understood as the sequence of events from the absorption of chemicals to a toxic outcome. Toxicogenomics potentially improves clinical diagnosis capabilities and facilitates the identification of potential toxicity in drug discovery [10] or in the design of bio-synthetic entities [21].

The main approach used in toxicogenomics employs empirical analysis like in the identification of molecular biomarkers, i.e., indicators of disease or toxicity in the form of specific gene expression patterns [7]. Clearly, biomarkers remain observational indicators linking genes related measures to toxic states. In this proposal, we complement these empirical methods with a computational technique that aims at discovering the molecular mechanisms of toxicity. This way, instead of studying the phenomenology of the toxic impacts, we focus on the processes triggering adverse effects on organisms. Usually, the toxicity process is defined as a sequence of physiological events that causes the abnormal behavior of a living organism with respect to its healthy state. Healthy physiological states generally correspond to homeostasis, namely a process that maintains a dynamic stability of internal conditions against changes in the external environment. Hence, we

[1] "Omics" technologies are methodologies such as genomics, transcriptomics, proteomics and metabolomics.

M. Heiner (Ed.): BioPPN 2014, a satellite event of PETRI NETS 2014,
CEUR Workshop Proceedings Vol. 1159, 2014.

will consider toxicity outcomes as deregulation of homeostasis processes, namely deviation of some intrinsic referential equilibrium of the system.

Biological processes are usually given in terms of pathways which are causal chains of the responses to stimuli, this way the deregulation of homeostasis appears as the activation or inhibition of unexpected but existing pathways. Moreover, in the context of toxicogenomics it is crucial to take into account at least two other parameters: the exposure time and the thresholds dosage delimiting the ranges of safe and hazardous effects.

In this paper, we depict and analyze the mechanistic process of toxicology using high-level Petri nets. Our work is inspired by the definition of reaction systems as given in [1]. A reaction system is a set of *reactions*, each of them defined as a triple (R, I, P) where R is the set of reactants, I the set of inhibitors and P the set of products, and R, I and P are taken from a common set of species $\mathcal{S}$. Reaction systems are based on three foundational principles:

1. a reaction can take place only if all the reactants involved are available but none of the inhibitors is;
2. if a species is available then a sufficient amount of it is present to trigger a reaction;
3. species are not persistent: they become unavailable if they are not sustained by a reaction.

From this model we retain the idea of reactions but we significantly change the semantics. The first change concerns principle 2: species are available at a given discrete abstract level. This is mainly related to the need of expressing toxicants doses. The corresponding discretization is built observing thresholds levels in dose-response curves. The second and more fundamental change regards the introduction of discrete time constraints. Time plays a role in the evolution of species, more precisely, species are associated to a decay time δ, meaning that their level diminishes with time. This accounts for the presence of a non-specified environment that consumes and degrades species, thus allowing to abstract away from reactions that may be neglected in the specified context. Each reaction (R, I, P) is extended with levels for all its reactants and inhibitors. Reactions can take place only if each reactant is present at least at a given level and each involved inhibitor is at a level strictly inferior to the given one. As a result, the level of products of the reaction can be increased or decreased.

Summing up, systems are build out of a series of behavioral reactions among involved agents or species. We model such systems into high-level Petri nets and apply it to toxicogenomics problems, namely deregulation of homeostatic processes. Toxicity questions are expressed using a suitable temporal logic like CTL [9]. By observing that our modeling has a finite state space, it is therefore natural to address the satisfiability of these formulae using classic verification techniques such as model checking.

We apply the modeling and verification process on the example of blood glucose regulation in human body showing the maintenance of the homeostasis. In particular, we highlight how the interplay between the assimilation of aspartame and glucose regulation causes the appearance of unwanted behaviors.

Organization of the paper. The paper is organized as follows: Section 2 recalls basic definitions and notations on high-level Petri nets. Next, Section 3 describes our running example of blood glucose regulation. Section 4 introduces the principles behind reaction networks and presents their high-level Petri net modeling. Then Section 5 shows how to check toxicology properties and finally, Section 7 concludes with some considerations on future work.

2 Preliminaries

We recall here the general notations together with some elements of the semantics of high-level Petri nets [15].

Definition 1. *A* high-level Petri net N *is a tuple* (Q, T, F, L, M_0) *where:*

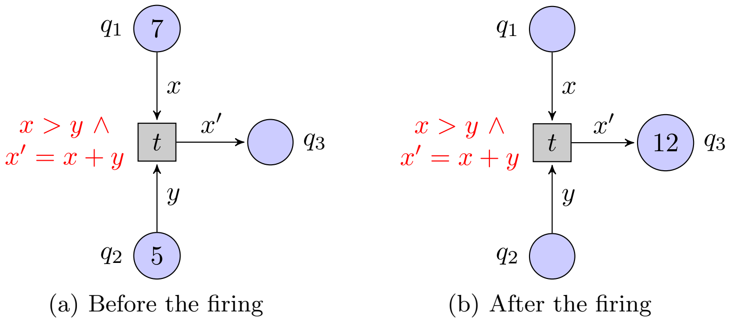
- Q *is the set of places,*
- T *is the set of transitions and* $Q \cap T = \emptyset$*;*
- $F \subseteq (Q \times T) \cup (T \times Q)$ *is the set of arcs;*
- L *is the labeling function from places* Q*, transitions* T *and arcs* F *to a set of labels defined as follows:*
  - $\forall q \in Q$*,* $L(q)$ *is the type of* q*, i.e., a (possibly infinite) set or Cartesian product of sets of integer values;*
  - $\forall t \in T$*,* $L(t)$ *is a computable boolean expression with variables and integer values;*
  - *and* $\forall f \in F$*,* $L(f)$ *is a tuple of variables and integer values compatible with the adjacent place.*
- M_0 *is the initial marking which associates to each place* $q \in Q$ *a multiset of tokens in* $L(q)$*.*

Observe that we are considering a subclass of high-level Petri nets where at most one arc per direction for each pair place/transition is allowed and only one token can flow through. The behavior of high-level Petri nets is defined as usual: markings are functions from places in Q to multisets of possibly structured tokens in $L(q)$ and a transition $t \in T$ is *enabled* at marking M, if there exists an evaluation σ of all variables in the labeling of t such that the guard $L(t)$ evaluates to true ($L_\sigma(t) = \mathsf{true}$) and there are enough tokens in all input places q to satisfy the corresponding input arcs, i.e., $L_\sigma((q,t)) \in M(q)$. Then, the firing of t produces the marking M':

$$\forall q \in Q, M'(q) = M(q) - L_\sigma((q,t)) + L_\sigma((t,q)).$$

with $L_\sigma(f) = 0$ if $f \notin F$, $-$ and $+$ are multiset operators for removal and adding of one element, respectively. We denote it by $M[t{:}\sigma\rangle M'$.

By convention, primed version of variables (e.g. x') are used to annotate output arcs of transitions, their evaluation is possibly computed using unprimed variables (e.g. x and y) appearing on input arcs. With an abuse of notation, singleton markings are denoted without brackets, the same is used in arc annotations. An example of firing is shown in Figure 1. We say that a marking M is *reachable* from the initial marking M_0 if there exists a firing sequence $(t_1, \sigma_1), \ldots, (t_n, \sigma_n)$ such that $M_0[t_1{:}\sigma_1\rangle M_1 \ldots M_{n-1}[t_n{:}\sigma_n\rangle M$.



(a) Before the firing (b) After the firing

Fig. 1. Example of firing with $\sigma = \{x = 7, y = 5, x' = 12\}$.

3 Blood glucose regulation

Here we introduce our running example: *glucose regulation* in human body (Figure 2). In the following, we are always referring to the process under normal circumstances in a healthy body.

Glucose regulation is a homeostatic process: i.e., the rates of glucose in blood (*glycemia*) must remain stable at what we call the equilibrium state. Glycemia is regulated by two hormones: *insulin* and *glucagon*. When glycemia rises (for instance as a result of the digestion of a meal), insulin promotes the storing of glucose in muscles through the glycogenesis process, thus decreasing the blood glucose levels. Conversely, when glycemia is critically low, glucagon stimulates the process of glycogenolysis that increases the blood glucose level by transforming glycogen back into glucose.

We will focus on the assimilation of sweeteners: i.e., sugars or artificial sweeteners such as aspartame. Whenever we eat something sweet either natural or artificial, the sweet sensation sends a signal to the brain (through *neurotransmitters*) that in turns stimulates the production of insulin by pancreas. In the case of sugar, the digestion transforms food into nutrients (i.e., glucose) that

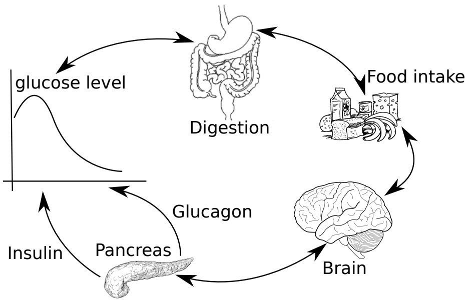


Fig. 2. Glucose metabolism

are absorbed by blood. This way, sugar through digestion increases glucose in blood giving the sensation of satiety. In case the income of glucose produces hyperglycemia, the levels of glucose are promptly equilibrated by the intervention of insulin. Unlike sugar, artificial sweeteners are not assimilated by the body, hence they do not increase the glucose levels in blood. Nevertheless the insulin produced under the stimuli originated by the sweet sensation, although weak, can still cause the rate of glucose to drop engendering hypoglycemia. In response to that, the brain induces the stimulus of *hunger*. As a matter of fact this appears as an unwanted/toxic behavior, indeed the assimilation of food (even if it contains aspartame) should calm hunger and induce satiety not the opposite.

This schema suggests that we should consider four levels for glycemia: low, hunger, equilibrium and high. Likewise for insulin we assume three levels: inactive, low and high. All other actors involved in glucose regulation, have only two levels (inactive or active). In the following sections, we will see how to model the glucose metabolism and how to verify the unexpected behaviors of artificial sweeteners.

4 Petri net modeling

A reaction network is composed of a set of species $\mathcal{S}$ governed by a set of reactions $\mathcal{R}$. We begin by giving some intuitions on their dynamics.

Species in $\mathcal{S}$ represent the actors of the modeled system. In the example introduced above, we have concrete species such as aspartame and also more abstract ones representing ratios or concepts like glycemia. Species may have several *expression* levels. Levels are determined by the observable behavior of species, i.e., they refer to a change in the capability of action of species. In toxicology, they may represent dosages. We assume, for each species s, an arbitrary but finite number $\mathcal{L}_s$ of levels, and each s is initialized at a certain level η_s. For certain species, we assume the presence of a non specified environment that acts on them by decreasing gradually their expression levels. This special activity is called *decay* and is modeled by various durations associated to expression levels. Decay may be unbounded indicating that the level of the species can only change by result of a reaction. It is formalized by a function that associates to each level either ω (unbounded) or its finite duration:

$$\delta_s : [0..\mathcal{L}_s - 1] \to \mathbb{N}^+ \cup \{\omega\}.$$

For all species $s \in \mathcal{S}$ we require that $\delta_s(0) = \omega$ meaning that the duration of the basal level must be unbounded.

Example 1 (Glucose metabolism – species). Take the example from Section 3. The set of involved species is

$$\mathcal{S} = \{Sugar, Aspartame, Glycemia, Glucagon, Insulin\}$$

and their expression levels and corresponding decays are:

levels	durations
$\mathcal{L}_{sugar} = \{0,1\}$	$\delta_{sugar}(1) = 2$
$\mathcal{L}_{aspartame} = \{0,1\}$	$\delta_{aspartame}(1) = 2$
$\mathcal{L}_{glycemia} = \{0,1,2,3\}$	$\delta_{glycemia}(1) = 8$ $\delta_{glycemia}(2) = 8$ $\delta_{glycemia}(3) = 8$
$\mathcal{L}_{glucagon} = \{0,1\}$	$\delta_{glucagon}(1) = 3$
$\mathcal{L}_{insulin} = \{0,1,2\}$	$\delta_{insulin}(1) = 3$ $\delta_{insulin}(2) = 3$

The levels of glycemia are: 0 corresponding to low, 1 to hunger, 2 to equilibrium and 3 to high. Likewise for insulin we have 0 that corresponds to inactive, 1 to low and 2 to high. All levels for the other species are 0 for inactive and 1 for active. ◇

The evolution of species $s \in \mathcal{S}$ is governed by a set of reactions $\mathcal{R}$, each being of the form:

$$\rho ::= \langle R_\rho, I_\rho, P_\rho \rangle \tag{1}$$

where R_ρ (reactants), I_ρ (inhibitors) are sets of pairs (s, η_s) and P_ρ (products) is a non empty set of pairs (s, z), where $\eta_s \in [0..\mathcal{L}_s - 1]$ and $z \in \mathbb{Z}$. Species can appear at most once in each set R_ρ, I_ρ and P_ρ. They can be present in both R_ρ and I_ρ but they must occur with different levels[2]. We write $s \in R_\rho$ to denote $(s, \cdot) \in R_\rho$ similarly for I_ρ and P_ρ and we omit index ρ if it is clear from the context ($\rho = \langle R, I, P \rangle$).

Example 2 (Glucose metabolism – reactions). The set of reactions $\mathcal{R} = \{\rho_k = (R_k, I_k, P_k) \mid k \in [1..9]\}$ for the glucose metabolism example is:

ρ_k	*Reactants* R_k	*Inhibitors* I_k	*Products* P_k
ρ_1	$(Sugar, 1)$	$\emptyset$	$(Insulin, +1), (Glycemia, +1)$
ρ_2	$(Aspartame, 1)$	$\emptyset$	$(Insulin, +1)$
ρ_3	$\emptyset$	$(Glycemia, 1)$	$(Glucagon, +1)$
ρ_4	$(Glycemia, 3)$	$\emptyset$	$(Insulin, +1)$
ρ_5	$(Insulin, 2)$	$\emptyset$	$(Glycemia, -1)$
ρ_6	$(Insulin, 1)$, $(Glycemia, 3)$	$\emptyset$	$(Glycemia, -1)$
ρ_7	$(Insulin, 1)$	$(Glycemia, 2)$	$(Glycemia, -1)$
ρ_8	$(Glucagon, 1)$	$\emptyset$	$(Glycemia, +1)$

ρ_1 and ρ_2 represent the assimilation of Sugar and Aspartame, respectively: while Aspartame only increases the level of Insulin, Sugar also increases Glycemia. ρ_3 takes care of hypoglycemia, i.e., a Glycemia level equal to 0 (obtained by

[2] Observe that a species can appear in the same reaction as reactant at level η_r, inhibitor at level $\eta_i > \eta_r$ and product.

using $(\textit{Glycemia}, 1)$ as inhibitor) engenders the production of Glucagon. On the contrary, hyperglycemia causes the production of Insulin (ρ_4). The presence of Insulin lowers Glycemia (reactions ρ_5, ρ_6, ρ_7). In particular Insulin level equal to 1 plays a role in the decrease of Glycemia only in case of hyperglycemia ρ_6 or hypoglycemia ρ_7, otherwise the signal is not strong enough and we need Insulin at level 2 to see the effect on Glycemia (ρ_5). Last reaction describes the role of Glucagon which if active increases the level of Glycemia.

◇

The dynamics of reaction networks is formalized using high-level Petri nets. We represent the state of a species s as a pair $\langle l_s, u_s\rangle$, where l_s is an integer value storing the current level from zero to $\mathcal{L}_s - 1$, and u_s is a counter storing the interval of time spent at level l_s. The system is initialized by setting the level of all species: i.e., each species s is set to $\langle \eta_s, 0\rangle$ where η_s is the given initial level.

Reaction networks can evolve in two ways:

Case 1. **Time progression and Decay:** Time progresses discretely of one unit at once. It affects species with finite decay only. More precisely if a species s has unbounded decay at level l $(\delta_s(l) = w)$ then its corresponding tuple (η_s, u_s) remains unchanged. Otherwise, if the species has a finite decay $(\delta_s(l) = d)$, it may stay at level l for d time units. Then, degradation happens as soon as d time units are elapsed and is obtained by decreasing the level to $l - 1$ and by setting u_s to zero.

Case 2. **Reaction:** A reaction ρ may happen if and only if all the reactants are available at least at the required level and all the inhibitors are expressed at a level strictly inferior to the required one. The triggering of a reaction results in the update of the level of all its products. Depending on the reaction, levels will be increased (+n) , maintained (0) or decreased (-n). We assume that each reaction can take place only once per time unit.

We now comment on some specific design choices concerning reactions:

- the set of reactants and inhibitors $R \cup I$ is allowed to be empty. This accounts for modeling an environment that is continuously sustaining the production of a species.
- a species can appear in the same reaction simultaneously as a reactant and an inhibitor. In such a case, we require them to occur with different levels:

$$(\{(s, \eta)\} \cup R, \{(s, \eta')\} \cup I, P)$$

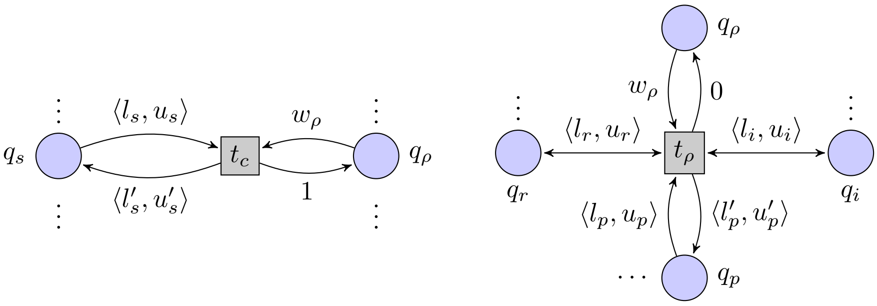
  where $\eta < \eta'$. This means that the reaction can take place only if the level l_s of s belongs to the interval $\eta \leq l_s < \eta'$. In particular, if s has to be present in a reaction exactly at level η, s should appear as a reactant at level η and as inhibitor at level $\eta' = \eta + 1$;
- species can appear only once in the set of products P. This implies that a product cannot be increased and decreased in the same reaction.

It is also worth observing that if a species is continuously sustained by some reactions then it remains available in the system at a certain level for a period that could be longer than the corresponding decay time.

Example 3 (Glucose metabolism – scenario). Take once again the example of glucose metabolism and observe the behavior of *Glycemia* in the following scenario:

initial state	$\langle 3, 0\rangle$
8 time units elapse, counter at level 3 updates	$\langle 3, 8\rangle$
one time unit elapses, *Glycemia* decays	$\langle 2, 0\rangle$
one time unit elapses, counter at level 2 updates	$\langle 2, 1\rangle$
reaction ρ_5 decreases *Glycemia* level	$\langle 1, 0\rangle$
8 time units elapse, counter at level 1 updates	$\langle 1, 8\rangle$
one time unit elapses, *Glycemia* decays	$\langle 0, 0\rangle$
one time unit elapses, no effect since $\delta_{glycemia}(0) = \omega$	$\langle 0, 0\rangle$. ◇

More formally, we now introduce the high-level Petri net modeling. Each species $s \in \mathcal{S}$ is modeled by a single place q_s whose type $L(q_s)$ is the set of tuples of the form $\langle l_s, u_s\rangle$, where $l_s \in [0..\mathcal{L}_s - 1]$ and $u_s \in [0..max_s]$, with $max_s = \mathsf{max}\{\delta_s(l) \mid \delta_s(l) \neq \omega \text{ and } l \in [0..\mathcal{L}_s - 1]\}$. In order to cope with time aspects we introduce a transition t_c (Figure 3(a)) connected to all species that is responsible for time progression and takes care of the decay of concerned species (as described in Case 1 above). Finally, every reaction ρ is modeled with a transition t_ρ (Figure 3(b)). To each transition t_ρ we associate a special place q_ρ that is used to ensure that the same reaction is not executed more than once in the same time unit. More detailed explanations for each type of transition follow Definition 2.


q_s
$\langle l_s, u_s\rangle$
t_c
$\langle l'_s, u'_s\rangle$
w_ρ
1
q_ρ
q_ρ
w_ρ
0
$\langle l_r, u_r\rangle$
t_ρ
$\langle l_i, u_i\rangle$
q_r
q_i
$\langle l_p, u_p\rangle$
$\langle l'_p, u'_p\rangle$
q_p

(a) Clock transition with only one place of each kind (q_s for $s \in \mathcal{S}$ and q_ρ for $\rho \in \mathcal{R}$).

(b) Transition for reaction $\rho = (R, I, P)$ with one place of each kind (q_ρ, q_r for $r \in R$, q_i for $i \in I$, q_p for $p \in P$).

Fig. 3. Scheme of Petri net modeling of reaction networks.

Definition 2. *Given a network $(\mathcal{S}, \mathcal{R})$ with initial state (s, η_s) for each $s \in \mathcal{S}$, its high-level Petri net representation is defined as tuple (Q, T, F, L, M_0) where $z, z', l, l', u, u', w, w'$ are variables and:*

- $Q = \{q_s \mid s \in \mathcal{S}\} \cup \{q_\rho \mid \rho \in \mathcal{R}\}$;
- $T = \{t_c\} \cup \{t_\rho \mid \rho \in \mathcal{R}\}$*;*
- $F = \{(q, t_c), (t_c, q) \mid q \in Q\} \quad \cup$
  $\{(q_s, t_\rho), (t_\rho, q_s), (q_\rho, t_\rho), (t_\rho, q_\rho) \mid \rho \in \mathcal{R}, s \in R_\rho \cup I_\rho \cup P_\rho\}$
- *Labels for places in Q:*

$$\begin{gathered} L(q_\rho) = \{0,1\} \textit{ for each } \rho \in \mathcal{R} \\ L(q_s) = [0..\mathcal{L}_s - 1] \times [0..max_s] \textit{ for each } s \in \mathcal{S} \end{gathered}$$

- *Labels for arcs in F:*

$$\begin{array}{ll} L((q_\rho, t_c)) = w & L((t_c, q_c)) = 1 \\ L((q_s, t_c)) = \langle l_s, u_s \rangle & L((t_c, q_s)) = \langle l'_s, u'_s \rangle \textit{ for each } s \in \mathcal{S} \end{array}$$

  For each reaction $\rho \in \mathcal{R}$ and $s \in R_\rho \cup I_\rho \cup P_\rho$:

$$\begin{array}{ll} L((q_s, t_\rho)) = \langle l_s, u_s \rangle & L((t_\rho, q_s)) = \begin{cases} \langle l_s, u_s \rangle & \textit{if } s \notin P_\rho \\ \langle l'_s, u'_s \rangle & \textit{otherwise} \end{cases} \\ L((q_\rho, t_\rho)) = w & L((t_\rho, q_\rho)) = 0 \end{array}$$

- *Labels for transitions in T:*

$$\begin{aligned} L(t_c) = \textstyle\bigwedge_{s \in \mathcal{S}} \big(& (\delta(l_s) = \omega \vee u_s + 1 \leq \delta(l_s)) \rightarrow \langle l'_s, u'_s \rangle = \langle l_s, u_s + 1 \rangle \quad \wedge \\ & (\delta(l_s) \neq \omega \wedge u_s + 1 > \delta(l_s)) \rightarrow \langle l'_s, u'_s \rangle = \langle l_s - 1, 0 \rangle \big). \end{aligned}$$

  For each reaction $\rho \in \mathcal{R}$:

$$\begin{aligned} L(t_\rho) = {} & (w = 1) \wedge \textstyle\bigwedge_{(r,\eta_r) \in R_\rho} (l_r \geq \eta_r) \wedge \bigwedge_{(i,\eta_i) \in I_\rho} (l_i < \eta_i) \quad \wedge \\ & \textstyle\bigwedge_{(p,z) \in P_\rho} (\langle l'_p, u'_p \rangle = \langle \max(0, \min(l_p + z, \mathcal{L}_p - 1)), 0 \rangle \end{aligned}$$

- *For each $q \in Q$, $s \in \mathcal{S}$ and $\rho \in \mathcal{R}$, the initial marking M_0 is:*

$$M_0(q) = \begin{cases} 1 & \textit{if } q = q_\rho, \\ \langle \eta_s, 0 \rangle & \textit{if } q = q_s. \end{cases}$$

We now comment on the transitions of the high-level Petri net. The result of the firing of a transition is handled by guards (namely transition labels $L(t_c)$ and $L(t_\rho)$) together with the evaluation σ as described after Definition 1. With an abuse of notation, in the following, we refer to evaluated variables without effectively mentioning the evaluation σ: i.e., we say that the current value of the token in q_ρ is w instead of $\sigma(w)$. Input and output arcs between the same place and transition with the same label (read arcs) are denoted in figures with a double-pointed arrow with a single label.

Clock transition t_c, depicted in Figure 3(a), takes care of Case 1 above. t_c is responsible for the decay of concerned species and the related update of counters u_s of each species. Moreover t_c updates the tokens of all places q_ρ to 1, thus re-enabling the possibility of performing a reaction ρ.

Next, we describe transitions for reactions, depicted in Figure 3(b). Given a reaction $\rho = (R, I, P)$ we detail the conditions and the results of firing of t_ρ. As described in Case 2 we have:

- each reactant $r \in R$ has to be present at least at level η_r, this is expressed by guard $l_r \geq \eta_r$;
- each inhibitor $i \in I$ has not to exceed level η_i, this is guaranteed by guard $(l_i < \eta_i)$;
- each product $p \in P$ corresponding to place q_p is updated to $\langle l'_p, u'_p \rangle = \langle \max(0, \min(l_p + z, \mathcal{L}_p - 1)), 0 \rangle$.

The role of place q_ρ is to forbid two consecutive executions of the same reaction in the same time unit. Initially, the marking of q_ρ is set to 1 and it becomes 0 when the transition t_ρ is fired; then clock transition t_c sets it to 1 again.

Observe that, because of the semantics of high level Petri nets, reaction may not occur even if all constraints are satisfied. This is interpreted as the action of an hostile (non-specified) environment (e.g., reactants are too far from each other to react).

Example 4 (Glucose metabolism – reaction network).

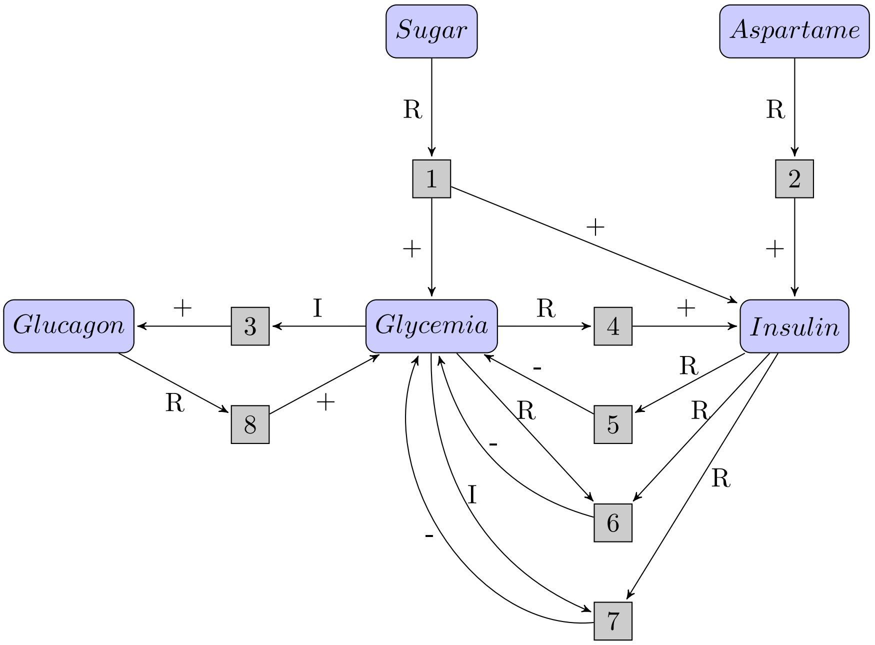


Fig. 4. Simplified reaction network of glucose metabolism.

Figure 4 shows a simplification of the reaction network $(\mathcal{S}, \mathcal{R})$ given in example 1. It focuses only on the reaction schema linking inputs (i.e., reactants and inhibitors) to products. Each input arc is labeled with either letter R or letter I denoting whether the input place is a reactant or an inhibitor, respectively. Likewise, each output arc is labeled with a + or a - to denote increase or decrease of product levels by 1. For each reaction transition ρ, we have omitted place q_ρ and all arcs in the opposite direction. The numbers inside each transition refers to the corresponding reaction in Example 2.

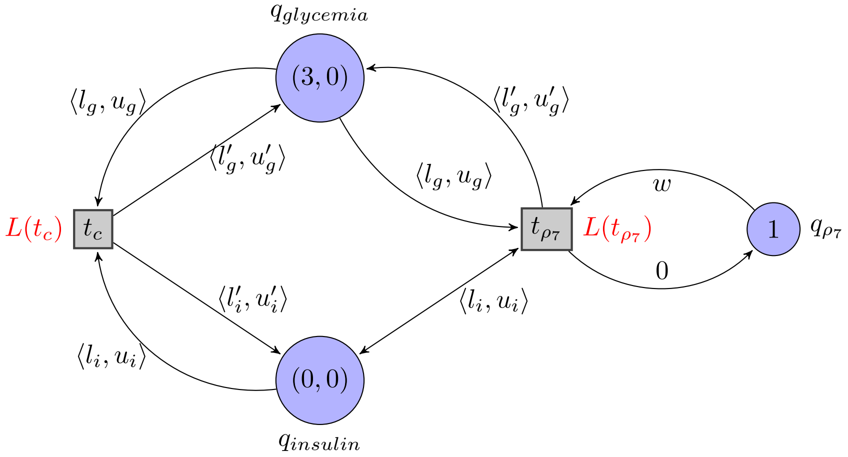

$q_{glycemia}$
$(3,0)$
$\langle l_g, u_g \rangle$
$\langle l'_g, u'_g \rangle$
$\langle l'_g, u'_g \rangle$
$\langle l_g, u_g \rangle$
w
$L(t_c)$
t_c
t_{ρ_7}
$L(t_{\rho_7})$
1
q_{ρ_7}
0
$\langle l'_i, u'_i \rangle$
$\langle l_i, u_i \rangle$
$\langle l_i, u_i \rangle$
$(0,0)$
$q_{insulin}$

Fig. 5. A portion of the reaction network of glucose metabolism with an initial marking.

Figure 5, instead, shows a portion of the complete initially marked reaction network for the glucose metabolism example, focusing only on reaction ρ_7. ◇

From the above definition and the transition rule of high-level nets, we have the following properties:

Proposition 1. *Given a reaction network $(\mathcal{S}, \mathcal{R})$ with initial values η_s for each species $s \in \mathcal{S}$:*

- *its Petri net representation has a finite structure with $|\mathcal{S}| + |\mathcal{R}|$ places, $|\mathcal{R}| + 1$ transitions and the number of arcs is bounded by $2(|\mathcal{S}| + |\mathcal{R}| + \Sigma_{\rho=(R_\rho, I_\rho, P_\rho)\in\mathcal{R}}(|R_\rho| + |I_\rho| + |P_\rho| + 1)$;*
- *each place type is a finite set;*
- *for each arc $(q,t) \in F$ there is an arc in opposite direction, i.e., $(t,q) \in F$ and each arc label is a singleton;*
- *the initial marking and all reachable markings have exactly one structured token per place;*
- *the number of all reachable markings from the initial one is finite.*

Proof. Follows by definition and by induction on the length of a firing sequence. □

5 Toxicology analysis

Such a Petri net representation of a reaction network is used to detect and predict toxic behaviors related to the dynamics of bio-molecular networks. In order to verify toxicology properties, we resort to temporal logics and model checking techniques [5]. For the sake of the present paper computation tree logic (CTL) allows to express properties of interest. Nonetheless different scenarios may require other more appropriate modal logic which we could be handled by our framework.

We recall here the basic concepts of CTL, provide the formal definition of the syntax and give some intuitions on the semantics, formally defined in [9].

A CTL formula is defined as:

$$\begin{aligned}\varphi ::= {} & \bot \mid a \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \\ & \mathbf{EX}\varphi \mid \mathbf{EG}\varphi \mid \mathbf{E}(\varphi\mathbf{U}\varphi) \mid \mathbf{EF}\varphi \mid \mathbf{AG}\varphi \mid \mathbf{AF}\varphi\end{aligned}$$

where $a \in A$ is an atomic proposition.

CTL is used to state properties on branching time structures. It uses usual boolean operators, path quantifiers and temporal operators. Path quantifiers can be of two kinds: $\mathbf{A}\varphi$ means that φ has to hold on all paths starting from the current state, while $\mathbf{E}\varphi$ stands for there exists at least one path starting from the current state where φ holds. We have four temporal operators: $\mathbf{X}\varphi$ holds if φ is true at the *next* state, $\mathbf{G}\varphi$ means that φ has to *globally* hold on the entire subsequent path, $\mathbf{F}\varphi$ stands for eventually (or *finally*) φ has to hold (at some point on the subsequent path), and $\varphi_1\mathbf{U}\varphi_2$ means that φ_1 has to hold at least *until* at some position φ_2 holds. In our context atomic formulae are represented by pairs of species and levels: $A = \{(s, \eta_s) \mid s \in \mathcal{S}\}$, for instance $(Glucose, 2)$.

As mentioned in the introduction, we are mainly interested in checking whether the inner equilibrium of an organism (tissue, cell, ...) is maintained when administrating drugs or applying stressors. More in detail, toxicology properties can be classified into two categories:

- properties checking for the appearance of particular symptoms, and
- properties characterizing causal relations between events.

The former class of properties basically consists in verifying reachability of some states, while the latter concerns pathways that highlight sequences of events leading to toxic outcomes. For instance, in the case of glucose regulation, we could verify whether glycemia levels are kept stable and whether they change in case of ingestion of aspartame. More precisely, we could examine the causes and the symptoms of the hypoglycemia induced by the assimilation of aspartame. Hence hypoglycemia is treated as a toxic state.

Example 5 (Glucose metabolism – properties). Take our running example of blood glucose regulation. The following properties can be expressed in CTL:

Symptoms: Is it possible to have an anomalous decrease of glucose levels in blood (revealing hypoglycemia)?

$$\mathbf{EF}(Glycemia, 0)$$

Mode-of-action: Recalling that the blood glucose regulation process normally maintains glycemia at equilibrium (level 2), is there an abnormal behavior leading to hypoglycemia?

$$\mathbf{E}(\mathbf{EF}(\mathit{Glycemia}, 2)\ \mathbf{U}\ (\mathbf{EF}(\mathit{Glycemia}, 0)))$$

Causality: Does assimilation of sweeteners cause hypoglycemia?

$$\mathbf{EF}[((\mathit{Sugar}, 1) \vee (\mathit{Aspartame}, 1)) \wedge (\mathit{Glycemia}, 1)] \rightarrow \mathbf{AF}(\mathit{Glycemia}, 2)$$

For the third formula we show two paths given as sequences of reactions (abstracting away from time transitions), one that satisfy the formula and the other that contradicts it. The first one corresponds to the assimilation of sugar. As described in Section 3, the digestion of sugar induces an increase of the production of insulin and an augmentation of the blood glucose levels. Nonetheless the levels of insulin produced are not enough to cause the glycemia to drop and the formula is satisfied.

$$\begin{array}{l}(\mathit{Sugar}, 1), (\mathit{Aspartame}, 0), (\mathit{Glycemia}, 1), (\mathit{Insulin}, 0), (\mathit{Glucagon}, 0) \xrightarrow{\rho_1} \\ (\mathit{Sugar}, 1), (\mathit{Aspartame}, 0), (\mathbf{Glycemia}, \mathbf{2}), (\mathit{Insulin}, 1), (\mathit{Glucagon}, 0)\end{array}$$

Unlike previous path, the assimilation of aspartame causes only an increase of insulin. Unfortunately, this increment is sufficient to induce a decrease of blood glucose levels thus contradicting the formula above.

$$\begin{array}{l}(\mathit{Sugar}, 0), (\mathit{Aspartame}, 1), (\mathit{Glycemia}, 1), (\mathit{Insulin}, 0), (\mathit{Glucagon}, 0) \xrightarrow{\rho_2} \\ (\mathit{Sugar}, 0), (\mathit{Aspartame}, 1), (\mathit{Glycemia}, 1), (\mathit{Insulin}, 1), (\mathit{Glucagon}, 0) \xrightarrow{\rho_7} \\ (\mathit{Sugar}, 0), (\mathit{Aspartame}, 0), (\mathbf{Glycemia}, \mathbf{0}), (\mathit{Insulin}, 1), (\mathit{Glucagon}, 0)\end{array}$$

This illustrates the toxic behavior caused by aspartame described in Section 3. ◇

6 Related work

The main application of our work concerns the verification of properties of systems defined in terms of rules or reactions. From a technical point of view, the closest related work is on reaction systems [1] or their Petri net representation [17]. Although we use a similar definition for reactions, the semantics that we have proposed is inherently different: in [1] all enabled reactions occur in one step while we have considered an interleaving semantics. In [2], the authors consider an extension of reaction systems with a notion of decay, this concept is different from the one considered here as we refer to an independent time progression while they count the number of maximally concurrent steps. In fact, our representation of time is considerably different from the approaches traditionally used in time and timed Petri nets ([3] presents a survey with insightful comparison of the different approaches). The main difference lies on the fact that the progression of time is implicit and external to the system. By contrast, in our proposal we have assumed the presence of an explicit way of incrementing durations (modeled by synchronized counters). This is also different from the notion of timestamps introduced in [12] that again refers to an implicit notion of time. Indeed, our approach is conceptually closer to Petri nets with causal time [22] for the presence of an explicit transition for time progression. Nevertheless, in reaction networks time cannot be suspended under the influence of the environment (as is the case in [22]).

In a broader sense, our work could also be related to P-systems [18,16] or the κ-calculus [6] that describe the evolution of cells through rules. Both these approaches are mainly oriented to simulation while we are interested in verification aspects. Finally, always related to the modeling in Petri nets but with a different aim, levels have been used in qualitative approaches to address problems related to the identification of steady states in genetic networks such as in [4]. Nevertheless these contributions abstract away from time related aspects that are instead central in our proposal.

7 Conclusion and future work

We have introduced a high-level Petri net modeling of reaction networks to address problems related to toxicogenomics. In reaction networks, systems consist of a set of species present in the environment at a given level. Species can degrade with time progression and their presence is governed by a set of rules (reactions). In a reaction, species can have the role of reactants, inhibitors or products. A reaction can take place only if all reactants are available and all inhibitors are not. Depending on the type of reaction, products levels are either increased or decreased. We have shown that properties of biological systems can be expressed in a suitable temporal logic and verified on the finite state space of the network. We have illustrated our framework in the modeling of blood glucose regulation.

We are currently investigating how to enrich reactions with response time, representing the required time for yielding products [8]. This poses new questions on how our model with time constraints could be compared to other existing time concepts for instance that in timed automata or that in stochastic models like in [13,11].

Finally, we have a prototype implemented with Snakes [19] and we plan to use Snoopy [14] and connected tools (Marcie [20]) to simulate and analyze CTL formulae.

Acknowledgments. We would like to thank Michel Malo and the anonymous reviewers for their comments and insightful suggestions. This work was supported by the French project ANR BLANC 0307 01 - SYNBIOTIC.

References

1. R. Brijder, A. Ehrenfeucht, M. G. Main, and G. Rozenberg. A tour of reaction systems. *Journal of Foundations of Computer Science*, 22(7):1499–1517, 2011.

2. R. Brijder, A. Ehrenfeucht, and G. Rozenberg. Reaction systems with duration. In J. Kelemen and A. Kelemenová, editors, *Computation, Cooperation, and Life*, volume 6610 of *LNCS*, pages 191–202. Springer, 2011.
3. A. Cerone and A. Maggiolo-Schettini. Time-based expressivity of time petri nets for system specification. *TCS*, 216(1-2):1–53, 1999.
4. C. Chaouiya, A. Naldi, E. Remy, and D. Thieffry. Petri net representation of multi-valued logical regulatory graphs. *Natural Computing*, 10(2):727–750, 2011.
5. E. M. Clarke, O. Grumberg, and D. Peled. *Model checking*. MIT Press, 2001.
6. V. Danos and C. Laneve. Formal molecular biology. *TCS*, 325(1):69–110, 2004.
7. M. DeCristofaro and K. Daniels. Toxicogenomics in biomarker discovery. In D. Mendrick and W. Mattes, editors, *Essential Concepts in Toxicogenomics*, volume 460 of *Methods in Molecular Biology^{TM}*, pages 185–194. Humana Press, 2008.
8. C. Di Giusto, H. Klaudel, and F. Delaplace. Reaction networks with delays applied to toxicity analysis. Technical report, IBISC, 2014. Available at .
9. E. Emerson and E. M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Science of Comp. Programming*, 2(3):241 – 266, 1982.
10. W. R. Foster, S.-J. Chen, A. He, A. Truong, V. Bhaskaran, D. M. Nelson, D. M. Dambach, L. D. Lehman-McKeeman, and B. D. Car. A retrospective analysis of toxicogenomics in the safety assessment of drug candidates. *Toxicologic pathology*, 35(5):621–35, Aug. 2007.
11. D. Gilbert, M. Heiner, F. Liu, and N. Saunders. Colouring space - a coloured framework for spatial modelling in systems biology. In J. M. Colom and J. Desel, editors, *Petri Nets*, volume 7927 of *Lecture Notes in Computer Science*, pages 230–249. Springer, 2013.
12. H.-M. Hanisch, K. Lautenbach, C. Simon, and J. Thieme. Timestamp petri nets in technical applications. In *WODES '98*, pages 321–326, 1998.
13. M. Heiner, D. Gilbert, and R. Donaldson. Petri nets for systems and synthetic biology. In M. Bernardo, P. Degano, and G. Zavattaro, editors, *SFM*, volume 5016 of *Lecture Notes in Computer Science*, pages 215–264. Springer, 2008.
14. M. Heiner, M. Herajy, F. Liu, C. Rohr, and M. Schwarick. Snoopy - a unifying petri net tool. In S. Haddad and L. Pomello, editors, *Petri Nets*, volume 7347 of *Lecture Notes in Computer Science*, pages 398–407. Springer, 2012.
15. K. Jensen. *Coloured Petri Nets - Basic Concepts, Analysis Methods and Practical Use - Volume 1*. EATCS Monographs on TCS. Springer, 1992.
16. J. Kleijn and M. Koutny. Membrane systems with qualitative evolution rules. *Fundam. Inform.*, 110(1-4):217–230, 2011.
17. J. Kleijn, M. Koutny, and G. Rozenberg. Modelling reaction systems with petri nets. In *BioPPN-2011*, pages 36–52, 2011.
18. A. Paun, M. Paun, A. Rodríguez-Patón, and M. Sidoroff. P systems with proteins on membranes: a survey. *International Journal of Foundations of Computer Science*, 22(1):39–53, 2011.
19. F. Pommereau. Quickly prototyping Petri nets tools with SNAKES. *Petri net newsletter*, (10-2008):1–18, 10 2008. SNAKES is available here.
20. M. Schwarick, M. Heiner, and C. Rohr. Marcie - model checking and reachability analysis done efficiently. In *QEST*, pages 91–100. IEEE Computer Society, 2011.
21. L. Serrano. Synthetic biology: promises and challenges. *Molecular Systems Biology*, 3(158), 2007.
22. C. B. Thanh, H. Klaudel, and F. Pommereau. Petri nets with causal time for system verification. *ENTCS*, 68(5):85–100, 2002.
23. M. D. Waters and J. M. Fostel. Toxicogenomics and systems toxicology: aims and prospects. *Nature reviews. Genetics*, 5(12):936–48, Dec. 2004.

Integrating prior knowledge in automatic network reconstruction

Marie C.F. Favre[1]*, Wolfgang Marwan[2], Annegret K. Wagler[1]
{marie.favre,wagler}@isima.fr, wolfgang.marwan@ovgu.de

[1] Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes (LIMOS, UMR CNRS 6158), Université Blaise Pascal, Clermont-Ferrand, France
[2] Magdeburg Center for Systems Biology (MaCS), Otto-von-Guericke-Universität, Magdeburg, Germany

Abstract. The reconstruction of models from experimental data is a challenging problem due to the inherited complexity of biological systems. We developed an exact, exclusively data-driven approach to reconstruct Petri nets from experimental time-series data. Our approach aims at reconstructing all such netwoks that fit the given experimental data, to provide all possible alternatives of mechanisms behind the experimental observations, which typically results in a large set of solution alternatives. To keep this solution set reasonably small while still guaranteeing its completeness, we firstly generate only Petri nets being minimal in the sense that all other networks fitting the data contain the reconstructed ones. We further aim at avoiding the generation of minimal solutions which are "technically correct" but would be ruled out later during a subsequent verification process to check whether the returned solutions are "biological meaningful" or even contradict well-established biological knowledge. For that, we propose to extent the considered input (beyond the information given with the experimental time-series data) for the reconstruction process and demonstrate with the help of a running example the influence on the generated solution set.

1 Introduction

Systems biology aims at the integrated experimental and theoretical analysis of molecular or cellular networks to achieve a holistic understanding of biological systems and processes. To gain the required insight into the underlying biological processes, experiments are performed and experimental data are interpreted in terms of models. Depending on the biological aim, the type and quality of the available data, different types of mathematical models are used and corresponding reconstruction methods have been developed. Our work is dedicated to Petri nets which turned out to coherently model both static interactions in terms of networks and dynamic processes in terms of state changes, see *e.g.* [7,10].

* This work was funded by the French National Research Agency, the European Commission (Feder funds) and the Région Auvergne within the LabEx IMobS^3.

M. Heiner (Ed.): BioPPN 2014, a satellite event of PETRI NETS 2014,
CEUR Workshop Proceedings Vol. 1159, 2014.

In fact, a network $\mathcal{P} = (P, T, A, w)$ reflects the involved components by places $p \in P$ and their interactions by transitions $t \in T$, linked by weighted directed arcs $(p, t), (t, p) \in A$. Each place $p \in P$ can be marked with an integral number x_p of tokens defining a system state $\boldsymbol{x} \in \mathbb{Z}_+^{|P|}$, *i.e.*, we obtain $\mathcal{X} := \{\boldsymbol{x} \in \mathbb{Z}^{|P|} : x_p \geq 0\}$ as set of potential states. A transition $t \in T$ is enabled in a state $\boldsymbol{x}$ if $x_p \geq w(p, t)$ for all p with $(p, t) \in A$, we denote the set of all such transitions by $T(\boldsymbol{x})$. Switching $t \in T(\boldsymbol{x})$ yields a successor state $\text{succ}(\boldsymbol{x}) = \boldsymbol{x}'$ with $x'_p = x_p - w(p, t)$ for all $(p, t) \in A$ and $x'_p = x_p + w(t, p)$ for all $(t, p) \in A$. Dynamic processes are represented by sequences of such state changes.

Our central question is to reconstruct models of this type from experimental time-series data by means of an exact, exclusively data-driven approach. This approach takes as input a set P of places and discrete time-series data $\mathcal{X}' \subseteq \mathcal{X}$ given by sequences $(\boldsymbol{x}^0; \boldsymbol{x}^1, \ldots, \boldsymbol{x}^k)$ of experimentally observed system states. The goal is to determine all Petri nets (P, T, A, w) that are able to reproduce the data, *i.e.*, that perform for each $\boldsymbol{x}^j \in \mathcal{X}'$ the experimentally observed state change to $\boldsymbol{x}^{j+1} \in \mathcal{X}'$ in a simulation. Hence, in contrast to the normally used stochastic simulation, we require that for states where at least two transitions are enabled, the decision between the alternatives is not taken randomly, but a specific transition is selected. Thus, (standard) Petri nets have to be equipped with additional activation rules to force the switching of specific transitions (to reach $\boldsymbol{x}^{j+1}$ from $\boldsymbol{x}^j$), and to prevent all others from switching. For that, different types of additional activation rules are possible.

On the one hand, control-arcs are used to represent catalytic or inhibitory dependencies. An *extended Petri net* $\mathcal{P} = (P, T, (A \cup A_R \cup A_I), w)$ is a Petri net which has, besides the (standard) arcs in A, two additional sets of so-called control-arcs: the set of read-arcs $A_R \subset P \times T$ and the set of inhibitor-arcs $A_I \subset P \times T$; we denote the set of all arcs by $\mathcal{A} = A \cup A_R \cup A_I$. Here, a transition $t \in T(\boldsymbol{x})$ coupled with a read-arc (resp. an inhibitor-arc) to a place $p \in P$ can switch only if at least $w(p, t)$ tokens (resp. less than $w(p, t)$ tokens) are present in p; we denote by $T_{\mathcal{A}}(\boldsymbol{x})$ the set of all such transitions.

On the other hand, in [9,12,13] priority relations among the transitions of a network are employed to reflect the rate of the corresponding reactions, where the fastest reaction has highest priority and, thus, is taken. In Marwan et al. [9] it is proposed to model such priorities with the help of partial orders $\mathcal{O}$ on the transitions. We call $(\mathcal{P}, \mathcal{O})$ an *(extended) Petri net with priorities*, if $\mathcal{P} = (P, T, \mathcal{A}, w)$ is an (extended) Petri net and $\mathcal{O}$ a priority relation on T. Priorities can prevent enabled transitions from switching: For each state $\boldsymbol{x}$, a transition $t \in T_{\mathcal{A}}(\boldsymbol{x})$ is allowed to switch only if there is no other enabled transition in $T_{\mathcal{A}}(\boldsymbol{x})$ with higher priority; we denote by $T_{\mathcal{A},\mathcal{O}}(\boldsymbol{x})$ the set of all such transitions.

We call $(\mathcal{P}, \mathcal{O})$ *$\mathcal{X}'$-deterministic* if $T_{\mathcal{A},\mathcal{O}}(x)$ contains at most one element for each experimentally observed state $\boldsymbol{x} \in \mathcal{X}'$. Based on earlier results in [3,4,5,9,13], an integrative method to reconstruct all $\mathcal{X}'$-deterministic extended Petri nets with priorities fitting given experimental time-series data $\mathcal{X}'$ was proposed in [6] (see Section 2).

Our approach aims at reconstructing all netwoks of the studied type that fit the given experimental data, to provide all possible alternatives of mechanisms behind the experimentally observed phenomena. Typically, this results in a large set of solution alternatives. To keep this solution set reasonably small while still guaranteeing its completeness, we generate only Petri nets being minimal in the sense that all other networks fitting the data contain the reconstructed ones. Here, we propose a method to insert only minimal sets of control-arcs during the reconstruction process (see Section 2). We further aim at avoiding the generation of minimal solutions which are "technically correct" but would be ruled out later during a subsequent verification process to check whether the returned solutions are "biological meaningful" or even contradict well-established biological knowledge. For that, we extent the considered input by integrating biological prior knowledge (beyond the information given with the experimental time-series data) into the reconstruction process and demonstrate with the help of a running example the influence on the generated solution set (see Section 3). We close with some concluding remarks and lines of future work.

2 Reconstructing extended Petri nets with priorities

We describe the input, the main ideas, and the output of our approach from [6].

Input. A set of components P (standing for proteins, enzymes, genes, receptors or their conformational states, later represented by the set of places) is chosen which is expected to be crucial for the studied phenomenon. If P contains known P-invariants (subsets $P' \subseteq P$ of places where the sum of the number of all tokens on all the places in P' is constant, e.g., different functional states of a cell or conformational states of a molecular complex), they are collected in a set $\mathcal{I}_P$.

To perform an experiment, one first triggers the system in some state $\boldsymbol{x}^0$ (by external stimuli like the exposition to a pathogen), to generate an initial state $\boldsymbol{x}^1$. Then the system's response to the stimulation is observed and the resulting state changes are measured for all considered components at certain time points. This yields a sequence of (discrete or discretized) states $\boldsymbol{x}^j \in \mathbb{Z}^{|P|}$ reflecting the time-dependent response of the system to the stimulation in $\boldsymbol{x}^1$, which typically terminates in a terminal state $\boldsymbol{x}^k$ where no further changes are observed. The corresponding experiment is $\mathcal{X}'(\boldsymbol{x}^1, \boldsymbol{x}^k) = (\boldsymbol{x}^0; \boldsymbol{x}^1, \dots, \boldsymbol{x}^k)$. Several experiments starting from different initial states in a set $\mathcal{X}'_{ini} \subseteq \mathcal{X}'$, reporting the observed state changes, and ending at different terminal states in a set $\mathcal{X}'_{term} \subseteq \mathcal{X}'$ describe the studied phenomenon, and yield experimental time-series data of the form $\mathcal{X}' = \{\mathcal{X}'(\boldsymbol{x}^1, \boldsymbol{x}^k) : \boldsymbol{x}^1 \in \mathcal{X}'_{ini}, \boldsymbol{x}^k \in \mathcal{X}'_{term}\}$. Thus, the input of the reconstruction approach is given by $(P, \mathcal{I}_P, \mathcal{X}')$.

Example 1. As running example, we will consider experimental biological data from the *light-induced sporulation of Physarum polycephalum* as in [6,13]. In *P. polycephalum* plasmodia, the photoreceptor involved in the control of sporulation *Spo* is a protein which occurs in two stages P_{FR} and P_R. The developmental decision of starving *P. polycephalum* plasmodia to enter the sporulation pathway is controlled by environmental factors like visible light [11]. If the dark-adapted form P_{FR} absorbs far-red light FR, the receptor is converted into its red-absorbing form P_R, which causes sporulation after several hours [8]. If P_R is exposed to red light R, it is photo-converted back to the initial state P_{FR} (photoreversion), which prevents sporulation if the red light pulse is given shortly after the far-red pulse, but not if the red pulse is delivered after more than an hour when the phytochrome photoreceptor has had sufficient time to cause the formation of a biochemical downstream signal G that subsequently causes the sporulation of the cell. The changes between the stages P_{FR} and P_R only require fractions of seconds and can be experimentally observed due to a change of color of the phytochrome protein. The experimental setting consists of

$$\begin{array}{lll} P = \{FR, R, P_{FR}, P_R, G, Spo\}, & \mathcal{X}'(\boldsymbol{x}^1, \boldsymbol{x}^4) = (\boldsymbol{x}^0; \boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{x}^3, \boldsymbol{x}^4), & \mathcal{X}'_{ini} \ = \{\boldsymbol{x}^1, \boldsymbol{x}^5, \boldsymbol{x}^6\}, \\ \mathcal{I}_P = \{P_{FR}, P_R\}, & \mathcal{X}'(\boldsymbol{x}^5, \boldsymbol{x}^0) = (\boldsymbol{x}^2; \boldsymbol{x}^5, \boldsymbol{x}^0), & \mathcal{X}'_{term} = \{\boldsymbol{x}^4, \boldsymbol{x}^0, \boldsymbol{x}^8\} \\ & \mathcal{X}'(\boldsymbol{x}^6, \boldsymbol{x}^8) = (\boldsymbol{x}^3; \boldsymbol{x}^6, \boldsymbol{x}^7, \boldsymbol{x}^8), & \end{array}$$

as input for the algorithm, we present all observed states schematically in Fig 1.

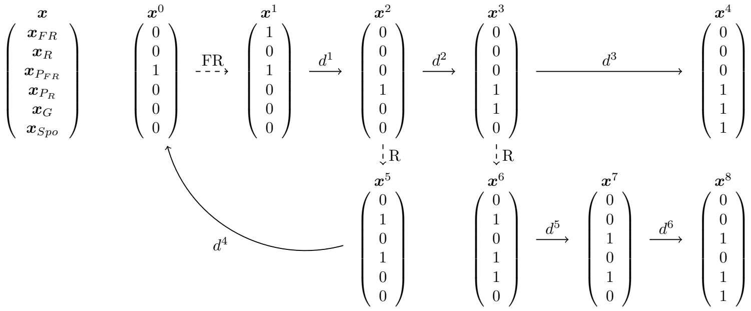

$\boldsymbol{x}$
$\boldsymbol{x}_{FR}$
$\boldsymbol{x}_R$
$\boldsymbol{x}_{P_{FR}}$
$\boldsymbol{x}_{P_R}$
$\boldsymbol{x}_G$
$\boldsymbol{x}_{Spo}$
$\boldsymbol{x}^0$ (0, 0, 1, 0, 0, 0)
FR
$\boldsymbol{x}^1$ (1, 0, 1, 0, 0, 0)
d^1
$\boldsymbol{x}^2$ (0, 0, 0, 1, 0, 0)
d^2
$\boldsymbol{x}^3$ (0, 0, 0, 1, 1, 0)
d^3
$\boldsymbol{x}^4$ (0, 0, 0, 1, 1, 1)
R
R
$\boldsymbol{x}^5$ (0, 1, 0, 1, 0, 0)
d^4
$\boldsymbol{x}^6$ (0, 1, 0, 1, 1, 0)
d^5
$\boldsymbol{x}^7$ (0, 0, 1, 0, 1, 0)
d^6
$\boldsymbol{x}^8$ (0, 0, 1, 0, 1, 1)

Fig. 1. Experimental time-series data $\mathcal{X}'$ for the light-induced sporulation of *Physarum polycephalum*. The experimental setting uses the set $P = \{FR, R, P_{fr}, P_r, G, Spo\}$ of studied components, observed states are represented by vectors of the form $\boldsymbol{x} = (x_{FR}, x_R, x_{P_{FR}}, x_{P_R}, x_G, x_{Spo})^T$ having 0/1-entries only. Dashed arrows represent a stimulation or perturbation of the system, solid arrows the observed responses.

For a successful reconstruction, the data $\mathcal{X}'$ need to satisfy two properties: reproducibility and monotonicity. The data $\mathcal{X}'$ are *reproducible* if for each $\boldsymbol{x}^j \in \mathcal{X}'$ there is a unique observed successor state $\text{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) = \boldsymbol{x}^{j+1} \in \mathcal{X}'$. Reproducibility is obviously necessary and can be ensured by preprocessing [15]. Whether a state $\boldsymbol{x}^j \in \mathcal{X}'$ and its observed successor $\text{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) = \boldsymbol{x}^{j+1} \in \mathcal{X}'$ are also consecutive system states depends on the chosen time points to measure the states in $\mathcal{X}'$. In fact, $\boldsymbol{x}^{j+1}$ may be obtained from $\boldsymbol{x}^j$ by a switching sequence of some length, where the intermediate states are not reported in $\mathcal{X}'$. The data $\mathcal{X}'$ are *monotone* if for each pair $(\boldsymbol{x}^j, \boldsymbol{x}^{j+1}) \in \mathcal{X}'$, the values of the elements do not oscillate in the possible intermediate states between $\boldsymbol{x}^j$ and $\boldsymbol{x}^{j+1}$. It was shown in [4] that monotonicity has to be required, too (which is equivalent to demand that all essential responses are indeed reported in $\mathcal{X}'$).

Output. An extended Petri net with priorities $(\mathcal{P}, \mathcal{O})$ with $\mathcal{P} = (P, T, \mathcal{A}, w)$ fits the given data $\mathcal{X}'$ when it is able to perform every observed state change from $\boldsymbol{x}^j \in \mathcal{X}'$ to $\boldsymbol{x}^{j+1} \in \mathcal{X}'$. For that, associate with $\mathcal{P}$ an incidence matrix $M \in \mathbb{Z}^{|P|\times|T|}$ whose rows correspond to the places $p \in P$ and whose columns $M_{\cdot t}$ to the *update vector* $\boldsymbol{r}^t$ of the transitions $t \in T$:

$$r_p^t = M_{pt} := \begin{cases} -w(p,t) & \text{if } (p,t) \in A, \\ +w(t,p) & \text{if } (t,p) \in A, \\ 0 & \text{otherwise.} \end{cases}$$

Reaching $\boldsymbol{x}^{j+1}$ from $\boldsymbol{x}^j$ by a switching sequence using the transitions from a subset $T' \subseteq T$ is equivalent to obtain $\boldsymbol{x}^{j+1}$ from $\boldsymbol{x}^j$ by adding the corresponding columns $M_{\cdot t}$ of M for all $t \in T'$, i.e., $\boldsymbol{x}^j + \sum_{t \in T'} M_{\cdot t} = \boldsymbol{x}^{j+1}$.

T has to contain enough transitions to perform all experimentally observed switching sequences. The underlying standard network $\mathcal{P} = (P, T, A, w)$ is *conformal* with $\mathcal{X}'$ if, for any two consecutive states $\boldsymbol{x}^j, \boldsymbol{x}^{j+1} \in \mathcal{X}'$, the linear equation system $\boldsymbol{x}^{j+1} - \boldsymbol{x}^j = M\boldsymbol{\lambda}$ has an integral solution $\boldsymbol{\lambda} \in \mathbb{N}^{|T|}$ such that $\boldsymbol{\lambda}$ represents a sequence $(t^1, ..., t^m)$ of transition switches, *i.e.*, there are intermediate states $\boldsymbol{x}^j = \boldsymbol{y}^1, \boldsymbol{y}^2, ..., \boldsymbol{y}^{m+1} = \boldsymbol{x}^{j+1}$ with $\boldsymbol{y}^l + M_{\cdot t^l} = \boldsymbol{y}^{l+1}$ for $1 \leq l \leq m$. The extended Petri net $\mathcal{P} = (P, T, \mathcal{A}, w)$ is *catalytic conformal* with $\mathcal{X}'$ if $t^l \in T_{\mathcal{A}}(\boldsymbol{y}^l)$ for each intermediate state $\boldsymbol{y}^l$, and the extended Petri net with priorities $(\mathcal{P}, \mathcal{O})$ is $\mathcal{X}'$*-deterministic* if $\{t^l\} = T_{\mathcal{A},\mathcal{O}}(\boldsymbol{y}^l)$ holds for all $\boldsymbol{y}^l$.

The desired output consists of all minimal $\mathcal{X}'$-deterministic extended Petri nets $(\mathcal{P}, \mathcal{O})$ (all having the same set P of places as part of the input).

Example 2. We represent in Fig. 3 (page 54) several alternative $\mathcal{X}'$-deterministic extended Petri nets fitting the experimental data $\mathcal{X}'$ from our running example.

We now briefly sketch the reconstruction procedure.

Representing the observed responses. As initial step, extract the observed changes of states from the experimental data. For that, define the set

$$\mathcal{D} := \left\{ \boldsymbol{d}^j = \boldsymbol{x}^{j+1} - \boldsymbol{x}^j \ : \ \boldsymbol{x}^{j+1} = \operatorname{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) \in \mathcal{X}' \right\}.$$

Generating the complete list of all $\mathcal{X}'$-deterministic extended Petri nets $\mathcal{P} = (P, T, \mathcal{A}, w)$ includes finding the corresponding standard networks and their incidence matrices $M \in \mathbb{Z}^{|P|\times|T|}$. Due to monotonicity [4], it suffices to represent any $\boldsymbol{d}^j \in \mathcal{D}$ using sign-compatible update vectors from the following set only:

$$\operatorname{Box}(\boldsymbol{d}^j) = \left\{ \boldsymbol{r} \in \mathbb{Z}^{|P|} : \begin{array}{r} 0 \leq r_p \leq d_p \text{ if } d_p^j > 0 \\ d_p \leq r_p \leq 0 \ \text{ if } d_p^j < 0 \\ r_p = 0 \ \text{ if } d_p^j = 0 \\ \sum_{p \in P'} r_p = 0 \ \ \forall P' \in \mathcal{I}_P \end{array} \right\} \setminus \{\boldsymbol{0}\}.$$

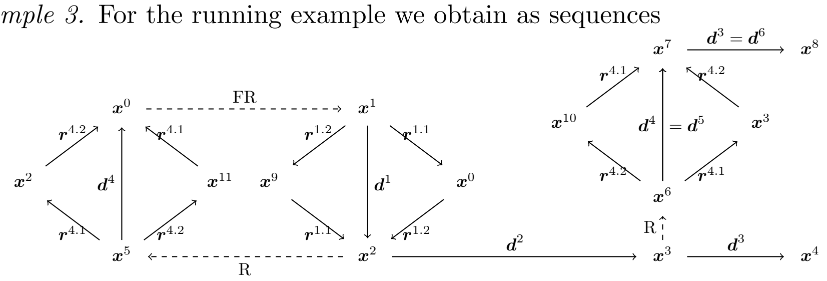
Next, we determine for any $\boldsymbol{d}^j \in \mathcal{D}$, the set $\Lambda(\boldsymbol{d}^j)$ of all integral solutions of

$$\boldsymbol{d}^j = \sum_{\boldsymbol{r}^t \in \operatorname{Box}(\boldsymbol{d}^j)} \lambda_t \boldsymbol{r}^t, \ \lambda_t \in \mathbb{Z}_+, \tag{1}$$

and for each $\lambda \in \Lambda(\boldsymbol{d}^j)$, the (multi-)set $\mathcal{R}(\boldsymbol{d}^j, \lambda) = \{\boldsymbol{r}^t \in \mathrm{Box}(\boldsymbol{d}^j) : \lambda_t \neq 0\}$ of update vectors used for this solution λ. By construction, $\mathrm{Box}(\boldsymbol{d}^j)$ and $\Lambda(\boldsymbol{d}^j)$ are always non-empty since $\boldsymbol{d}^j$ itself is always a solution due to reproducibility [6]. Every permutation $\pi = (\boldsymbol{r}^{t_1}, \ldots, \boldsymbol{r}^{t_m})$ of the elements of a set $\mathcal{R}(\boldsymbol{d}^j, \lambda)$ gives rise to a sequence of intermediate states $\boldsymbol{x}^j = \boldsymbol{y}^1, \boldsymbol{y}^2, ..., \boldsymbol{y}^m, \boldsymbol{y}^{m+1} = \boldsymbol{x}^{j+1}$ with

$$\sigma = \sigma_{\pi,\lambda}(\boldsymbol{x}^j, \boldsymbol{d}^j) = \left((\boldsymbol{y}^1, \boldsymbol{r}^{t_1}), (\boldsymbol{y}^2, \boldsymbol{r}^{t_2}), \ldots, (\boldsymbol{y}^m, \boldsymbol{r}^{t_m})\right).$$

Example 3. For the running example we obtain as sequences



with $\boldsymbol{x}^9 = (1, 0, 0, 1, 0, 0)^T$, $\boldsymbol{x}^{10} = (0, 1, 1, 0, 1, 0)^T$ and $\boldsymbol{x}^{11} = (0, 1, 1, 0, 0, 0)^T$.

To compose all possible standard networks, we have to select exactly one solution $\boldsymbol{\lambda} \in \Lambda(\boldsymbol{d}^j)$ for each $\boldsymbol{d}^j \in \mathcal{D}$ and to take the union of the corresponding sets $\mathcal{R}(\boldsymbol{d}^j, \boldsymbol{\lambda})$ in order to yield the columns $M_{\cdot t} = \boldsymbol{r}^t$ of an incidence matrix M of a conformal network. To ensure that the generated conformal networks can be made $\mathcal{X}'$-deterministic, we proceed as follows.

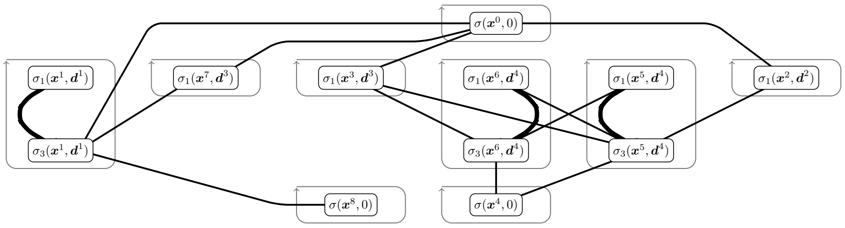
Detecting priority conflicts between sequences. By construction, every sequence $\sigma_{\pi,\boldsymbol{\lambda}}(\boldsymbol{x}^j, \boldsymbol{d}^j)$ induces a priority relation $\mathcal{O}_\sigma$, since it implies which transition t^i is supposed to have highest priority (and thus switches) for every intermediate state $\boldsymbol{y}^i$. To impose valid priority relations $\mathcal{O}$ among all transitions of the reconstructed networks, we have to take conflicts between priority relations $\mathcal{O}_\sigma$ induced by different sequences σ into account. Two sequences σ and σ' are in *priority conflict* if there are update vectors $\boldsymbol{r}^t \neq \boldsymbol{r}^{t'}$ and intermediate states $\boldsymbol{y}, \boldsymbol{y}'$ such that $t, t' \in T(\boldsymbol{y}) \cap T(\boldsymbol{y}')$ and $(\boldsymbol{y}, \boldsymbol{r}^t) \in \sigma$ but $(\boldsymbol{y}', \boldsymbol{r}^{t'}) \in \sigma'$ (since this implies $t > t'$ in $\mathcal{O}_\sigma$ but $t' > t$ in $\mathcal{O}_{\sigma'}$). We have a weak (resp. strong) priority conflict if $\boldsymbol{y} \neq \boldsymbol{y}'$ (resp. $\boldsymbol{y} = \boldsymbol{y}'$) which can (resp. cannot) be resolved by adding control-arcs.

We construct a *priority conflict graph* $\mathcal{G} = (V_D \cup V_{term}, E_D \cup E_W \cup E_S)$ whose nodes correspond to all possible sequences $\sigma_{\pi,\lambda}(\boldsymbol{x}^j, \boldsymbol{d}^j)$ and whose edges reflect weak and strong priority conflicts (WPC and SPC for short):

- V_D contains the sequences $\sigma_{\pi,\boldsymbol{\lambda}}(\boldsymbol{x}^j, \boldsymbol{d}^j)$ for all $\boldsymbol{x}^j \in \mathcal{X}' \setminus \mathcal{X}'_{term}$ and $\boldsymbol{d}^j = \mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) - \boldsymbol{x}^j$, for all $\boldsymbol{\lambda} \in \Lambda(\boldsymbol{d}^j)$ and all permutations π of $\mathcal{R}(\boldsymbol{d}^j, \boldsymbol{\lambda})$.
- V_{term} contains for all $\boldsymbol{x}^k \in \mathcal{X}'_{term}$ the trivial sequence $\sigma(\boldsymbol{x}^k, \boldsymbol{0})$.
- E_D contains all edges between two sequences σ, σ' stemming from the same difference vector.
- E_S (resp. E_W) reflects all SPCs (resp. WPCs) between sequences σ, σ' stemming from distinct difference vectors.

In $\mathcal{G}$, we generate all node subsets S selecting exactly one sequence $\sigma_{\pi,\lambda}(\boldsymbol{x}^j, \boldsymbol{d}^j)$ per difference vector $\boldsymbol{d}^j \in \mathcal{D}$ such that no SPCs occur between the selected sequences and the nodes in V_{term}. Clearly, a node $\sigma \in V_D$ can never be selected for any solution S if it is in SPC with a terminal state sequence or with all sequences σ' stemming from one difference vector $\boldsymbol{d}^j \in \mathcal{D}$. Removing all such nodes and their incident edges from $\mathcal{G}$ yields the *reduced priority conflict graph* $\mathcal{G}'$. We can show that the sets of suitable selections S obtained in $\mathcal{G}$ and $\mathcal{G}'$ are equal and that there is always at least one feasible selection.

Example 4. We obtain the following reduced priority conflict graph $\mathcal{G}'$ for the running example, where bold edges indicate SPCs and thin edges WPCs.


$\sigma(\boldsymbol{x}^0, 0)$
$\sigma_1(\boldsymbol{x}^1, \boldsymbol{d}^1)$
$\sigma_1(\boldsymbol{x}^7, \boldsymbol{d}^3)$
$\sigma_1(\boldsymbol{x}^3, \boldsymbol{d}^3)$
$\sigma_1(\boldsymbol{x}^6, \boldsymbol{d}^4)$
$\sigma_1(\boldsymbol{x}^5, \boldsymbol{d}^4)$
$\sigma_1(\boldsymbol{x}^2, \boldsymbol{d}^2)$
$\sigma_3(\boldsymbol{x}^1, \boldsymbol{d}^1)$
$\sigma_3(\boldsymbol{x}^6, \boldsymbol{d}^4)$
$\sigma_3(\boldsymbol{x}^5, \boldsymbol{d}^4)$
$\sigma(\boldsymbol{x}^8, 0)$
$\sigma(\boldsymbol{x}^4, 0)$

From $\mathcal{G}'$, we obtain as feasible subsets $S_i = S' \cup S'_i$ with

$$S' = \{\sigma_1(\boldsymbol{x}^2, \boldsymbol{d}^2), \sigma_1(\boldsymbol{x}^3, \boldsymbol{d}^3), \sigma_1(\boldsymbol{x}^7, \boldsymbol{d}^3)\}$$

$$\begin{aligned}
S'_1 &= \{\sigma_1(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_1(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_1(\boldsymbol{x}^6, \boldsymbol{d}^4)\}, & S'_5 &= \{\sigma_3(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_1(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_1(\boldsymbol{x}^6, \boldsymbol{d}^4)\}, \\
S'_2 &= \{\sigma_1(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_1(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_3(\boldsymbol{x}^6, \boldsymbol{d}^4)\}, & S'_6 &= \{\sigma_3(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_1(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_3(\boldsymbol{x}^6, \boldsymbol{d}^4)\}, \\
S'_3 &= \{\sigma_1(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_3(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_1(\boldsymbol{x}^6, \boldsymbol{d}^4)\}, & S'_7 &= \{\sigma_3(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_3(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_1(\boldsymbol{x}^6, \boldsymbol{d}^4)\}, \\
S'_4 &= \{\sigma_1(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_3(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_3(\boldsymbol{x}^6, \boldsymbol{d}^4)\}, & S'_8 &= \{\sigma_3(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_3(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_3(\boldsymbol{x}^6, \boldsymbol{d}^4)\}.
\end{aligned}$$

Constructing $\mathcal{X}'$-deterministic Petri nets. Every selected subset S corresponds to a conformal standard network $\mathcal{P}_S = (P, T_S, A_S, w)$: we obtain the columns of the incidence matrix M_S of the network by taking the union of all sets $\mathcal{R}(\boldsymbol{d}^j, \boldsymbol{\lambda})$ corresponding to the sequences $\sigma = \sigma_{\pi,\boldsymbol{\lambda}}(\boldsymbol{x}^j, \boldsymbol{d}^j)$ selected by $\sigma \in S$.

Example 5. We apply the method to the feasible sets S_1 and S_4 from Exp. 4 and obtain the standard networks presented in Fig. 2 with $T_{S_1} = \mathcal{D}$ and in Fig. 3 with $T_{S_4} = \{\boldsymbol{d}^1, \boldsymbol{d}^2, \boldsymbol{d}^3, \boldsymbol{r}^{4.1}, \boldsymbol{r}^{4.2}\}$, respectively.

If there are weak priority conflicts $\sigma\sigma' \in E_W$ for nodes $\sigma, \sigma' \in S \cup V_{term}$, denoted by $\mathrm{WPC}(\sigma, \sigma')$, the constructed standard network $\mathcal{P}_S$ needs to be made $\mathcal{X}'$-deterministic by inserting appropriate control-arcs. By [6], a $\mathrm{WPC}(\sigma, \sigma')$ between two sequences σ and σ' involving update vectors $\boldsymbol{r}^t \neq \boldsymbol{r}^{t'}$ and intermediate states $\boldsymbol{y} \neq \boldsymbol{y}'$ with $t, t' \in T(\boldsymbol{y}) \cap T(\boldsymbol{y}')$ such that $(\boldsymbol{y}, \boldsymbol{r}^t) \in \sigma$ but $(\boldsymbol{y}', \boldsymbol{r}^{t'}) \in \sigma'$ has to be resolved by adding appropriate control-arcs that either turn $\boldsymbol{r}^t$ into a transition t which is disabled at $\boldsymbol{y}'$ (then $t > t'$ forces t to switch in $\boldsymbol{y}$ whereas only t' is enabled at $\boldsymbol{y}'$), or vice versa. Let $P(\boldsymbol{y}, \boldsymbol{y}')$ be the set of places where $\boldsymbol{y}$ and $\boldsymbol{y}'$ differ and $\mathrm{CA}(\sigma, \sigma')$ the set of all possible control-arcs that can resolve $\mathrm{WPC}(\sigma, \sigma')$. Then $\mathrm{CA}(\sigma, \sigma')$ partitions into two subsets $\mathrm{CA}_{t>t'}(\sigma, \sigma')$ disabling t at $\boldsymbol{y}'$ containing

- a read-arc $(p,t) \in A_R$ with weight $w(p,t) > y'_p \ \forall p \in P(\boldsymbol{y},\boldsymbol{y'})$ with $y_p > y'_p$,
- an inhibitor-arc $(p,t) \in A_I$ with $w(p,t) < y_p \ \forall p \in P(\boldsymbol{y},\boldsymbol{y'})$ with $y_p < y'_p$,

and $\mathrm{CA}_{t<t'}(\sigma,\sigma')$ disabling t' at $\boldsymbol{y}$ containing

- a read-arc $(p,t') \in A_R$ with weight $w(p,t') > y_p \ \forall p \in P(\boldsymbol{y},\boldsymbol{y'})$ with $y'_p > y_p$,
- an inhibitor-arc $(p,t') \in A_I$ with $w(p,t') < y'_p \ \forall p \in P(\boldsymbol{y},\boldsymbol{y'})$ with $y'_p < y_p$.

Remark 1. If one of $\boldsymbol{y},\boldsymbol{y'}$ is a terminal state, say $\boldsymbol{y'}$, then $\mathrm{CA}_{t<t'}(\sigma,\sigma') = \emptyset$ follows since t has to be disabled at $\boldsymbol{y'}$ and $t > t' = \boldsymbol{0}$ holds automatically. Moreover, if $\boldsymbol{y} = \boldsymbol{y'}$ then $P(\boldsymbol{y},\boldsymbol{y'}) = \emptyset$ follows which is the reason why SPCs cannot be resolved by adding control-arcs.

Due to [6], inserting one control-arc from $\mathrm{CA}(\sigma,\sigma')$ resolves the $\mathrm{WPC}(\sigma,\sigma')$ in $\mathcal{P}_S$. Here, we further discuss mutual influences of control-arcs in the resulting extended Petri nets as well as the issue of only constructing minimal catalytic conformal networks.

On the one hand, inserting a control-arc $(p,t) \in \mathrm{CA}(\sigma,\sigma')$ in $\mathcal{P}_S$ might disable t at a state in another sequence $\sigma'' \in S \setminus \sigma,\sigma'$ where t is supposed to switch. In this case, (p,t) has to be removed from $\mathrm{CA}(\sigma,\sigma')$, resulting in a reduced set $\mathrm{CA}_S(\sigma,\sigma')$. On the other hand, one control-arc may resolve several WPCs in $\mathcal{P}_S$ if the corresponding sets $\mathrm{CA}_S(\sigma,\sigma')$ intersect.

Therefore, we propose the following consideration: Introduce one variable $z_{(p,t)} \in \{0,1\}$ for each possible control-arc $(p,t) \in \mathrm{CA}_S(\sigma,\sigma')$ for all WPCs in $\mathcal{P}_S$. Construct a 0/1-matrix A_S whose columns correspond to all those variables (resp. control-arcs) and whose rows encode the incidence vectors of the sets $\mathrm{CA}_S(\sigma,\sigma')$ for all WPCs in $\mathcal{P}_S$. Then any 0/1-solution $\boldsymbol{z}$ of the system $A_S\boldsymbol{z} \geq \boldsymbol{1}$ encodes a suitable set of control-arcs resolving all WPCs in $\mathcal{P}_S$ and, thus, a *hitting set* or *cover* of A_S. By [14], we are only interested in finding minimal models fitting $\mathcal{X}'$, where minimality is interpreted in the sense that all non-minimal models contain another one also fitting the data. We can show that using non-minimal covers of A_S yields non-minimal extended Peri nets but that we need all of them for the sake of completeness, which corresponds to determining the so-called blocker $b(A_S)$ of A_S.

Example 6. We list all WPCs between sequences in our running example:

WPC1 between	$\sigma_1(\boldsymbol{x}^2,\boldsymbol{d}^2)$	and	$\sigma(\boldsymbol{x}^0,\boldsymbol{0})$	due to	$\boldsymbol{d}^2,\boldsymbol{0}$	$\in$	$T(\boldsymbol{x}^0) \cap T(\boldsymbol{x}^2)$
WPC2 between	$\sigma_1(\boldsymbol{x}^3,\boldsymbol{d}^3)$	and	$\sigma(\boldsymbol{x}^0,\boldsymbol{0})$	due to	$\boldsymbol{d}^3,\boldsymbol{0}$	$\in$	$T(\boldsymbol{x}^0) \cap T(\boldsymbol{x}^3)$
WPC3 between	$\sigma_1(\boldsymbol{x}^7,\boldsymbol{d}^3)$	and	$\sigma(\boldsymbol{x}^0,\boldsymbol{0})$	due to	$\boldsymbol{d}^3,\boldsymbol{0}$	$\in$	$T(\boldsymbol{x}^0) \cap T(\boldsymbol{x}^7)$
WPC4 between	$\sigma_3(\boldsymbol{x}^1,\boldsymbol{d}^1)$	and	$\sigma_1(\boldsymbol{x}^7,\boldsymbol{d}^3)$	due to	$\boldsymbol{d}^3,\boldsymbol{r}^{1.2}$	$\in$	$T(\boldsymbol{x}^1) \cap T(\boldsymbol{x}^7)$
WPC5 between	$\sigma_3(\boldsymbol{x}^1,\boldsymbol{d}^1)$	and	$\sigma(\boldsymbol{x}^0,\boldsymbol{0})$	due to	$\boldsymbol{r}^{1.2},\boldsymbol{0}$	$\in$	$T(\boldsymbol{x}^0) \cap T(\boldsymbol{x}^1)$
WPC6 between	$\sigma_3(\boldsymbol{x}^1,\boldsymbol{d}^1)$	and	$\sigma(\boldsymbol{x}^8,\boldsymbol{0})$	due to	$\boldsymbol{r}^{1.2},\boldsymbol{0}$	$\in$	$T(\boldsymbol{x}^8) \cap T(\boldsymbol{x}^1)$
WPC7 between	$\sigma_1(\boldsymbol{x}^2,\boldsymbol{d}^2)$	and	$\sigma_3(\boldsymbol{x}^5,\boldsymbol{d}^4)$	due to	$\boldsymbol{d}^2,\boldsymbol{r}^{4.2}$	$\in$	$T(\boldsymbol{x}^2) \cap T(\boldsymbol{x}^5)$
WPC8 between	$\sigma_3(\boldsymbol{x}^5,\boldsymbol{d}^4)$	and	$\sigma_1(\boldsymbol{x}^3,\boldsymbol{d}^3)$	due to	$\boldsymbol{d}^3,\boldsymbol{r}^{4.2}$	$\in$	$T(\boldsymbol{x}^3) \cap T(\boldsymbol{x}^5)$
WPC9 between	$\sigma_3(\boldsymbol{x}^5,\boldsymbol{d}^4)$	and	$\sigma(\boldsymbol{x}^4,\boldsymbol{0})$	due to	$\boldsymbol{r}^{4.2},\boldsymbol{0}$	$\in$	$T(\boldsymbol{x}^4) \cap T(\boldsymbol{x}^5)$
WPC10 between	$\sigma_3(\boldsymbol{x}^6,\boldsymbol{d}^4)$	and	$\sigma_1(\boldsymbol{x}^3,\boldsymbol{d}^3)$	due to	$\boldsymbol{d}^3,\boldsymbol{r}^{4.2}$	$\in$	$T(\boldsymbol{x}^3) \cap T(\boldsymbol{x}^6)$
WPC11 between	$\sigma_3(\boldsymbol{x}^6,\boldsymbol{d}^4)$	and	$\sigma(\boldsymbol{x}^4,\boldsymbol{0})$	due to	$\boldsymbol{r}^{4.2},\boldsymbol{0}$	$\in$	$T(\boldsymbol{x}^4) \cap T(\boldsymbol{x}^6)$
WPC12 between	$\sigma_1(\boldsymbol{x}^5,\boldsymbol{d}^4)$	and	$\sigma_3(\boldsymbol{x}^6,\boldsymbol{d}^4)$	due to	$\boldsymbol{d}^4,\boldsymbol{r}^{4.2}$	$\in$	$T(\boldsymbol{x}^5) \cap T(\boldsymbol{x}^6)$
WPC13 between	$\sigma_3(\boldsymbol{x}^5,\boldsymbol{d}^4)$	and	$\sigma_1(\boldsymbol{x}^6,\boldsymbol{d}^4)$	due to	$\boldsymbol{d}^4,\boldsymbol{r}^{4.2}$	$\in$	$T(\boldsymbol{x}^5) \cap T(\boldsymbol{x}^6)$

We obtain the following control-arcs to resolve WPCs between sequences:

$$\begin{aligned}
\text{CA(WPC1)} ={}& \{(P_{FR},\boldsymbol{d}^2) \in A_I, (P_R,\boldsymbol{d}^2) \in A_R\},\\
\text{CA(WPC2)} ={}& \{(P_{FR},\boldsymbol{d}^3) \in A_I, (P_R,\boldsymbol{d}^3) \in A_R, (G,\boldsymbol{d}^3) \in A_R\},\\
\text{CA(WPC3)} ={}& \{(G,\boldsymbol{d}^3) \in A_R\},\\
\text{CA(WPC4)} ={}& \{(FR,\boldsymbol{d}^3) \in A_I, (G,\boldsymbol{d}^3) \in A_R, (FR,\boldsymbol{r}^{1.2}) \in A_R, (G,\boldsymbol{r}^{1.2}) \in A_I\},\\
\text{CA(WPC5)} ={}& \{(FR,\boldsymbol{r}^{1.2}) \in A_R\},\\
\text{CA(WPC6)} ={}& \{(Spo,\boldsymbol{r}^{1.2}) \in A_I, (G,\boldsymbol{r}^{1.2}) \in A_I\},\\
\text{CA(WPC7)} ={}& \{(R,\boldsymbol{r}^{4.2}) \in A_R, (R,\boldsymbol{d}^2) \in A_I\},\\
\text{CA(WPC8)} ={}& \{(R,\boldsymbol{r}^{4.2}) \in A_R, (G,\boldsymbol{r}^{4.2}) \in A_I, (R,\boldsymbol{d}^3) \in A_I, (G,\boldsymbol{d}^3) \in A_R\},\\
\text{CA(WPC9)} ={}& \{(R,\boldsymbol{r}^{4.2}) \in A_R, (Spo,\boldsymbol{r}^{4.2}) \in A_I, (G,\boldsymbol{r}^{4.2}) \in A_I\},\\
\text{CA(WPC10)} ={}& \{(R,\boldsymbol{d}^3) \in A_I, (R,\boldsymbol{r}^{4.2}) \in A_R\},\\
\text{CA(WPC11)} ={}& \{(R,\boldsymbol{r}^{4.2}) \in A_R, (Spo,\boldsymbol{r}^{4.2}) \in A_I\},\\
\text{CA(WPC12)} ={}& \{(G,\boldsymbol{d}^4) \in A_I, (G,\boldsymbol{r}^{4.2}) \in A_R\},\\
\text{CA(WPC13)} ={}& \{(G,\boldsymbol{d}^4) \in A_R, (G,\boldsymbol{r}^{4.2}) \in A_I\}.
\end{aligned}$$

The following reductions of sets of possible control-arcs are necessary: for all standard networks $\mathcal{P}_{S_i}$, we have $\sigma_1(\boldsymbol{x}^3,\boldsymbol{d}^3)$ and $\sigma_1(\boldsymbol{x}^7,\boldsymbol{d}^3)$ selected simultaneously, which both are in WPC with $\sigma(\boldsymbol{x}^0,\boldsymbol{0})$, see WPC2 and WPC3. To resolve WPC2, we have $\text{CA(WPC2)} = \{(P_{FR},\boldsymbol{d}^3) \in \mathcal{A}_I, (P_R,\boldsymbol{d}^3) \in \mathcal{A}_R, (G,\boldsymbol{d}^3) \in \mathcal{A}_R\}$.

However, $(P_{FR},\boldsymbol{d}^3) \in \mathcal{A}_I$ and $(P_R,\boldsymbol{d}^3) \in \mathcal{A}_R$ do not only disable $\boldsymbol{d}^3$ at $\boldsymbol{x}^0$, but also $\boldsymbol{d}^3$ at $\boldsymbol{x}^7$ (due to $\boldsymbol{x}^0_{P_{FR}} = \boldsymbol{x}^7_{P_{FR}} = 1$ and $\boldsymbol{x}^0_{P_R} = \boldsymbol{x}^7_{P_R} = 0$). Since $\boldsymbol{d}^3$ is supposed to switch at $\boldsymbol{x}^7$, we obtain $\text{CA}_{S_i}(\text{WPC2}) = \{(G,\boldsymbol{d}^3) \in \mathcal{A}_R\}$ as reduced set of possible control-arcs to resolve WPC2 in all networks $\mathcal{P}_{(S_i)}$. Similarly, $(G,\boldsymbol{r}^{4.2}) \in \mathcal{A}_I$ has to be excluded from CA(WPC8) and CA(WPC9) in $\mathcal{P}_{S_4}$ and $\mathcal{P}_{S_8}$ as otherwise $\boldsymbol{r}^{4.2}$ would be disabled at $\boldsymbol{x}^6$ where it is supposed to switch by $\sigma_3(\boldsymbol{x}^6,\boldsymbol{d}^4) \in S_4, S_8$.

For the feasible set S_4, we obtain as matrix A_{S_4}:

	$(\mathrm{P}_{FR},\boldsymbol{d}^2) \in A_I$	$(\mathrm{P}_R,\boldsymbol{d}^3) \in A_R$	$(\mathrm{G},\boldsymbol{d}^3) \in A_R$	$(\mathrm{R},\boldsymbol{r}^{4.2}) \in A_R$	$(\mathrm{R},\boldsymbol{d}^2) \in A_I$	$(\mathrm{R},\boldsymbol{d}^3) \in A_I$	$(\mathrm{Spo},\boldsymbol{r}^{4.2}) \in A_I$
WPC1	*X*	*X*					
WPC2			*X*				
WPC3			*X*				
WPC7				*X*	*X*		
WPC8			*X*	*X*		*X*	
WPC9				*X*			*X*
WPC10				*X*		*X*	
WPC11				*X*			*X*

The blocker $b(A_{S_4})$ contains four minimal covers of A_{S_4} which correspond to the different sets of control-arcs in the four extended Petri nets depicted in Fig. 3, all arising from the standard network $\mathcal{P}_{S_4}$.

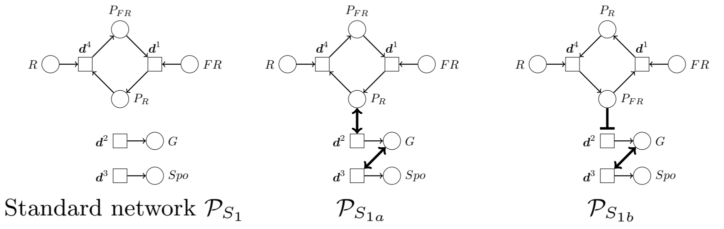
For S_1, the matrix A_{S_1} contains the first 3 rows and columns of A_{S_4}, the blocker $b(A_{S_1})$ contains two minimal covers of A_{S_1} which correspond to the control-arcs in the two extended Petri nets in Fig. 2 arising from $\mathcal{P}_{S_1}$.

Note that $b(A_S)$ is non-empty if and only if none of the sets $\text{CA}_S(\sigma,\sigma')$ is empty. We can show that there is at least one catalytic conformal network for any given $\mathcal{X}'$. All catalytic conformal extended Petri nets $\mathcal{P}_{S,P'} = (P, T_S, \mathcal{A}_{S,P'}, w)$ based on $\mathcal{P}_S$ can be made $\mathcal{X}'$-deterministic by taking all the priorities $\mathcal{O}_\sigma$ for all $\sigma \in S$, where $\mathcal{O}_\sigma$ is defined by $\mathcal{O}_\sigma = \left\{t_i > t : t \in T_{\mathcal{A}_{S,P'}}(\boldsymbol{y}^i) \setminus t_i, 1 \leq i \leq m\right\}$. By construction, there are no priority conflicts in the extended network $\mathcal{P}_{S,P'}$ between $\mathcal{O}_\sigma$ and $\mathcal{O}_{\sigma'}$ for any $\sigma,\sigma' \in S$, thus we obtain the studied partial order

$$\mathcal{O}_{S,P'} = \bigcup_{\sigma \in S} \mathcal{O}_\sigma.$$

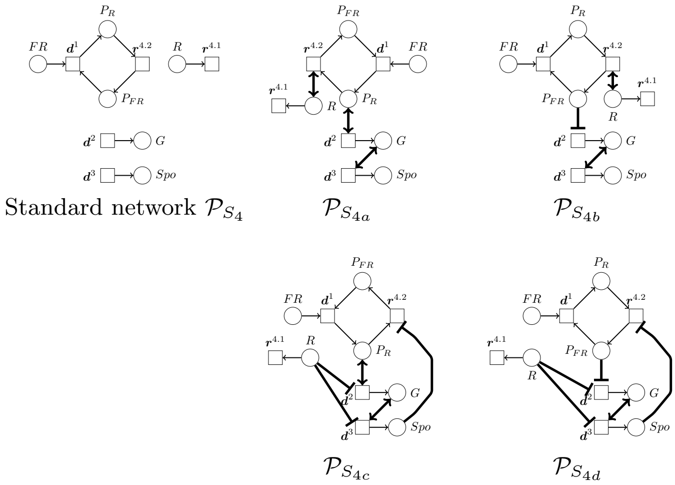
This finally implies:

Theorem 1. *Every extended network $\mathcal{P}_{S,P'} = (P, T_S, \mathcal{A}_{S,P'}, w)$ together with the partial order $\mathcal{O}_{S,P'}$ is an $\mathcal{X}'$-deterministic extended Petri net, and there are no other minimal extended Petri nets with priorities fitting the given data $\mathcal{X}'$.*



Standard network $\mathcal{P}_{S_1}$ $\mathcal{P}_{S_{1a}}$ $\mathcal{P}_{S_{1b}}$

Fig. 2. Standard network $\mathcal{P}_{S_1} = (P, T_{S_1}, A_{S_1}, w)$ from solution S_1 and the two catalytic conformal extended Petri nets resulting from $\mathcal{P}_{S_1}$.



Standard network $\mathcal{P}_{S_4}$ $\mathcal{P}_{S_{4a}}$ $\mathcal{P}_{S_{4b}}$

$\mathcal{P}_{S_{4c}}$ $\mathcal{P}_{S_{4d}}$

Fig. 3. Standard network $\mathcal{P}_{S_4} = (P, T_{S_4}, A_{S_4}, w)$ from solution S_4 and the four catalytic conformal extended Petri nets resulting from $\mathcal{P}_{S_4}$.

Example 7. For the two extended Petri nets based on $\mathcal{P}_{S_1}$ in Fig. 2, no priorities are needed to obtain $\mathcal{X}'$-deterministic extended Petri nets $(\mathcal{P}_{S_{1a}}, \emptyset)$ and $(\mathcal{P}_{S_{1b}}, \emptyset)$.

For the four extended Petri nets in Fig. 3 based on $\mathcal{P}_{S_4}$, the priority relation $\mathcal{O}_4 = \{(\boldsymbol{r}^{4.2} > \boldsymbol{d}^2)\}$ is required for $\mathcal{P}_{S_{4_a}}$ and $\mathcal{P}_{S_{4_b}}$, whereas $\mathcal{O}_4 = \{(\boldsymbol{d}^2 > \boldsymbol{r}^{4.2}), (\boldsymbol{d}^3 > \boldsymbol{r}^{4.2})\}$ is required for $\mathcal{P}_{S_{4_c}}$ and $\mathcal{P}_{S_{4_d}}$, to obtain $\mathcal{X}'$-deterministic extended Petri nets.

In total, the complete solution set contains 66 minimal $\mathcal{X}'$-deterministic extended Petri nets with priorities, see Table 1.

3 Integrating prior biological knowledge

3.1 Indecomposability of difference vectors

In the step *Representing the observed responses*, the set of potential update vectors which might constitute the incidence matrices of the networks are considered. Hereby, for each $\boldsymbol{d}^j \in \mathcal{D}$, the set $\mathrm{Box}(\boldsymbol{d}^j)$ contains only sign-compatible vectors due to monotonicity (which avoids homogeneous solutions in (1) according to minimality) and takes P-invariants into account (which avoids infeasible intermediate states according to prior biological knowledge). In some cases, one could restrict $\mathrm{Box}(\boldsymbol{d}^j)$ further, e.g., if

- $\boldsymbol{d}^j$ exactly corresponds to a well-known biochemical reaction (including the correct stoichiometry) or to a well-known mechanism (that a certain trigger is detected by a suitable receptor),
- experiments have shown that subsets of the input components of $\boldsymbol{d}^j$ do not lead to the observed response,
- $\boldsymbol{d}^j$ is treated as black box-like reaction where only input and output matter, but not the intermediate mechanism due to the chosen level of detail.

In such cases, we propose to exclude the corresponding response $\boldsymbol{d}^j \in \mathcal{D}$ from decomposition and, instead, just define $\mathrm{Box}(\boldsymbol{d}^j) := \{\boldsymbol{d}^j\}$ in accordance with the existing knowledge.

Example 8. Since the light-dependent reactions of the photoreceptor are so much faster than the subsequent processes that are considered in the reconstruction process, the difference vector describing the phytochrome photoconversion will not be decomposed into different reaction vectors.

Thus, for the difference vectors $\boldsymbol{d}^1$, $\boldsymbol{d}^4$ and $\boldsymbol{d}^5$ only the canonical sequences $\sigma_1(\boldsymbol{x}^1, \boldsymbol{d}^1)$, $\sigma_1(\boldsymbol{x}^5, \boldsymbol{d}^4)$ and $\sigma_1(\boldsymbol{x}^6, \boldsymbol{d}^4)$ remain in the priority conflict graph, and S_1 remains as only possible selection. Accordingly, the total number of solutions reduces from 66 to the 2 presented in Fig. 2, see Table 1.

3.2 Treating equal difference vectors in the same way

In the step *Detecting priority conflicts between sequences*, all observed responses $\boldsymbol{d}^j \in \mathcal{D}$ are treated independently from each other, so far. If, however, two difference vectors $\boldsymbol{d}^i, \boldsymbol{d}^j \in \mathcal{D}$ are equal, we clearly have $\mathrm{Box}(\boldsymbol{d}^i) = \mathrm{Box}(\boldsymbol{d}^j)$ and, thus, $\Lambda(\boldsymbol{d}^i) = \Lambda(\boldsymbol{d}^j)$. Here, it is natural to require that both $\boldsymbol{d}^i, \boldsymbol{d}^j$ are decomposed in the same way (i.e., by the same $\lambda \in \Lambda(\boldsymbol{d}^i) = \Lambda(\boldsymbol{d}^j)$) and that the involved reactions are applied in the same order (i.e., by the same permutation π of the elements in the sets $\mathcal{R}(\boldsymbol{d}^i, \lambda) = \mathcal{R}(\boldsymbol{d}^j, \lambda)$) to obtain the same transitions (with equal control-arcs and priorities) in both cases. Indeed, using the same

- $\lambda \in \Lambda(\boldsymbol{d}^i) = \Lambda(\boldsymbol{d}^j)$ corresponds to the fact that the same subset of molecules involved in a reaction will not interact according to different mechanisms,

- permutation π of the elements in $\mathcal{R}(\boldsymbol{d}^i, \lambda) = \mathcal{R}(\boldsymbol{d}^j, \lambda)$ corresponds to the fact that the order in which the reactions are applied reflects the relative rates of the reactions in $\mathcal{R}(\boldsymbol{d}^i, \lambda) = \mathcal{R}(\boldsymbol{d}^j, \lambda)$, so the same relation between reaction rates shall lead to the same priorities within the resulting sequences.

We call two sequences $\sigma_{\pi,\boldsymbol{\lambda}}(\boldsymbol{x}^i, \boldsymbol{d}^i)$ and $\sigma'_{\pi,\boldsymbol{\lambda}}(\boldsymbol{x}^j, \boldsymbol{d}^j)$ *twin sequences* if $\boldsymbol{d}^i = \boldsymbol{d}^j$ and the same $\boldsymbol{\lambda}$ and π has been used. To force that twin sequences are always selected together, we propose to add strong priority conflicts between all other sequences stemming from a pair $\boldsymbol{d}^i, \boldsymbol{d}^j$ of equal difference vectors while creating the priority conflict graph, since no two sequences in strong priority conflict are selected for the same network.

Example 9. In our running example, we have $\boldsymbol{d}^3 = \boldsymbol{d}^6$ and $\boldsymbol{d}^4 = \boldsymbol{d}^5$. The latter vectors can be decomposed in different ways, among the resulting sequences we have $\sigma_1(\boldsymbol{x}^5, \boldsymbol{d}^4)$, $\sigma_1(\boldsymbol{x}^6, \boldsymbol{d}^5)$ and $\sigma_3(\boldsymbol{x}^5, \boldsymbol{d}^4)$, $\sigma_3(\boldsymbol{x}^6, \boldsymbol{d}^5)$ as pairs of twin sequences. Forcing to select these pairs together rules out the four selections S_2, S_3, S_6, S_7 so that only S_1, S_4, S_5, S_8 remain as possible selections. Accordingly, the total number of solutions reduces from 66 to 18, as reported in Table 1.

3.3 Knowledge on relative reaction rates

In the step *Constructing $\mathcal{X}'$-deterministic Petri nets*, to resolve a WPC(σ, σ') between σ, σ' involving update vectors $\boldsymbol{r}^t \neq \boldsymbol{r}^{t'}$ and intermediate states $\boldsymbol{y} \neq \boldsymbol{y}'$, either transition t has to be disabled at $\boldsymbol{y}'$ or transition t' at $\boldsymbol{y}$, while the decision between t and t' on the other state can be handled by a priority. Here, prior knowledge about the relative reaction rates of t and t' (e.g. gained from the time-scales during the experiments) could help to decide whether $t > t'$ or $t < t'$ better reflects the reality, and than choose between control-arcs either from $\mathrm{CA}_{t>t'}(\sigma, \sigma')$ or from $\mathrm{CA}_{t'>t}(\sigma, \sigma')$.

So far, this idea is already applied for WPCs involving a terminal state: if $\boldsymbol{y}'$ is a terminal state and $\sigma' = \sigma(\boldsymbol{y}', \boldsymbol{0})$ its trivial sequence, then $t > \boldsymbol{0}$ holds automatically at $\boldsymbol{y}$, but t has to be disabled at $\boldsymbol{y}'$ using control-arcs from $\mathrm{CA}_{t>\boldsymbol{0}}(\sigma, \sigma')$ whereas $\mathrm{CA}_{t<\boldsymbol{0}}(\sigma, \sigma')$ is empty.

This idea can be generalized to any WPC(σ, σ') where the time-scale of the corresponding experimental observations clearly differs in order to deduce the correct priority $t > t'$ or $t < t'$. In such cases, we propose to reduce $\mathrm{CA}(\sigma, \sigma')$ accordingly either to $\mathrm{CA}_{t>t'}(\sigma, \sigma')$ or to $\mathrm{CA}_{t'>t}(\sigma, \sigma')$.

Example 10. In our running example, we have WPC1, WPC2, WPC3, WPC5, WPC6, WPC9, WPC11 involving a terminal state; the according reductions of the sets $\mathrm{CA}(\sigma, \sigma')$ to $\mathrm{CA}_{t>\boldsymbol{0}}(\sigma, \sigma')$ are already applied in Exp. 6.

Moreover, WPC4, WPC7, WPC8, WPC10 involve reactions with clearly different time-scales during the experimental observations:

- $\boldsymbol{d}^1$ and $\boldsymbol{d}^4 = \boldsymbol{d}^5$ need only milliseconds to occur,
- $\boldsymbol{d}^2$ needs about 1 hour to occur, and
- $\boldsymbol{d}^3 = \boldsymbol{d}^6$ need at least 10 hours.

Accordingly, we can reduce the sets $\mathrm{CA}(\sigma, \sigma')$ as follows:

– due to $\boldsymbol{r}^{1.2} > \boldsymbol{d}^3$, for WPC4 only $(FR, \boldsymbol{r}^{1.2}) \in A_R$ and $(G, \boldsymbol{r}^{1.2}) \in A_I$ remain;
– due to $\boldsymbol{r}^{4.2} > \boldsymbol{d}^2$, for WPC7 only $(R, \boldsymbol{r}^{4.2}) \in A_R$ remains;
– due to $\boldsymbol{r}^{4.2} > \boldsymbol{d}^3$, for WPC8 only $(R, \boldsymbol{r}^{4.2}) \in A_R$ and $(G, \boldsymbol{r}^{4.2}) \in A_I$ remain, while for WPC10 only $(R, \boldsymbol{r}^{4.2}) \in A_R$ is left.

At least one of these WPCs occurs in the standard networks coming from the selected sets $S_2 - S_4, S_6 - S_8$. Note that in the two remaining WPCs the time-scale of the involved responses is equal (see Exp. 6). Consequently, the number of extended Petri nets decreases from 66 to 36 as reported in Table 1.

4 Discussion

The subject of this paper was an approach from [6] that aims at reconstructing all $\mathcal{X}'$-deterministic extended Petri nets that fit given experimental data $\mathcal{X}'$, to provide all possible alternatives of mechanisms behind the experimentally observed phenomena. This typically results in a large set of solution alternatives. To keep this solution set reasonably small while still guaranteeing its completeness, we firstly generate only Petri nets being minimal in the sense that all other networks fitting the data contain the reconstructed ones. In the presented approach, the minimality concept is applied twice:

– *monotone data*: using only sign-compatible vectors in $\mathrm{Box}(\boldsymbol{d}^j)$ avoids homogeneous solutions during the decomposition of $(\boldsymbol{d}^j)$ (and superflous transitions in the networks), see [4].
– *minimal hitting sets*: we here proposed to use only minimal sets of control-arcs to resolve all weak priority conflicts in a standard network which avoids unnecessary control-arcs (and artificial dependencies), see Section 2.

This ensures that the presented approach exactly generates all minimal extended Petri nets with priorities (Theorem 1). We further avoid generating minimal solutions which are "technically correct" but would be ruled out later during a subsequent verification process to check whether the returned solutions are "biological meaningful" or even contradict well-established biological knowledge as in [2]. For that, we extent the considered input by integrating prior biological knowledge beyond the information given with the experimental time-series data into the reconstruction process. We propose to integrate prior knowledge in the following way:

– *P-invariants*: helps to obtain feasible intermediate states in all sequences which avoids the generation of solutions contradicting known facts, see [13].
– *indecomposable difference vectors*: help to keep already known subnetworks or mechanisms, see Section 3.1.
– *treating equal difference vectors in the same way*: helps to keep consistency in the interpretation of experimental observations, see Section 3.2.
– *obeying terminal states and reaction rates*: helps to chose meaningful priorities and to avoid artificial control-arcs, see Section 3.3.

So far, the algorithmic procedure due to [6] takes as input $(P, I_P, \mathcal{X}')$ where the list of P-invariants and the monotonicity of $\mathcal{X}'$ are already used in the first step *Decomposing difference vectors* to determine $\text{Box}(\boldsymbol{d}^j)$. To integrate further knowledge in the reconstruction procedure to force the algorithm to make "the right decisions" in some intermediate steps, we suggest, based on the previous discussions, to extend the input to $(P, I_P, \mathcal{X}', \mathcal{D}_{in}, \mathcal{D}_{eq}, \mathcal{O}_D)$ where

- $\mathcal{D}_{in}$ contains all indecomposable difference vectors (for that, carefully select them according to the before mentioned critera, e.g., to preserve known subnetworks or mechanisms or to take a certain level of detail into account);
- $\mathcal{D}_{eq}$ contains all pairs of equal difference vectors that shall be treated in the same way (for that, only chose equal difference vectors where also the time elapsed during the experimental observation was equal);
- $\mathcal{O}_D$ contains information about the reaction rates between difference vectors (for that, only impose priorities for sufficiently different rates according to clearly different time scales during the experiments).

Example 11. The three examples from Section 3 can be interpreted as follows:

- Exp. 8 shows the result taking $(P, I_P, \mathcal{X}', \mathcal{D}_{in} = \{\boldsymbol{d}^1, \boldsymbol{d}^4, \boldsymbol{d}^5\}, \emptyset, \emptyset)$ as input;
- Exp. 9 shows the result with $(P, I_P, \mathcal{X}', \emptyset, \mathcal{D}_{eq} = \{\boldsymbol{d}^3 = \boldsymbol{d}^6, \boldsymbol{d}^4 = \boldsymbol{d}^5\}, \emptyset)$;
- Exp. 10 the result with $(P, I_P, \mathcal{X}', \emptyset, \emptyset, \mathcal{O}_D = \{\boldsymbol{d}^1, \boldsymbol{d}^4, \boldsymbol{d}^5 > \boldsymbol{d}^2 > \boldsymbol{d}^3, \boldsymbol{d}^6\})$.

Whereas the first setting reduces the number of solution alternatives to 2, the combination of the two latter scenarios reduces the number of solution alternatives to 12, see Table 1 below.

	$S1$	$S2$	$S3$	$S4$	$S5$	$S6$	$S7$	$S8$	TOTAL
minimal	2	8	8	4	4	16	16	8	66
$\mathcal{D}_{in} \neq \emptyset$	2	0	0	0	0	0	0	0	2
$\mathcal{D}_{eq} \neq \emptyset$	2	0	0	4	4	0	0	8	18
$\mathcal{O}_D \neq \emptyset$	2	4	4	2	4	8	8	4	36
$\mathcal{D}_{eq}, \mathcal{O}_D \neq \emptyset$	2	0	0	2	4	0	0	4	12

Table 1. Number of solutions depending on different input settings.

To conclude, we notice that providing indecomposable difference vectors has the largest impact on the solution set. However, even if no indecomposable difference vectors can be identified, treating equal difference vectors in the same way and deducing relative reaction rates from the time-scale of the experimental observations leads to a substantial reduction of the solution set, keeping only "biological meaningful" network alternatives.

The further goal is to provide an implementation for the presented reconstruction method, including the option of integrating prior knowledge as additional input during the reconstruction. For that, we will use Answer Set Programming as done for the reconstruction of standard networks in [1]. Finally, we plan to apply the presented reconstruction approach to different biological experimental data. We expect an important impact of Automatic Network Reconstruction on the integrated experimental and theoretical analysis of biological systems towards their holistic understanding, since computational models derived from experimental data by our exact, exclusively data-driven approach have predictive ability due to completeness of the solution set guaranteed by mathematical proofs.

References

1. M. Durzinsky, W. Marwan, M. Ostrowski, T. Schaub, and A. K. Wagler. Automatic network reconstruction using ASP. *Th. and Pract. of Logic Progr.*, 11:749–66, 2011.
2. M. Durzinsky, W. Marwan, and A. K. Wagler. Reconstruction of extended Petri nets from time series data and its application to signal transduction and to gene regulatory networks. *BMC Systems Biology*, 5, 2011.
3. M. Durzinsky, W. Marwan, and A. K. Wagler. Reconstruction of extended Petri nets from time-series data by using logical control functions. *Journal of Mathematical Biology*, 66:203–223, 2013.
4. M. Durzinsky, A. K. Wagler, and R. Weismantel. A combinatorial approach to reconstruct Petri nets from experimental data. *Lecture Notes in Comp. Sc. (special Issue CMSP 2008)*, 5307:328–346, 2008.
5. M. Durzinsky, A. K. Wagler, and R. Weismantel. An algorithmic framework for network reconstruction. *J.Theor. Computer Science*, 412(26):2800–2815, 2011.
6. M. Favre and A. K. Wagler. Reconstructing $\mathcal{X}'$-deterministic extended Petri nets from experimental time-series data $\mathcal{X}'$ (extended abstract). *CEUR Workshop Proceedings (Special Issue BioPPN 2013)*, 988:45–59, 2013.
7. I. Koch and M. Heiner. Petri nets. In B. H. Junker, F. Schreiber, editor, *Biological Network Analysis*, Wiley Book Series on Bioinformatics, pages 139–179, 2007.
8. T. Lamparter and W. Marwan. Spectroscopic detection of a phytochrome-like photoreceptor in the myxomycete physarum polycephalum and the kinetic mechanism for the photocontrol of sporulation. *Photochem Photobiol*, 73:697–702, 2001.
9. W. Marwan, A. K. Wagler, and R. Weismantel. A mathematical approach to solve the network reconstruction problem. *Math. Methods of Operations Research*, 67(1):117–132, 2008.
10. J. W. Pinney, R. D. Westhead, and G. A. McConkey. Petri net representations in systems biology. *Biochem. Soc. Tarns.*, 31:1513–1515, 2003.
11. C. Starostzik and W. Marwan. Functional mapping of the branched signal transduction pathway that controls sporulation in *Physarum polycephalum. Photochem. Photobiol.*, 62:930–933, 1995.
12. L. M. Torres and A. K. Wagler. Encoding the dynamics of deterministic systems. *Math. Methods of Operations Research*, 73:281–300, 2011.
13. A. K. Wagler. Prediction of network structure. In I. Koch, F. Schreiber, and W. Reisig, editors, *Modeling in Systems Biology*, volume 16 of *Computational Biology*, pages 309–338, 2010.
14. A. K. Wagler and J.-T. Wegener. On minimality and equivalence of Petri nets (extended abstract). *Fundamenta Informaticae*, 128:209–222, 2013.
15. A. K. Wagler and J.-T. Wegener. Preprocessing for network reconstruction: Feasibility test and handling infeasibilty (extended abstract). *CEUR Workshop Proceedings (Special Issue CS&P 2013)*, (1032):434–47, 2013.

Coloured hybrid Petri nets for systems biology

Mostafa Herajy[1]*, Fei Liu[2]* and Christian Rohr[3]*

[1] Department of Mathematics and Computer Science, Faculty of Science,
Port Said University, 42521 - Port Said, Egypt
[2] Control and Simulation Center, Harbin Institute of Technology,
Postbox 3006, 150080, Harbin, China
[3] Computer Science Institute, Brandenburg University of Technology
Postbox 10 13 44, 03013 Cottbus, Germany

Abstract. Coloured Petri nets are imperative for studying bigger biological models, particularly, those which expose repetition of components. Such models can be easily scaled by minor changes of a few parameters. Similarly, studying certain biological phenomena necessitates the existence of discrete and continuous variables as well as continuous and stochastic processes in one and the same model. Thus, hybrid modelling and simulation is indispensable to deal with these challenges. In this paper we introduce a generic Petri net class, Coloured Generalised Hybrid Petri Nets ($\mathcal{GHPN}^{\mathcal{C}}$) by combining coloured Petri nets and Generalised Hybrid Petri Nets ($\mathcal{GHPN}$), which integrates discrete and continuous places as well as stochastic and deterministic transitions on the coloured level. Moreover, we present a case study which illustrates a possible and typical application where such a Petri net class is highly required.

Keywords: coloured Petri nets; hybrid Petri nets; stochastic and continuous simulation

1 Introduction

Petri nets have been widely used for modelling and analysis of biological systems, see e.g.,[2,10,18,21,31,33,34,35,36,39,40]. Their intuitive graphical representation makes them easily approachable by biologists. Furthermore, Petri nets possess well-established mathematical notations which may render them as the future de facto standard for modelling biological systems compared to other graphical languages currently in use. Nevertheless, with the rapid progress of systems biology, standard place/transition Petri nets become insufficient to accommodate the needs of systems biologists to study larger models. Therefore, many different extensions have been adapted for their potential contribution to systems biology. Among such promising extensions are hybrid Petri nets ($\mathcal{HPN}$) [1,6] and coloured Petri nets ($\mathcal{PN}^{\mathcal{C}}$) [23].

[*] corresponding authors

M. Heiner (Ed.): BioPPN 2014, a satellite event of PETRI NETS 2014,
CEUR Workshop Proceedings Vol. 1159, 2014.

Hybrid Petri nets [1,6] have been increasingly motivated for their contribution to systems biology [17,18,35,36]. On the one hand (biological) systems may occur at different time scales: slow and fast [14,25]. Using one modelling paradigm (i.e., discrete or continuous paradigm alone) tends to be inefficient for studying multi-timescale biological models [18]. On the other hand the integration of discrete and continuous variables as well as deterministic and stochastic processes in the same model is necessary in certain application scenarios to implement particular model semantics [18,36]. For instance, in [21] discrete places and immediate transitions were employed to implement the part related to cell division of the eukaryotic cell cycle while continuous and stochastic transitions were used to represent and quantitatively simulate biological reactions. Additionally, the $\mathcal{HPN}$ formalism provides a different approach to control the accuracy and the speed of biological models during simulation [18]. Thus, they provide the modeller with a tool to make a trade-off between the simulation's efficiency and the result's accuracy.

Another, yet powerful extension of standard Petri nets is coloured Petri nets ($\mathcal{PN}^{\mathcal{C}}$) [23,31,40], where a group of similar components are represented by one component, each of which is defined as and thus distinguished by a colour [27]. $\mathcal{PN}^{\mathcal{C}}$ are very useful for modelling larger biological systems where low-level Petri nets do not scale well, while a (biological) system modelled via coloured Petri nets can easily be scaled with minor modifications of certain coloured variables. Furthermore, modelling of biological systems is moving from single to multiple scales (multi-scale modelling), which introduces a series of challenges such as repetition of components, communication between components, organisation of components, and pattern formation of components [27]. All these challenges potentially could be tackled by coloured Petri nets rather than standard Petri nets.

Nevertheless, with the rapid change of type and size of models of biological systems which have to be analysed, a Petri net class that integrates all the features of hybrid Petri nets and coloured Petri nets is highly required. The high-level representation of coloured Petri nets can be used to model larger biochemical networks and stochastic and continuous components can be simultaneously used to facilitate the efficient simulation of multi-timescale models.

Thus, in this paper we introduce a class of Petri nets, Coloured Generalised Hybrid Petri Nets ($\mathcal{GHPN}^{\mathcal{C}}$) by combining all features of Generalised Hybrid Petri Nets ($\mathcal{GHPN}$) [18] and the merits of Coloured Petri Nets ($\mathcal{PN}^{\mathcal{C}}$) [27]. The new class supports a rich set of transition types as well as discrete and continuous places. Moreover, it permits the full interplay between stochastic and deterministic processes at the coloured level. All the feature of $\mathcal{GHPN}^{\mathcal{C}}$ are implemented in Snoopy [15] – a unifying Petri net tool which is available free of charge for academic use.

The paper is structured as follows: first we briefly discuss the related work of hybrid Petri nets and coloured Petri nets followed by a brief background of each of those net types. Next, we present the formal definition of the new Petri net class followed by its semantic as well as an outline of the simulation algorithm used to produce the dynamic of $\mathcal{GHPN}^{\mathcal{C}}$. Afterwards, we present one possible and typical application of $\mathcal{GHPN}^{\mathcal{C}}$ in the context of systems biology, the repressilator model, which is an engineered synthetic system encoded on a plasmid. We conclude by summarizing possible extensions of Coloured Generalised Hybrid Petri Nets.

2 Related Work

In this section, we will briefly describe related work on hybrid and coloured Petri nets for systems biology.

2.1 Hybrid Petri Nets for Systems Biology

Hybrid Petri nets have been introduced in [1] to deal with situations where discrete and continuous entities exist in the same model. The motivation given in early publications was in modelling technical systems whereby discrete places are used to model the states of switch-like components while continuous places are used to represent their fluid counterparts. The pioneer work of introducing hybrid Petri nets to systems biology was done by Matsuno et al., in [36]. They had noticed that using equal inflow and outflow in the fluid part of hybrid Petri nets is not natural to use them to model biological processes. Thus, they introduced Hybrid Functional Petri Nets (HFPN) [36,35]. Many successful models have been implemented using HFPN (e.g., see [8,34]).

In [18], we have used Hybrid Petri nets in a different way whereby stochastic transitions are used to model slow biological processes, while fast processes are modelled via continuous transitions. In [21], this approach is used to model the eukaryotic cell cycle.

2.2 Coloured Petri Nets for Systems Biology

The early applications of coloured Petri nets for systems biology were limited, which, to our knowledge, can almost only be seen in [2,3,5,11,26,38,40,41,42]. These studies are rather small and usually resort to `Design/CPN` [4] or its successor `CPN Tools` [23]. However these tools were not specifically designed with the requirements of systems biology in mind. Thus they are not suitable in many aspects, e.g., they do not directly support stochastic or continuous modelling, nor stochastic or deterministic simulation.

Since coloured Petri nets were introduced to our Petri net modelling tool, Snoopy, we have extensively explored the application of coloured Petri nets for (multiscale) systems biology. For example, in [29], we used coloured stochastic Petri nets to model and analyse stochastic membrane systems, where each compartment is encoded as a colour. In [30], we described the multiscale modelling of coupled Ca^{2+} channels using coloured stochastic Petri nets by considering two levels: Ca^{2+} release sites and Ca^{2+} channels. In [10], Coloured stochastic and coloured continuous Petri nets are used for multiscale modelling and analysis of Planar Cell Polarity in the Drosophila wing, and the built model consists of more than 800 cells. In [37], a case study of phase variation in bacterial colony growth is given, in which cells are distributed on a two-dimensional grid represented by both Cartesian and polar coordinate systems.

3 Background

3.1 Generalised Hybrid Petri Nets

In [18] we have introduced a special class of Hybrid Petri nets called Generalised Hybrid Petri Nets ($\mathcal{GHPN}$). The main two objectives of $\mathcal{GHPN}$ are: to provide systems biologists with a convenient and flexible graphical tool to model and simulate biological processes with different time scales and to facilitate the process of constructing hybrid models where stochastic and deterministic (i.e, continuous) processes are tightly coupled. To fulfil such objectives, $\mathcal{GHPN}$ contain a rich set of element types: places (discrete, and continuous), transitions (stochastic, immediate, deterministic, stochastic, and continuous), and arcs (standard, read, inhibitor, equal, and reset) [18,17]. Figure 1 depicts the graphical representation of the different element types of $\mathcal{GHPN}$.

Discrete places (single line circle) hold non-negative integer numbers which may represent the number of molecules of a given species (tokens in Petri net notions). On the other hand, continuous places - which are represented by the shaded line circle - hold non-negative real numbers which represent the concentration of a certain species. Please note that, except when otherwise mentioned, the number which a place p_i holds, also called its marking, is referred to by $m(p_i)$.

Furthermore, $\mathcal{GHPN}$ offer five transition types: stochastic, immediate, deterministically delayed, scheduled, and continuous transitions [18]. Stochastic transitions, which are drawn in Snoopy as a single line square, fire randomly with an exponentially distributed random delay. The user can specify a set of firing rate functions that determine the random firing delay. The transitions' pre-places can be used to define the firing rate functions of stochastic transitions. Immediate transitions (black bar) fire with zero delay, and have always highest priority in the case of conflicts with other transitions. They may carry weights (which can also be defined by state-dependent functions) that specify the relative firing frequency in the case of conflicts between immediate transitions. Deterministically delayed transitions (represented as black squares) fire after a specified constant time delay. Scheduled transitions (grey squares) fire at user-specified absolute time points. Continuous transitions (shaded line square) fire continuously in the same way as in continuous Petri nets. Their semantics are governed by a set of ordinary differential equations (ODEs) which define the changes in the transitions' pre- and post-places. More details about the biochemical interpretation of deterministically delayed, scheduled, and immediate transitions can be found in [16].

The connection between those two types of nodes (places and transitions) takes place using a set of different arcs. $\mathcal{GHPN}$ offer six types of arcs: standard, inhibitor, read, equal, reset, and modifier arcs. Standard arcs connect transitions with places or vice versa. They can be discrete, i.e., carry non-negative integer-valued weights (stoichiometry in the biochemical context), or continuous, i.e.,

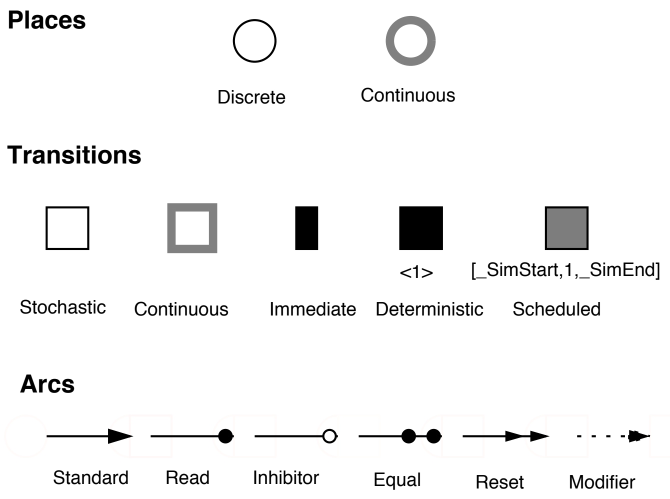


Fig. 1. Graphical representation of the $\mathcal{GHPN}$ elements [18]. Places are classified as discrete and continuous; transitions as continuous, stochastic, immediate, deterministically delayed and scheduled; and arcs as standard, inhibitor, read, equal, reset, and modifier.

carry non-negative real-valued weights. In addition to their influence on the enabling of transitions, they also affect the place marking when a transition fires by adding (removing) tokens from the transition's post-places (pre-places).

Extended arcs such as inhibitor, read, equal, reset, and modifier arcs can only be used to connect places to transitions, and not vice versa. A transition connected with an inhibitor arc is enabled (with respect to the corresponding pre-place) if the marking of the pre-place is less than the arc weight. In contrast, a transition connected with a read arc is enabled if the marking of the pre-place is greater than or equal to the arc weight. Similarly, a transition connected using an equal arc is enabled if the marking of the pre-place is equal to the arc weight.

The other two remaining arcs do not affect the enabling of transitions. A reset arc is used to reset a place marking to zero when the corresponding transition fires. Modifier arcs permit one to include any place in the transitions' rate functions and simultaneously preserve the net structure restriction. Besides, the markings of places connected using read, inhibitor, equal, or modifier arcs does not change when the corresponding transition fires.

3.2 Coloured Petri nets

Coloured Petri nets [22] consist, as standard Petri nets, of places, transitions and arcs that connect places and transitions. Additionally, a coloured Petri net is characterised by a set of finite colour sets. Each place gets assigned a colour set and may contain distinguishable tokens; each token has a colour of this colour set.

As there can be several tokens of the same colour on a given place, the tokens on a place define a multiset over the place's colour set.

Each transition gets a guard, which is a Boolean expression over variables, constants, functions, etc. The guard of a transition must be evaluated to true for the enabling of the transition. The trivial guard "true" is usually not explicitly given.

Each arc gets assigned an expression, rather than an integer number in standard Petri nets; the result type of this expression is a multiset over the colour set of the connected place.

4 Coloured Generalised Hybrid Petri Nets

In this section we present the definition of coloured generalised hybrid Petri nets. $\mathcal{GHPN}^{\mathcal{C}}$ reuse all elements of $\mathcal{GHPN}$, however on the coloured level. We start off by defining $\mathcal{GHPN}^{\mathcal{C}}$. Afterwards, we present the semantics of $\mathcal{GHPN}^{\mathcal{C}}$ that governs the execution of models constructed by it.

4.1 Formal Definition

Definition 1 *A Coloured Generalised Hybrid Petri Net* $\mathcal{GHPN}^{\mathcal{C}}$ *is a nine-tuple* $N =< P, T, A, \sum, C, F, V, G, m_0 >$*[27,18], where:*

- $P = P_{disc} \cup P_{cont}$ *whereby* P_{disc} *is the set of discrete places and* P_{cont} *is the set of continuous places.*
- $T = T_{stoch} \cup T_{im} \cup T_{timed} \cup T_{scheduled} \cup T_{cont}$ *with:*
  1. T_{stoch} *is the set of stochastic transitions, which fire stochastically after an exponentially distributed waiting time.*
  2. T_{im} *is the set of immediate transitions, which fire with waiting time zero; they have higher priority compared with other transitions.*
  3. T_{timed} *is the set of deterministically delayed transitions, which fire after a deterministic time delay.*
  4. $T_{scheduled}$ *is the set of scheduled transitions, which fire at predefined time points.*
  5. T_{cont} *is the set of continuous transitions, which fire continuously over time.*
- $A = A_{disc} \cup A_{cont} \cup A_{inhibit} \cup A_{read} \cup A_{equal} \cup A_{reset} \cup A_{modifier}$ *is the set of directed arcs, with:*
  1. $A_{disc} \subseteq ((P \times T) \cup (T \times P))$ *defines the set of discrete arcs.*
  2. $A_{cont} \subseteq ((P_{cont} \times T) \cup (T \times P_{cont}))$ *defines the set of continuous arcs.*
  3. $A_{read} \subseteq (P \times T)$ *defines the set of read arcs.*
  4. $A_{inhibit} \subseteq (P \times T)$ *defines the set of inhibit arcs.*
  5. $A_{equal} \subseteq (P_{disc} \times T)$ *defines the set of equal arcs.*
  6. $A_{reset} \subseteq (P \times T^D)$ *defines the set of reset arcs,*
  7. $A_{modifier} \subseteq (P \times T)$ *defines the set of modifier arcs.*

  where $T^D = T_{stoch} \cup T_{im} \cup T_{timed} \cup T_{scheduled}$ *is the set of discrete transitions.*

- $\sum$ *is a finite, non-empty set of colour sets.*
- $C : P \rightarrow \sum$ *is a colour function that assigns to each place* $p \in P$ *a colour set* $C(p) \in \sum$.
- $F : A \rightarrow EXP$ *is an arc function that assigns to each arc* $a \in A$ *an arc expression of a multiset type* $C(p)_{MS}$*, where* p *is the place connected to the arc* a.
- V *is a set of functions* $V = \{g, d, w, f\}$ *where :*
  1. $g : T_{stoch} \rightarrow H_s$ *is a function which assigns a stochastic hazard function* h_{s_t} *to each stochastic transition instance* $t_j \in T_{stoch}$*, whereby* $H_s = \{h_{s_t} | h_{s_t} : \mathbb{R}_0^{|^\bullet t_j|} \rightarrow \mathbb{R}_0^+, t_j \in T_{stoch}\}$ *is the set of all stochastic hazard functions, and* $g(t_j) = h_{s_t}, \forall t_j \in T_{stoch}$.
  2. $w : T_{im} \rightarrow H_w$ *is a function which assigns a weight function* h_w *to each immediate transition instance* $t_j \in T_{im}$*, such that* $H_w = \{h_{w_t} | h_{w_t} : \mathbb{R}_0^{|^\bullet t_j|} \rightarrow \mathbb{R}_0^+, t_j \in T_{im}\}$ *is the set of all weight functions, and* $w(t_j) = h_{w_t}, \forall t_j \in T_{im}$.
  3. $d : T_{timed} \cup T_{scheduled} \rightarrow \mathbb{R}_0^+$*, is a function which assigns a constant time to each deterministically delayed and scheduled transition instance representing the (relative or absolute) waiting time.*
  4. $f : T_{cont} \rightarrow H_c$ *is a function which assigns a rate function* h_c *to each continuous transition instance* $t_j \in T_{cont}$*, such that* $H_c = \{h_{c_t} | h_{c_t} : \mathbb{R}_0^{|^\bullet t_j|} \rightarrow \mathbb{R}_0^+, t_j \in T_{cont}\}$ *is the set of all rates functions and* $f(t_j) = h_{c_t}, \forall t_j \in T_{cont}$.
- $G : T \rightarrow EXP$ *is a guard function that assigns to each transition* $t \in T$ *a guard expression of the Boolean type.*
- $m_0 : P \rightarrow EXP$ *is an initialisation function that assigns to each place* $p \in P$ *an initialisation expression of a multiset type* $C(p)_{MS}$.

Here, $\mathbb{R}_0^+$ *denotes the set of non-negative real numbers,* $^\bullet t_j$ *denotes the set of pre-places of a transition* t_j.

□

4.2 Semantics

The formal semantics of $\mathcal{GHPN}^{\mathcal{C}}$ is defined by unfolding the coloured hybrid Petri nets into the equivalent low level one. Thus we firstly discuss how $\mathcal{GHPN}^{\mathcal{C}}$ can be unfolded into $\mathcal{GHPN}$. Then, we extend the formal semantics which has been presented in [18] and apply it to $\mathcal{GHPN}^{\mathcal{C}}$.

Unfolding Uncoloured Petri nets can be folded into coloured Petri nets, if partitions of the place and transition sets are given. Vice versa, coloured Petri nets with finite colour sets can be automatically unfolded into uncoloured Petri nets, which then allows the use of all of the existing powerful standard Petri net analysis techniques. The conversion between uncoloured and coloured Petri nets changes the style of representation, but does not change the actual net structure of the underlying biological reaction network.

In [32], we have presented an efficient unfolding method for coloured Petri nets with finite coloured sets, in which we provide two approaches to compute transition instances. For a transition, if the colour set of each variable in its guard has a finite integer domain, a constraint satisfaction approach is used to obtain all valid transition instances. Otherwise, a general unfolding algorithm is adopted, in which some optimisation techniques are used like partial binding–partial test and pattern matching [27,32].

Hybrid semantics The semantics of $\mathcal{GHPN}^{\mathcal{C}}$ is given in terms of the discrete and continuous transitions. To harmonise the mathematical notations, let $T^C = T - T^D = T_{cont}$ denote the set of continuous transitions. Before we proceed, we assume that the coloured expression which is assigned to an arc is evaluated to a numeric value. Likewise the initial marking.

Definition 2 (Enabling condition) *Let $N =< P, T, A, \sum, C, F, V, G, m_0 >$ be a coloured generalised hybrid Petri net and m be the marking of N at time τ. A transition $t_j \in T$ is enabled in the marking m, denoted by $m[t_j\rangle$, iff $\forall p_i \in {}^\bullet t_j$:*

- $m(p_i) \geq F(p_i, t_j)$, *if* $(p_i, t_j) \in A_{cont} \cup A_{disc} \wedge t_j \in T^D$,
- $m(p_i) > 0$, *if* $(p_i, t_j) \in A_{cont} \wedge t_j \in T^C$,
- $m(p_i) \geq F(p_i, t_j)$, *if* $(p_i, t_j) \in A_{read}$,
- $m(p_i) < F(p_i, t_j)$, *if* $(p_i, t_j) \in A_{inhibit}$,
- $m(p_i) = F(p_i, t_j)$, *if* $(p_i, t_j) \in A_{equal}$.
- $G(t_j) = true$.

□

Definition 3 (Firing rule of discrete transitions) *Let $N =< P, T, A, \sum, C, F, V, G, m_0 >$ be a coloured generalised hybrid Petri net, m a marking of N, and $t_j \in T^D$ a transition enabled in the marking m, $m[t_j\rangle$, at time τ. The transition t_j can fire and reach a new marking m', denoted by $m[t_j\rangle m'$, at time $\tau + d_j$ if it is still enabled at that new time, with:*

- $\forall p_i \in {}^\bullet t_j$

$$m'(p_i) = \begin{cases} m(p_i) - F(p_i, t_j) & \textit{if } (p_i, t_j) \in A_{cont} \cup A_{disc} \\ 0 & \textit{if } (p_i, t_j) \in A_{reset} \\ m(p_i) & \textit{else} \end{cases}$$

- $\forall p_i \in t_j^\bullet$

$$m'(p_i) = m(p_i) + F(t_j, p_i)$$

where

$$d_j = \begin{cases} d(t_j) & \textit{if } t_j \in T_{timed} \\ \tau + d(t_j) & \textit{if } t_j \in T_{scheduled} \\ d_{stoch}(t_j) & \textit{if } t_j \in T_{stoch} \\ 0 & \textit{if } t_j \in T_{im} \end{cases}$$

is a delay which is associated to the discrete transition t_j and $d_{stoch}(t_j)$ is the random firing delay with negative exponential probability density function calculated for each stochastic transition t_j using its rate $g(t_j)$.

□

According to the above enabling and firing definitions, discrete transitions follow a policy which is called an *enabling memory policy* [24].

Moreover, when multiple immediate transitions are concurrently enabled, conflicts are solved by computing the relative firing frequencies of each enabled immediate transition. That is if an immediate transition t_j is enabled in the current marking m, then it fires with the probability given by (1).

$$\frac{w(t_j)(m)}{\sum\limits_{t_k \in T_{im} \wedge isEnabled(t_k,m)} w(t_k)(m)} \tag{1}$$

where w is the weight assigned to immediate transitions.

Firing of continuous transitions The semantics of continuous transitions is analogue to the ones in continuous Petri nets with maximal firing speeds depending on time as introduced in [6] and tailored to the specific needs in systems biology. The transitions' current firing rates (instantaneous firing speeds) depend on the current marking of their pre-places (i.e., species concentrations). In what follows, the firing semantics of continuous transitions is formally given.

We introduce the following notation. Let $v_j(\tau)$ represent the current firing rate of a transition $t_j \in T^C$ at time τ, $m_i(\tau) = m(p_i)$ denotes the current marking of a place p_i at time τ, and $f_j(\tau)$=$f(t_j)$ denotes the maximal firing rate of a transition t_j at time τ, then:

$$v_j(\tau) = \begin{cases} f_j(\tau) & \text{if } t_j \text{ is enabled} \\ 0 & \text{else} \end{cases} \tag{2}$$

Equation (2) implies that a continuous transition can fire with its maximal rate if it is enabled or its rate will be zero otherwise.

When a continuous transition is enabled, it fires as soon as possible and its effect on the connected places can be given by the following definition.

Definition 4 (Firing of continuous transitions) *Let $N =< P, T, A, \sum, C, F, V, G, m_0 >$ be a coloured generalised hybrid Petri net, m a marking of N, $t_j \in T^C$ a transition enabled in the marking m, $m[t_j\rangle$, at time τ, and $v_j(\tau)$ denotes the current firing rate of the transition t_j. The transition t_j fires with:*

– $\forall p_i \in {}^{\bullet}t_j$

$$m_i(\tau + d\tau) = m_i(\tau) - F(p_i, t_j) \cdot v_j(\tau)d\tau \tag{3}$$

– $\forall p_i \in t_j^{\bullet}$

$$m_i(\tau + d\tau) = m_i(\tau) + F(t_j, p_i) \cdot v_j(\tau)d\tau \tag{4}$$

□

Equations (3) and (4) are called outflow and inflow of a place p_i, respectively, due to the firing of a transition t_j [6].

Generation of the corresponding ODEs For a given transition $t_j \in T^C$, the functions $read(u, p_i)$, $inhibit(u, p_i)$ are defined as follows:

$$read(u, m(p_i)) = \begin{cases} 1 & \text{if } m(p_i) \geq u \\ 0 & \text{else} \end{cases}$$

with $u = F(p_i, t_j) \wedge (p_i, t_j) \in A_{read}$, and

$$inhibit(u, m(p_i)) = \begin{cases} 1 & \text{if } m(p_i) < u \\ 0 & \text{else} \end{cases}$$

with $u = F(p_i, t_j) \wedge (p_i, t_j) \in A_{inhibit}$.

Then the ODE corresponding to each continuous place in $\mathcal{GHPN^C}$ can be generated using (5)

$$\begin{aligned} \frac{dm\left(p_i\right)}{d\tau} = &\sum_{t_j \in {}^{\bullet}p_i} F\left(t_j, p_i\right) \cdot v_j\left(\tau\right) \cdot read(u, m(p_i)) \cdot inhibit(u, m(p_i)) - \\ &\sum_{t_j \in p_i{}^{\bullet}} F\left(p_i, t_j\right) \cdot v_j\left(\tau\right) \cdot read(u, m(p_i)) \cdot inhibit(u, m(p_i)) \end{aligned} \tag{5}$$

4.3 Simulation

The semantics of $\mathcal{GHPN^C}$ which we have discussed in Section 4.2 can be produced via simulation. In simulating $\mathcal{GHPN^C}$ models, the following two main issues need be considered: the partitioning of the net's transitions into stochastic and continuous ones, and the synchronisation between the stochastic and continuous regimes.

The partitioning of the net elements has an important influence on both the accuracy and the efficiency of $\mathcal{GHPN^C}$ simulation. The more continuous transitions we use, the faster the simulation speed we get. However, this will have a high impact on the result accuracy. Thus, it is important to appropriately partition the model transitions into stochastic and continuous ones. This can be done using either static or dynamic partitioning. In static partitioning, the process of assignment of each transition to either the stochastic or continuous category is done off line (i.e., before the simulation starts). Transitions with high firing rate will be considered continuously while transitions with low firing rates are considered stochastically. Additionally, the number of tokens of the transition's pre-places plays a role in determining the transition type. An important merit of this approach is that there is no additional overhead to the simulation due to the partitioning process. However, transition rates can dramatically change during the simulation. This motivates the use of dynamic partitioning. Using dynamic partitioning, transitions can change their type during simulation from stochastic to continuous or vice versa. Such change is usually based on either the transition rate and/or the number of tokens in the transition's pre-paces. Nevertheless, dynamic partitioning will result in additional overhead due to the repeated check of fulfilment of the partitioning conditions. Therefore, if the gain due to the use of dynamic partitioning is less than the partitioning overhead, static partitioning should be used instead. A detailed discussion about this issue can be found in [18].

Similarly, the synchronisation between stochastic and continuous regimes is vital for the accuracy of the simulation result. At the end, stochastic and continuous transitions are not isolated from each other. They mutually affect each other. Thus, we need a mechanism to better perform the synchronisation. One choice is to use (6) to detect the occurrence of a stochastic event while the numerical integration is progressing [12],

$$g(\mathbf{x}) = \int_{t}^{t+\tau} a_0^s(\mathbf{x})dt - \xi = 0\,, \tag{6}$$

where ξ is a random number exponentially distributed with a unit mean, and $a_0^s(\mathbf{x})$ is the cumulative rate of all stochastic transitions.

Once we agreed on a selection of a partitioning algorithm and a synchronisation approach we can execute the $\mathcal{GHPN}^{\mathcal{C}}$ model by repeating a set of steps. In [18], we have presented a hybrid simulation algorithm based on (6) using both static and dynamic partitioning. The main steps are outlined here while the details can be found in [18].

Starting from an initial marking, the simulation algorithm computes state changes over the time which are regarded as the current marking at each simulation step. Initially, the rates of stochastic and continuous transitions are calculated. Next, the accumulated rate function of stochastic transitions is calculated. When the model contains one or more continuous transitions, the simulation algorithm constructs an ODE for each place. ODE construction is done via (5). Afterwords, the simulator numerically integrates the resulting set of ODEs simultaneously with (6) until an event occurs. The event can be one of the following: the enabling of an immediate transition, the enabling of a deterministically delayed transition, a deterministically delayed transition has finished its delay, or a stochastic event has occurred (i.e., Equation (6) is satisfied).

In all cases the ODE integrator is interrupted and the appropriate action is taken (e.g., by firing a discrete transition). After that the ODE integrator is restarted with the new marking to account for the discontinuity which is due to the firing of a discrete transition.

When there is no continuous transition in the system being simulated, the simulation of $\mathcal{GHPN}^{\mathcal{C}}$ model is simplified to the simulation of stochastic Petri nets which can be carried out using, e.g., Gillespie's stochastic simulation algorithm [13].

5 Implementation

All features of $\mathcal{GHPN}^{\mathcal{C}}$ which have been discussed in this paper are implemented in Snoopy [15] – a unifying Petri net tool that supports the construction and execution of different Petri net classes among them are: standard, extended, continuous, stochastic, and hybrid Petri nets. Those classes are available both on the uncoloured and coloured level. Snoopy is platform-independent and provided free of charge for academic purposes.

Snoopy also supports the automatic unfolding of $\mathcal{GHPN}^{\mathcal{C}}$. In Snoopy, we can explicitly unfold a coloured stochastic/continuous/hybrid Petri net to its counterpart, a stochastic/continuous/hybrid Petri net, or we can do this implicitly during the execution of the Petri net model.

Furthermore, we have recently released a server-based version of the Snoopy simulator called S^4 (Snoopy Steering and Simulation Server) [19,20]. S^4 also supports the simulation of $\mathcal{GHPN}^{\mathcal{C}}$. Besides, an $\mathcal{GHPN}^{\mathcal{C}}$ model submitted to S^4 can be steered (i.e., key simulation parameters can be changed on the fly), and collaboratively executed by different users. Snoopy and S^4 can be downloaded from `http://www-dssz.informatik.tu-cottbus.de/snoopy.html`.

6 Applications

In this section we present one case study as a typical application of Coloured Generalised Hybrid Petri Nets.

We will demonstrate the $\mathcal{GHPN}^{\mathcal{C}}$ using an example of a synthetic circuit – the repressilator, which is an engineered synthetic system encoded on a plasmid, and designed to exhibit oscillations [7]. The repressilator system is a regulatory cycle of three genes, denoted by, e.g., gene_a, gene_b and gene_c, where each gene represses its successor, namely, gene_a inhibits gene_b, gene_b inhibits gene_c, and gene_c inhibits gene_a. This negative regulation is realised by the repressors, protein_a, protein_b and protein_c, generated by the genes gene_a, gene_b and gene_c, respectively.

The 1-bounded places as determined by P-invariant analysis and the related transitions as determined by T-invariant analysis are kept discrete. The unbounded places and related transitions are approximated by continuous places and transitions, respectively. That is, places gene_i and blocked_i, and transitions

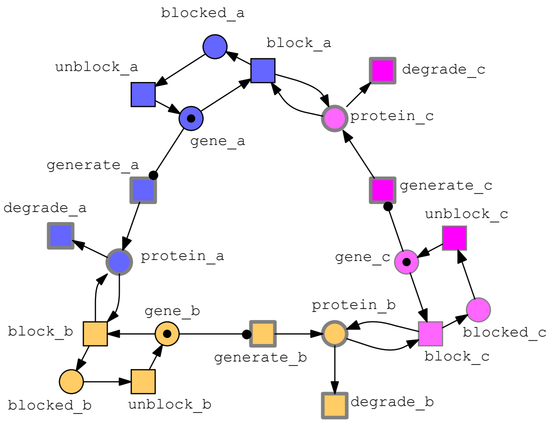


Fig. 2. The $\mathcal{GHPN}$ model of the repressilator, where the continuous places (transitions) are represented by shaded line circles (squares), which is taken from [31].

Table 1. Rate functions for the $\mathcal{GHPN}$ repressilator model. $MA(c)$ denotes the mass action function, where c is a kinetic parameter. See last column for the explicit rate functions for gene a.

transition class	kinetic parameter c	rate function pattern	example: gene a
generate	0.1	$MA(0.1)$	$0.1 * gene\_a$
block	1.0	$MA(1.0)$	$1.0 * gene\_a * protein\_c$
unblock	0.0001	$MA(0.0001)$	$0.0001 * blocked\_a$
degrade	0.001	$MA(0.001)$	$0.001 * protein\_a$

block_i and unblock_i are treated as discrete, where i=a,b,c, and all other nodes as continuous. To distinguish between discrete and continuous nodes, we choose different graphical representations, see Figure 2. The rate functions of the $\mathcal{GHPN}$ model are given in Table 1.

For example, if we assign the rates in Table 1 to the hybrid model and consider them as stochastic or deterministic rates, depending on the transition type, hybrid simulation yields plots as illustrated in Figure 3.

Figure 4 gives an $\mathcal{GHPN}^{\mathcal{C}}$ for the repressilator model in Figure 2. A colour set GeneSet is defined with three colours, a, b and c, to distinguish the three genes. Each place gets assigned this colour set GeneSet. A multiset expression, 1‘all()=1‘a++1‘b++1‘c, is assigned to the place *protein*. In Figure 4, we define a variable x of *GeneSet*, which is used in arc expressions. The predecessor operator “-” in the arc expression $-x$ returns the predecessor of x in an ordered finite

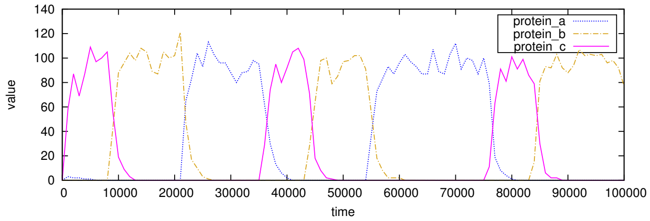


Fig. 3. Plot of one hybrid simulation run for the repressilator. For rate functions, see Table 1. This plot suggests that $\mathcal{GHPN}$ are able to capture the oscillation. Repeated runs look differently; thus stochasticity is captured as well.

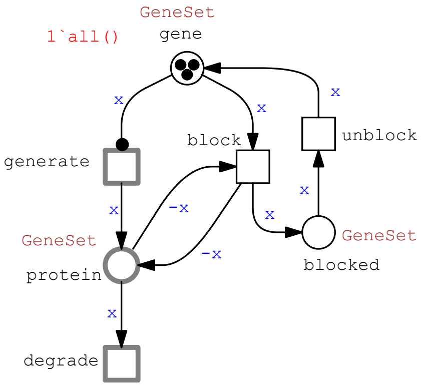


Fig. 4. A colored Petri net model for the repressilator. The declarations: colorset GeneSet = enum with a,b,c, and variable x: GeneSet. With this colour set, this model corresponds exactly to the Petri nets in Figure 2

colour set. For example, if $x = b$, then $-x$ returns a. If x is the first colour, then it returns the last colour. For example, if $x = a$, then $-x$ returns c.

The coloured Petri net model in Figure 4 when unfolded yields the same uncoloured Petri net model in Figure 2. That is, the $\mathcal{GHPN}^{\mathcal{C}}$ model reduces the size of the original stochastic Petri net model to one third, which is a big advantage of coloured Petri nets.

7 Conclusions and Future Work

In this paper we have introduced a new class of coloured Petri nets called Coloured Generalised Hybrid Petri Nets ($\mathcal{GHPN}^{\mathcal{C}}$). $\mathcal{GHPN}^{\mathcal{C}}$ are particularly tailored to systems biologists' needs to model and analyse multiscales models. Moreover, $\mathcal{GHPN}^{\mathcal{C}}$ provide the interplay between stochastic and continuous regimes on the coloured level which eventually provide a tool to control both the accuracy and the speed of running simulation.

In [21] we have proved that marking-dependent arc weights are necessary to model certain biological systems. However, this feature is still not implemented in Snoopy for $\mathcal{GHPN}^\mathcal{C}$. Thus we plan to add it in order to widen the types of models that can be constructed via $\mathcal{GHPN}^\mathcal{C}$.

Furthermore, the current simulation of $\mathcal{GHPN}^\mathcal{C}$ is carried out by firstly unfold the coloured model into an uncoloured one, then it is simulated on that level. Although the unfolding process is fully automated, it may take a considerable amount of time, particularly for bigger models. Thus if an error is observed at the (beginning) of the simulation due to modelling mistakes, the entire unfolding process will completely be repeated. Therefore, simulating $\mathcal{GHPN}^\mathcal{C}$ directly on the coloured level [9] will increase the productivity and eventually save a lot of time.

Finally, the presented case study in this paper aims to demonstrate how $\mathcal{GHPN}^\mathcal{C}$ works. More sophisticated biological models can be constructed in the future using $\mathcal{GHPN}^\mathcal{C}$. For example, we are working on using $\mathcal{GHPN}^\mathcal{C}$ to model diffusion-reaction systems described by partial differential equations, where we first discretise the space into a set of grid cells, and then we obtain a $\mathcal{GHPN}^\mathcal{C}$ model by representing each grid cell as a colour [28].

References

1. Alla, H., David, R.: Continuous and hybrid Petri nets. J. Circ. Syst. Comp. 8(1), 159 –188 (1998)
2. Banks, R., Steggles, L.J.: A high-level Petri net framework for genetic regulatory networks. Journal of Integrative Bioinformatics 4(3), 1–12 (2007)
3. Chaouiya, C., Remy, E., Thieffry, D.: Qualitative Petri net modelling of genetic networks. In: Transactions on Computational Systems Biology. pp. 95–112. LNCS 4220, Springer (2006)
4. Christensen, S., Jørgensen, J.B., Kristensen, L.M.: Design/CPN - a computer tool for coloured Petri nets. In: Proc. of the Third International Workshop on Tools and Algorithms for Construction and Analysis of Systems. pp. 209–223. LNCS 1217, Springer (1997)
5. Comet, J., Klaudel, H., Liauzu, S.: Modeling multi-valued genetic regulatory networks using high-level Petri nets. In: Proc. of International Conference on Application and Theory of Petri Nets. pp. 208–227 (2005)
6. David, R., Alla, H.: Discrete, Continuous, and Hybrid Petri Nets. Springer (2010)
7. Elowitz, M.B., Leibler, S.: A synthetic oscillatory network of transcriptional regulators. Nature 403(6767), 335–338 (2000)
8. Fujita, S., Matsui, M., Matsuno, H., Miyano, S.: Modeling and simulation of fission yeast cell cycle on hybrid functional Petri net. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E87-A(11), 2919–2927 (2004)
9. Gaeta, R.: Efficient discrete-event simulation of colored Petri nets. IEEE Transactions on Software Engineering 22(9), 629–639 (1996)
10. Gao, Q., Gilbert, D., Heiner, M., Liu, F., Maccagnola, D., Tree, D.: Multiscale Modelling and Analysis of Planar Cell Polarity in the Drosophila Wing. IEEE/ACM Transactions on Computational Biology and Bioinformatics 10(2), 337–351 (2013)

11. Genrich, H., Küffner, R., Voss, K.: Executable Petri net models for the analysis of metabolic pathways. International Journal on Software Tools for Technology Transfer 3(4), 394–404 (2001)
12. Gillespie, D.: Markov processes: an introduction for physical scientists. Academic Press (1991)
13. Gillespie, D.: Stochastic simulation of chemical kinetics. Annual review of physical chemistry 58(1), 35–55 (2007)
14. Haseltine, E., Rawlings, J.: Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. J. Chem. Phys. 117(15), 6959–6969 (2002)
15. Heiner, M., Herajy, M., Liu, F., Rohr, C., Schwarick, M.: Snoopy – a unifying Petri net tool. In: Haddad, S., Pomello, L. (eds.) Proc. PETRI NETS 2012. LNCS, vol. 7347, pp. 398–407. Springer (2012)
16. Heiner, M., Lehrack, S., Gilbert, D., Marwan, W.: Extended stochastic Petri nets for model-based design of wetlab experiments. In: Transactions on Computational Systems Biology XI, pp. 138–163. Springer, Berlin, Heidelberg (2009)
17. Herajy, M.: Computational Steering of Multi-Scale Biochemical Networks. Ph.D. thesis, BTU Cottbus, Dep. of CS (January 2013)
18. Herajy, M., Heiner, M.: Hybrid representation and simulation of stiff biochemical networks. J. Nonlinear Analysis: Hybrid Systems 6(4), 942–959 (November 2012)
19. Herajy, M., Heiner, M.: A Steering Server for Collaborative Simulation of Quantitative Petri Nets. In: Ciardo, G., Kindler, E. (eds.) Proc. PETRI NETS 2014. LNCS, vol. to appear. Springer (June 2014)
20. Herajy, M., Heiner, M.: Petri net-based collaborative simulation and steering of biochemical reaction networks. Fundamenta Informatica (129), 49–67 (2014)
21. Herajy, M., Schwarick, M., Heiner, M.: Hybrid Petri Nets for Modelling the Eukaryotic Cell Cycle. ToPNoC VIII, LNCS 8100 pp. 123–141 (2013)
22. Jensen, K.: Coloured Petri nets and the invariant-method. Theoretical Computer Science 14(3), 317–336 (1981)
23. Jensen, K., Kristensen, L.: Coloured Petri Nets. Springer (2009)
24. Kartson, D., Balbo, G., Donatelli, S., Franceschinis, G., Conte, G.: Modelling with generalized stochastic Petri nets. John Wiley & Sons, Inc., 1st edn. (1994)
25. Kiehl, T., Mattheyses, R., Simmons, M.: Hybrid simulation of cellular behavior. Bioinformatics 20, 316–322 (February 2004)
26. Lee, D., Zimmer, R., Lee, S., Park, S.: Colored Petri net modeling and simulation of signal transduction pathways. Metabolic Engineering 8(2), 112–122 (2006)
27. Liu, F.: Colored Petri Nets for Systems Biology. Ph.D. thesis, Department of Computer Science, Brandenburg University of Technology Cottbus (2012)
28. Liu, F., Blätke, M., Heiner, M., Yang, M.: Modelling and simulating reaction-diffusion systems using coloured petri nets. Computers in Biology and Medicine p. under review (2014)
29. Liu, F., Heiner, M.: Modeling membrane systems using colored stochastic Petri nets . Nat. Computing 12(4), 617 – 629 (2013)
30. Liu, F., Heiner, M.: Multiscale modelling of coupled Ca^{2+} channels using coloured stochastic Petri nets. IET Systems Biology 7(4), 106 – 113 (August 2013)
31. Liu, F., Heiner, M.: Petri Nets for Modeling and Analyzing Biochemical Reaction Networks, chap. 9, pp. 245–272. Springer (2014)
32. Liu, F., Heiner, M., Yang, M.: An efficient method for unfolding colored Petri nets. In: Proc. of the Winter Simulation Conference. IEEE (2012)
33. Marwan, W., Rohr, C., Heiner, M.: Petri nets in Snoopy: A unifying framework for the graphical display, computational modelling, and simulation of bacterial regulatory networks, Methods in Molec. Biol., vol. 804, chap. 21, pp. 409–437 (2012)

34. Matsui, M., Fujita, S., Suzuki, S., Matsuno, H., Miyano, S.: Simulated cell division processes of the xenopus cell cycle pathway by genomic object net. Journal of Integrative Bioinformatics p. 0001 (2004)
35. Matsuno, H., Nagasaki, M., Miyano, S.: Hybrid Petri net based modeling for biological pathway simulation. Natural Computing 10(3), 1099–1120 (2011)
36. Matsuno, H., Tanaka, Y., Aoshima, H., Doi, A., Matsui, M., Miyano, S.: Biopathways representation and simulation on hybrid functional Petri net. In silico biology 3(3) (2003)
37. Parvu, O., Gilbert, D., Heiner, M., Liu, F., Saunders, N.: Modelling and Analysis of Phase Variation in Bacterial Colony Growth. In: Gupta, A., Henzinger, T. (eds.) Proc. CMSB 2013. LNCS/LNBI, vol. 8130, pp. 78–91. Springer (September 2013)
38. Peleg, M., Gabashvili, I.S., Altman, R.B.: Qualitative models of molecular function: Linking genetic polymorphisms of trna to their functional sequelae. Proceedings of the IEEE 90(12), 1875–1886 (2002)
39. Reddy, V., Mavrovouniotis, M., Liebman, M.: Petri net representations in metabolic pathways. In: Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology. pp. 328–336 (1993)
40. Runge, T.: Application of coloured Petri nets in systems biology. In: Proc. of the 5th Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools. pp. 77–95. University of Aarhus (2004)
41. Täubner, C., Mathiak, B., Kupfer, A., Fleischer, N., Eckstein, S.: Modelling and simulation of the tlr4 pathway with coloured Petri nets. In: Proc. of the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. pp. 2009–2012. IEEE (2006)
42. Voss, K., Heiner, M., Koch, I.: Steady state analysis of metabolic pathways using Petri nets. In Silico Biology 3, 0031 (2003)

Author Index