Gianfranco Balbo
Monika Heiner

Biological Processes & Petri Nets

Editors:

Gianfranco Balbo
Università degli Studi di Torino, Dipartimento di Informatica, Italy
`balbo@di.unito.it`

Monika Heiner
Brandenburgische Technische Universität Cottbus, Institut für Informatik,
03013 Cottbus, Germany
`monika.heiner@informatik.tu-cottbus.de`

BibTEX entry:

```
@proceedings{bioppn2013,
  editor    = {Gianfranco Balbo and Monika Heiner},
  title     = {Proceedings of the 4th International Workshop on
                Biological Processes \& Petri Nets (BioPPN 2013),
                satellite event of Petri Nets 2013, Milano, Italy, June 24, 2013},
  booktitle = {Biological Processes \& Petri Nets},
  location  = {Milano, Italy},
  publisher = {CEUR-WS.org},
  series    = {CEUR Workshop Proceedings},
  volume    = {Vol-988},
  year      = {2013},
  url       = {http://CEUR-WS.org/Vol-988/}
}
```

Preface

These proceedings contain the six peer-reviewed contributions as well as the abstract of one complementary accepted presentation of the Fourth International Workshop on Biological Processes & Petri Nets (BioPPN 2013), held as a satellite event of PETRI NETS 2013, in Milano, Italy, on Monday, June 24, 2013.

The workshop has 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.

We received two types of contributions: research papers and work-in-progress papers. All submissions have been reviewed by four to six reviewers coming from or being recommended by the workshop's Program Committee. The list of reviewers comprises 20 professionals of the field, writing in total 37 reviews. The six accepted peer-reviewed papers (with an acceptance rate of 75%) involve 25 authors coming from 6 different countries. In summary, the workshop proceedings enclose theoretical contributions as well as biological applications, demonstrating the interdisciplinary nature of the topic.

The workshop programme was complemented by the invited talk *'Dreaming about models: a biologist's perspective'* given by Raffaele Calogero from the Medical School of the University of Torino, Italy.

For more details see the workshop's website http://www-dssz.informatik.tu-cottbus.de/BME/BioPPN2013.

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, 2013   Gianfranco Balbo
                                                      Monika Heiner

iii

Table of Contents

Program Committee

Gianfranco Balbo	I
Rainer Breitling	UK
Jorge Carneiro	PT
Claudine Chaouiya	PT
Ming Chen	CN
David Gilbert	UK
Simon Hardy	CA
Monika Heiner	D
Peter Kemper	US
Chen Li	CN
Fei Liu	CN
Wolfgang Marwan	D
Hiroshi Matsuno	JP
K. Sriram	IN
P.S. Thiagarajan	SG

Additional Reviewers

G
Gao, Qian
Grosan, Crina

H
Herajy, Mostafa

P
Parvu, Ovidiu

S
Sugii, Manabu

Comparing Metabolic Pathways through Potential Fluxes: a Selectively Open Approach

Paolo Baldan[1], Martina Bocci[2], Nicoletta Cocco[2] and Marta Simeoni[2]

[1] Dipartimento di Matematica, Università di Padova
via Trieste 63, 35121 Padova, Italy
email: baldan@math.unipd.it
[2]Dipartimento di Scienze Ambientali, Statistica e Informatica,
Università Ca' Foscari Venezia,
via Torino 155, 30172 Venezia Mestre, Italy
email: [martina.bocci, cocco,simeoni]@unive.it

Abstract. In our previous work we developed CoMeta, a tool for comparing metabolic pathways of different organisms, using the KEGG database as data source. The similarity measure adopted combines homology of reactions and functional aspects of the pathways. The latter are captured by T-invariants in the Petri net representation, which correspond to potential fluxes in the pathways. A Petri net can model a metabolic pathway of an organism either in isolation, focussing on its internal behaviour (isolated net), or as an interactive subsystem of the full metabolic network (open net). Modelling a pathway as an isolated net normally works fine for comparison purposes, but unsatisfactory results can arise as it supplies a partial view on internal fluxes. A representation as an open net makes additional information available, but the choice of the interactions of the pathway with the environment is non-trivial. Considering all possible interactions with the environment (an information automatically retrieved from KEGG) is not appropriate. Some interactions may add noise to the model, the size of invariants bases grows up to an order of magnitude and the comparison results might be less precise than with the isolated representation. Here we propose an extension of CoMeta which allows the user to select which metabolites should be considered as interactions of interest, discriminating between input and output metabolites. We illustrate some experiments which show the advantages of this more flexible approach. Our experience suggests that in general a good choice is to take as open metabolites those which are the input and output compounds for the pathway.

1 Introduction

Subsystems of metabolism dealing with some specific function are called metabolic pathways. Comparing metabolic pathways of different species yields interesting information on their evolution and it may help in understanding metabolic functions. This is important for metabolic engineering and for studying diseases and drugs design.

In [9, 6] we proposed to represent pathways as Petri nets (PNs) and compare them by considering static aspects, provided by the reactions, and information on the behaviour, as captured by the T-invariant bases of the corresponding Petri net models. Petri nets seem to be particularly natural for modelling metabolic pathways (see, e.g., [7] and references therein). The graphical representations used by biologists for metabolic pathways and the ones used in PNs are similar; the stoichiometric matrix of a metabolic pathway is analogous to the incidence matrix of a PN; the flux modes and the conservation relations for metabolites correspond to specific properties of PNs. In particular minimal (semi-positive) T-invariants correspond to elementary flux modes [21] of a metabolic pathway, i.e., minimal sets of reactions that can operate at a steady state. The space of semi-positive T-invariants has a unique basis of minimal T-invariants which is characteristic of the net and we used it in the comparison.

We developed CoMeta, a tool implementing our proposal. Given a set of organisms and a set of metabolic pathways, CoMeta automatically gets the corresponding data from the KEGG database [2], builds the corresponding Petri nets, computes the T-invariants and the similarity measure, and shows the results of the comparison among organisms as a phylogenetic tree.

The prototype version of CoMeta presented in [9] produced *isolated* PN models. In an *isolated model* the connections of the metabolic pathway with the environment are not represented. The potential fluxes which can be observed and compared with thus only the internal ones. According to our experiments, isolated net models normally work fine, but in some cases they may lead to unsatisfactory results. This happens when internal fluxes do not sufficiently characterise the behaviour of the net, for example when a pathway has very few internal cycles. In this case neglecting the interactions with the enviroment becomes problematic.

In [6] an extended version of CoMeta is proposed which gives the choice of producing either *isolated* or *open* PN models. In an *open model*, in order to express the interaction of the pathway with the environment, some compounds are represented as open places, i.e. places where the environment can freely put/remove substances through corresponding input/output transitions. Open places may be both the compounds which link the pathway to the rest of the metabolic network and the compounds which are only substrates or only products (the sources and the sinks of the net). In [6], by choosing an open model representation, all the compounds shared with the rest of the network and all the sources and sinks of a pathway are automatically modelled as open places. But further experiments show that this choice might lead to unsatisfactory results: some interactions reported in KEGG may be not precise or they may introduce noise in the comparison. Moreover, the added open places determine a growth in the size of invariants bases up to an order of magnitude, with consequences on the effieciency of the comparison.

The new version of CoMeta presented in this paper extends the previous ones with the possibility, for the user, to selectively open the model. The set of potentially open compounds is proposed to the user, who can decide, on

the basis of her/his knowledge, which should be the open input and output metabolites. Then, in addition to the internal fluxes, potential fluxes involving the chosen input/output metabolites will be considered in the comparison. From our experience it is always convenient to open the sources and the sinks in the networks. Hence currently in COMETA this is the default choice proposed to the user, which is however free to add or remove metabolites as open places.

The paper is organised as follows. In Section 2 we show how a Petri net can model a metabolic pathway in the isolated and open approach. In Section 3 we briefly illustrate the new version of COMETA and in Section 4 we present some experiments with it. A short conclusion follows in Section 5.

2   Petri net representation of a metabolic pathway

PNs are a well known formalism originally introduced in computer science for modelling discrete concurrent systems. PNs have a sound theory and many applications both in computer science and in real life systems (see [16] and [10] for surveys on PNs and their properties). A large number of tools have been developed for analysing properties of PNs. A quite comprehensive list can be found at the *Petri Nets World* site [4].

Starting with [19, 14], Petri nets have been used as a model for representing and analysing metabolic pathways. A large body of literature exists on the topic (see, e.g, [7] for a survey). The structural representation of a metabolic pathway by means of a PN can be obtained by exploiting the natural correspondence between PNs and biochemical networks. In fact places are associated with molecular species, such as metabolites, proteins or enzymes; transitions correspond to chemical reactions; input places represent the substrate or reactants; output places represent reaction products. The incidence matrix of the PN is identical to the stoichiometric matrix of the system of chemical reactions. The number of tokens in each place indicates the amount of substance associated with that place. Quantitative data can be added to refine the representation. In particular, extended PNs can be enriched with a transition rate which depends on the kinetic law of the corresponding reaction.

When metabolic pathways are represented as Petri nets, we may consider their behavioural aspects as captured by the T-invariants (transition invariants) of the nets which, roughly, represent potential cyclic behaviours in the system. More precisely a T-invariant is a multiset of transitions whose execution starting from a state will bring the system back to the same state. Therefore presence of T-invariants in a metabolic pathway is biologically of great interest as it can reveal the presence of steady states, in which concentrations of substances have reached a possibly dynamic equilibrium.

The set of semi-positive T-invariants of a finite PN N admits a finitary representation by means of the so-called Hilbert basis [20], denoted $\mathcal{B}(N)$, which consists of the set of minimal T-invariants. Any T-invariant can be obtained as a linear combination (with positive integer coefficient) of elements of the basis.
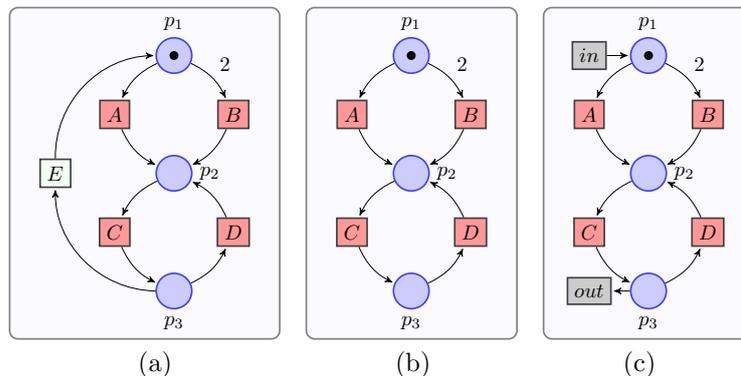
Fig. 1. A net system.

Uniqueness of the basis $\mathcal{B}(N)$ allows us to take it as a characteristic feature of the net.

In a PN model of a metabolic pathway, a minimal T-invariant corresponds to an elementary flux mode, a term introduced in [21] to refer to a minimal set of reactions that can operate at a steady state. It can be interpreted as a minimal self-sufficient subsystem which is associated to a function. Minimal T-invariants have been used in Systems Biology as fundamental tool in model validation techniques (see, e.g., [13, 15]) and in analysis and decomposition techniques (see, e.g., [12, 11]).

The Petri nets corresponding to the metabolic pathways of an organism are subnets of a larger net representing its full metabolic network. They can be considered as *isolated* subnets, by ignoring their interactions with the environment, or as *open* subnets, i.e., interactive subsystems which exchange compounds with the environment. This is obtained by taking their input/output metabolites as open places, where the environment can freely put/remove substances. The minimal T-invariants of these subnets have a clear relation with (minimal) T-invariants of the full network. It can be easily seen that modelling the pathway as an isolated subsystem guarantees correctness: minimal T-invariants of the pathway are minimal T-invariants of the full network, but they capture only internal fluxes. If, instead, we consider the pathway as an open subsystem, then we get completeness: any invariant of the full network, once projected onto the pathway, is an invariant of the open pathway. The converse does not hold, i.e., there may be invariants of the open pathway which do not correspond to invariants of the full network. Hence, in the open approach, we may loose correctness, but, still, as shown in [18], minimal T-invariants of the full network can be obtained compositionally from those of the subnetworks.

As an example, consider the simple Petri net in Fig. 1(a). It has two minimal invariants, namely $I_1 = \{A, C, E\}$ and $I_2 = \{C, D\}$. Note that $\{B, C, E\}$ is not an invariant, since B requires two tokens in p_1. Assume that the subnet

of interest consists of the dark red transitions A, B, C, D (with their pre- and post-places, i.e., p_1, p_2 and p_3). The isolated representation of this subnet is given in Fig. 1(b). It is obtained by just removing transition E. Note that invariant I_2 is still there, while I_1 is lost. In the open representation of Fig. 1(c), places p_1 and p_2 are opened in input and output, respectively, meaning that the environment can put and remove arbitrarily many tokens in such places. This is represented by inserting the transitions *in* and *out*. As a consequence, there are three invariants in the open subnet: I_2, which was already in the original net, $\{in, A, C, out\}$, which is the projection of I_1 over the subnet, and $\{2 \cdot in, B, C, out\}$ which, instead, does not correspond to any invariant of the original net.

The present version of COMETA allows the user to choose either the isolated or the open view, and, in the latter case, to finely tune the representation of the compounds on which the interaction with the environment takes place.

3   The tool CoMeta

COMETA, (COmparing METAbolic pathways) is a tool for comparing metabolic pathways in different organisms relying on their PN representation. The comparison is based on the combination of two distances, a "static" one, d_R, taking into account the reactions in the pathways and a "behavioural" one, d_I, taking into account potential fluxes in the pathways at steady state, as expressed by the T-invariants of the corresponding PNs. Given two pathways represented as PNs, P_1 and P_2, each distance is derived from a corresponding similarity score: $d_X(P_1, P_2) = 1 - score_X(P_1, P_2)$, with $X \in \{R, I\}$.

When computing d_R, $score_R(P_1, P_2)$ represents the similarity between the reactions in P_1 and the ones in P_2. Each reaction is represented by the enzymes which catalyse it and, in turn, each enzyme is identified by its EC number [26]. The similarity between enzymes is simply the identity, but finer similarity measures between enzymes could be easily accommodated in our setting. Concretely, $score_R(P_1, P_2)$ is a similarity index between the multisets of the EC numbers associated to the reactions in P_1 and P_2, respectively. The present version of COMETA offers the choice between the Sørensen [24] and the Tanimoto [25] index extended to multisets.

When computing d_I, the sets of minimal T-invariants (Hilbert bases) $\mathcal{B}(P_1)$ and $\mathcal{B}(P_2)$ of the two nets are compared. Each invariant is represented as a multiset of EC numbers, corresponding to the reactions in the invariant, and the similarity between two invariants is given, as before, by a similarity index. The similarity score is computed through a heuristic match between the two Hilbert bases and it represents the similarity of the matching pairs. In COMETA the two distances may be combined: $d_C(P_1, P_2) = \alpha\, d_R(P_1, P_2) + (1-\alpha)\, d_I(P_1, P_2)$, with $\alpha \in [0, 1]$, to move the focus between reactions and functional components, and two organisms can be compared on n metabolic pathways $P_1, \ldots, P_n$ by considering their average distance on the n pathways. More details on the distances may be found in [6] where a prototype version of COMETA was presented.

CoMeta is a user-friendly tool written in Java and running under Linux and Mac. CoMeta offers a set of integrated functionalities through a graphical user interface shown in Figure 2(a). In the upper part of the window the desired KEGG organisms and pathways can be selected. In the lower part a tabbed panel offers the commands to be performed. The first tab of the panel is shown in the main window, while the others are shown in Figure 2(b), 2(c), and 2(d), respectively. The main functionalities of the tool are the following ones:
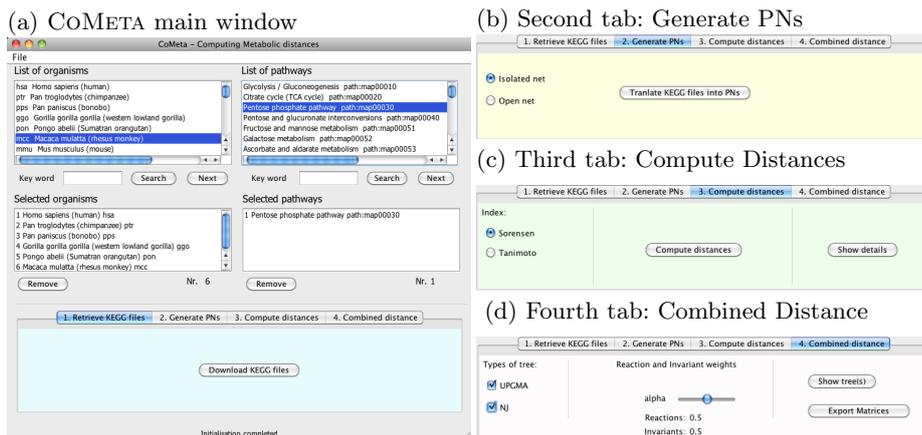
(a) CoMeta main window

(b) Second tab: Generate PNs

(c) Third tab: Compute Distances

(d) Fourth tab: Combined Distance

Fig. 2. The CoMeta graphical user interface

- *Select organisms and pathways* (Figure 2(a)): CoMeta proposes the lists of all KEGG organisms and pathways and allows the user to select the ones to be compared by double-clicking them.
- *Retrieve KEGG information:* CoMeta automatically downloads from the KEGG database the selected organisms and pathways.
- *Translate into PNs* (Figure 2(b)): CoMeta translates the selected organisms and pathways into corresponding PNs by using the tool *MPath2PN* [8]. *MPath2PN* produces a translation enzyme-based and without ubiquitous substances from KGML (KEGG Markup Language) [1] to PNML [3], a standard format for PNs tools. CoMeta produces the stoichiometric matrix of the net in a text file.

  Currently CoMeta offers the possibility of representing the pathways either as an isolated or as an open subnet of the full metabolism. The user can model the pathway as an isolated subsystem and focus only on internal fluxes, or he can consider an open net and choose the open places among the compounds shared with the rest of the network and the compounds which are sources/sinks for the pathway. To assist the user, the tool proposes a canonical choice of open places, namely the sources and the sinks of the net, but this choice can be modified by adding and removing places in the

list of potential open places of the specific pathway and organism. Figure 3 shows the selectively open window for the organism *Vitis vinifera* wrt. the *Sulfate metabolism* pathway. Note that the first three checkbox columns in the window specify which metabolites link to other pathways and which ones are sources or sinks. By clicking on the checkboxes in the two rightmost columns the user can select to open in input or in output any metabolite in the pathway. The canonical choice, sources in input and sinks in output, automatically selected and proposed to the user, is shown in Figure 3.

Fig. 3. The selectively open window with the canonical choice for *Vitis vinifera* wrt. *Sulfate metabolism*

- *Compute Distances* (Figure 2(c)): d_R and d_I are computed as previously described. The user can select either the Sørensen or the Tanimoto index. For computing Hilbert bases CoMeta resorts to 4ti2 [5], an efficient tool offered in a software package for solving algebraic, geometric and combinatorial problems on linear spaces. The details of the comparison between any pair of organisms (T-invariants bases, invariants matches, reactions and invariants scores, etc.) can be displayed to be analysed by the user.
- *Show Phylogenetic trees* (Figure 2(d)): CoMeta computes d_C, the distance which combines d_R and d_I according to a weight parameter α specified by the user. Such a distance may be used to produce and visualise corresponding phylogenetic trees. The user can specify the method for the generation of the phylogenetic trees. Currently CoMeta offers the UPGMA [23, 22] and Neighbour Joining methods [17, 22]. The matrices for d_R, d_I and d_C can be exported as text files for further analyses.

4   Experiments

In this section we discuss some experiments performed with CoMeta in order to illustrate how the choice of the isolated or the open approach may affect the results of the comparison of metabolic pathways. We consider a small group of organisms and analyse d_I, the distance based on T-invariants, with the Sørensen

index, when modelling the pathways as isolated, fully open and selectively open
PNs. By fully open we mean a model in which all potentially open places, namely
the ones linking the pathway to the rest of the metabolic network and the ones
which are only substrates (sources) or only products (sinks), are indeed open.
In the selectively open approach we use the default choice, i.e., we open in input
source places by adding an input transition to each source, and we open in output
sink places by adding an output transition to each sink.

The following experiments have a common feature: the pathways of the or-
ganisms we compare have few reversible reactions and few internal cycles. As
a consequence, the comparison of the isolated PN models is not very detailed
because of the small number of internal T-invariants and it produces only a
rough classification of the organisms. On the other hand, by considering fully
open models the information on the links among pathways given by KEGG be-
come mostly relevant in the comparison, even if they are imprecise or not so
important for distinguihing the specific functionalities. The classification of the
organisms results distorted. In such cases, the selectively open approach seems
to give the best results in the comparison, in fact it permits to add relevant in-
formation to the pathway model without overweighting boundary information.
The resulting classification is more precise than in the other two approaches. In
particular the canonical choice, i.e. opening in input source places and in output
sink places, can be used with good results when no special knowledge on the
pathway boundary is available.

4.1 Sulfur metabolism pathway

The *Sulfur metabolism* pathway describes the sulfur metabolism, including re-
duction and fixation processes. Sulfur enters in the composition of proteins
(amino acids cysteine and methionine) and from the catabolism of these amino
acids, it is liberated in the form of hydrogen sulphide (H_2S). Bacteria in soils and
waters oxidise hydrogen sulfide in various steps, to its highest oxidation state −
sulphate (SO_4^{2-}). Algae, Plants and Bacteria are capable to take the sulfur as
sulphate and to process it to the most reduced form (sulfide) for incorporation
into amino acids (cysteine and methionine). Animals are not able to synthesise
methionine which is an essential amino acid to be assumed with the diet.

We report here on two different experiments performed with *Sulfur metabolism*.
The first experiment aims at evaluating the ability of d_I to discriminate among
very different groups of organisms. The second experiment aims at checking
whether d_I is able to identify fine-grained differences among organisms.

First experiment. For this experiment we consider Archaea, Bacteria, Fungi,
Plants (that are able to utilise sulfur as sulphate), Birds and Mammals (Animals,
other than ruminants, take up sulfate only in reduced form in amino acids).
The organisms are therefore expected to show similarity within each group and
strong dissimilarity between groups. The list of selected organisms is shown in
the following table.

Code	Organism	Reign
hsa	*Homo sapiens*	*Mammals*
ecb	*Equus caballus*	*Mammals*
gga	*Gallus gallus*	*Birds*
tgu	*Taeniopygia guttataa*	*Birds*
ath	*Arabidopsis thaliana*	*Plants*
osa	*Oryza sativa japonica*	*Plants*
bdi	*Brachypodium distachyon*	*Plants*
nfi	*Neosartorya fischeri*	*Fungi*
ang	*Aspergillus niger*	*Fungi*
cpw	*Coccidioides posadasii*	*Fungi*
cow	*Caldicellulosiruptor owensensis*	*Bacteria*
toc	*Thermosediminibacter oceani*	*Bacteria*
hsl	*Halobacterium salinarum R1*	*Archaea*
hvo	*Haloferax volcanii*	*Archaea*
pto	*Picrophilus torridus*	*Archaea*

By using CoMeta, we compute d_I, the distance based on T-invariants, for the isolated, selectively open and fully open approaches. The *Sulfur metabolism*
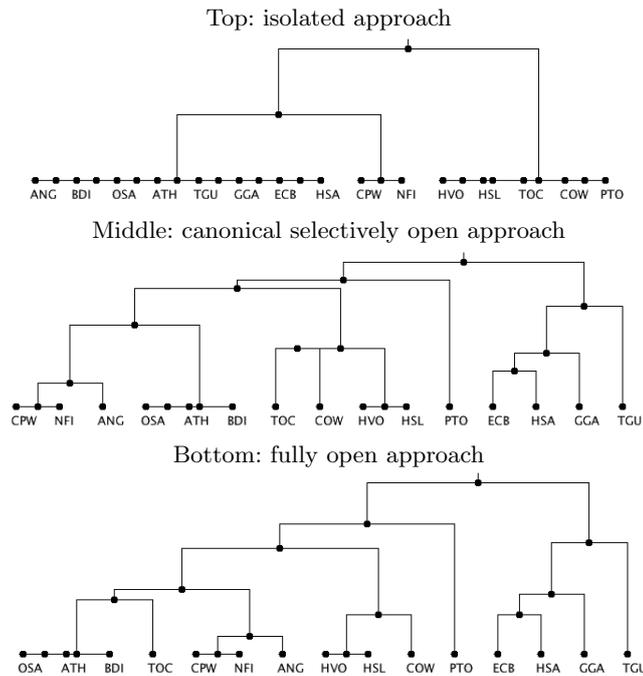

Fig. 4. First experiment: UPGMA trees based on d_I wrt. *Sulfur metabolism*

pathway has very small PN models. Depending on the organism, in the isolated models there are at most 9 enzymes/reactions and at most 1 T-invariant, in

the fully open models at most 19 enzymes/reactions and 6 invariants and in the canonical selectively open models at most 15 enzymes/reactions and 5 invariants. The selection is among 13 compounds at most.

Figure 4 shows the UPGMA tree corresponding to d_I in the isolated (top tree), selectively open (middle tree) and fully open (bottom tree) approaches. Note that in the isolated approach (top tree) Archaea and Bacteria are first discriminated from Fungi, Plants and Animals. At a second level, two out of three Fungi species are discriminated from Plants and Animals. No other grouping is evident and the classification is rather coarse. The selectively open approach (middle tree) discriminates at a first level the Animals (Mammals and Birds) from other groups. At a lower classification level all the three Fungi species are grouped together, as well as the three Plants species and four out of five Archaea and Bacteria species. Only the Archaea *pto* is not corectly grouped with the other Archaea (*hsl* and *hvo*) and Bacteria (*cow*, *toc*). Finally, with the fully open approach (bottom tree) the Archaea and Bacteria groups are not so well defined as in the previous case, i.e. *toc* and *cow* are not grouped together. Hence, in this experiment, the selectively open approach seems to better identify and group together organisms according to major taxonomic groups.

Second experiment We consider the *Sulfur metabolism* and the organisms in the following table.

Code	Organism	Reign
pae	*Pseudomonas aeruginosa PAO1*	*Bacteria*
pfo	*Pseudomonas fluorescens Pf0-1*	*Bacteria*
tin	*Thiomonas intermedia*	*Bacteria*
tcx	*Thiomicrospira crunogena*	*Bacteria*
cpr	*Clostridium perfringens SM101*	*Bacteria*
cst	*Clostridium stricklandii*	*Bacteria*
ddn	*Desulfovibrio desulfuricans ND132*	*Bacteria*
vvi	*Vitis vinifera*	*Plants*
zma	*Zea mays*	*Plants*

For this experiment we select within the Bacteria Reign some species having different sulfur metabolism and playing different roles within the bio-geo-chemical cycle of sulfur. The organisms *pae* and *pfo* are capable to oxidize elemental sulfur to sulfate. Sulfate can be assimilated by Plants and by Bacteria, such as the Clostridium species considered in the experiment. The sulfate-reducing bacteria, as *ddn*, are able to reduce sulfate to sulphide and responsible of bio-corrosion. On the contrary, *tin* and *tcx* oxidize sulphide back to sulfur. By using CoMeta, we compute d_I, the distance based on T-invariants. Figure 5 shows the UPGMA tree corresponding to d_I in the isolated (top tree), selectively open (middle tree) and fully open (bottom tree) approaches. Note that with the isolated approach (top tree) Plants, which assimilate sulfur as sulphate from the soil, are classified together with Bacteria (Pseudomonas genus) which are capable of oxidising sulfur to sulphate. On the right side of the tree, decomposing Bacteria (*cst* and

cpr) are grouped together with other oxidising Bacteria (*tcx* and *tin*). The selectively open approach (middle tree) provides a better classification, with all the sulphide/sulfur oxidising Bacteria grouped together (*tcx* and *tin*; *pfo* and *pae*). *Desulfovibrio desulfuricans* (*ddn*), a sulfate-reducing Bacteria is also well discriminated, as well as the two Plant species (*zma* and *vvi*), which assimilate sulphate, and the two decomposing Bacteria species (*cpr* and *cst*). The fully open approach (bottom tree) does not provide a well defined grouping of organisms as in the selectively open approach (e.g., *cst* and *cpr* are not grouped together; *pae* is erroneously grouped with *ddn*). In this experiment the canonical selectively open approach shows the ability to distinguish organisms at a fine classification level. In this case it is able to discriminate organisms belonging to the Bacteria Reign, having different ecological roles within the biological sulfur cycle.
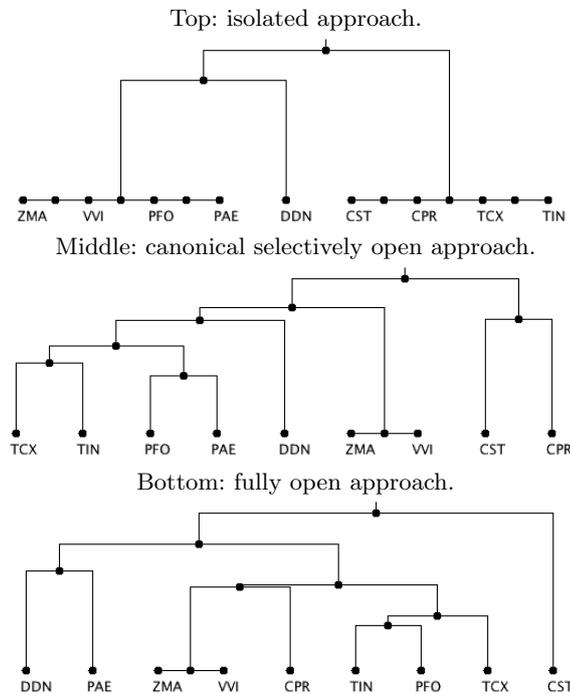
Fig. 5. Second experiment: UPGMA trees based on d_I wrt. *Sulfur metabolism*

Note that, by considering together all the organisms of the two experiments on the *Sulfur metabolism* the classification follows the same pattern: the more detailed classification is obtained by using the canonical selectively open approach, the isolated approach produces a rudimentary classification and the fully open approach does not produce a well defined classification. We preferred to anal-

yse the two groups separately in order to be able to show more precisely the granularity of the obtained classifications.

4.2   Carbon fixation pathway

In this experiment we consider the pathway *Carbon fixation in photosynthetic organisms*. This cycle consists of a series of reactions that lead to the biosynthesis of carbohydrates in the so called "dark phase" of photosynthesis. In most photosynthetic organisms this cycle is denominated *Calvin cycle* or *reductive pentose-phosphates cycle*. Some plants, in relation to environmental adaptations, exhibit specific variants of this cycle (C4 plants, CAM plants). A peculiarity of this pathway is that it is mainly composed by irreversible reactions.

We consider the organisms in Fig. 6 and we compute d_I, with the Sørensen index, for the isolated, selectively open and fully open approaches. Depending on the organism, the PN models contain at most 35 enzimes/reactions and 5 invariants in the isolated case, at most 52 enzimes/reactions and 42 invariants in the open case and at most 41 enzimes/reactions and 9 invariants in the canonical selectively open. In the selectively open approach the choice is among at most 34 compounds.

Code	Organism	Reign
gmx	*Glycine max*	*Plants, Eudicots*
pop	*Populus trichocarpa*	*Plants, Eudicots*
vvi	*Vitis vinifera*	*Plants, Eudicots*
osa	*Oryza sativa japonica*	*Plants, Monocots*
zma	*Zea mays*	*Plants, Monocots*
bdi	*Brachypodium distachyon*	*Plants, Monocots*
cre	*Chlamydomonas reinhardtii*	*Plants, green algae*
vcn	*Volvox carteri f. nagariensis*	*Plants, green algae*
npu	*Nostoc punctiforme*	*Bacteria*
acy	*Anabaena cilindrica*	*Bacteria*
oni	*Oscillatoria nigro-viridis*	*Bacteria*
mar	*Microcystis aeruginosa*	*Bacteria*

Fig. 6. Organisms for the experiment on the pathway *Carbon fixation in photosynthetic organisms*

Figure 7 shows the UPGMA tree corresponding to d_I in the isolated (top tree), canonical selectively open (middle tree) and fully open (bottom tree) approaches. Note that the isolated approach produces a rough classification, separating the bacteria from the other organisms. The selectively open approach permits the discrimination among the photosynthetic bacteria. The organism *vcn* is isolated due to its very simplified cycle, with a reduced number of intermediate products and enzymes involved. It is well-known that this genus is a very ancient group of organisms, originated from unicellular organisms. The

Top: isolated approach

Middle: canonical selectively open approach

Bottom: fully open approach
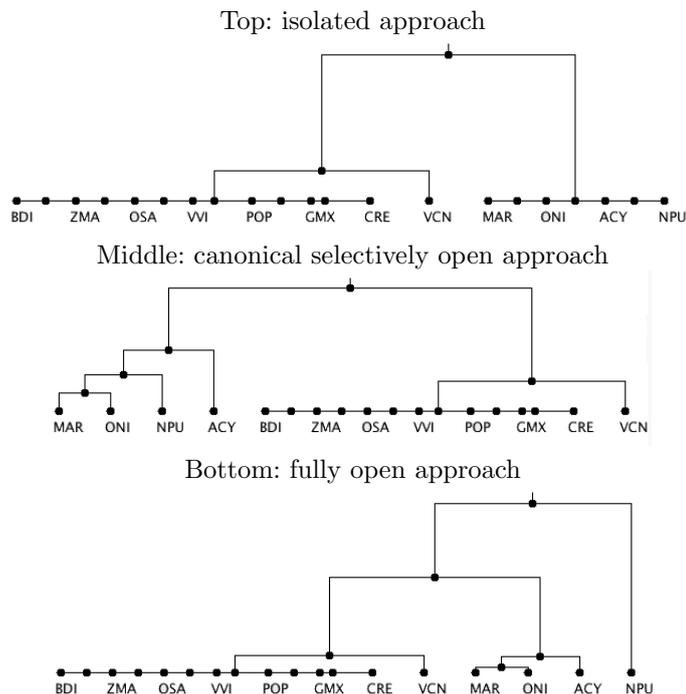
Fig. 7. Third experiment: UPGMA trees based on d_I wrt. *Carbon fixation in photosynthetic organisms*

fully open approach gives similar results to the selectively open one, but it does not group all the photosynthetic Bacteria together (*npu* is separated from all other organisms).

5   Conclusions

Metabolic pathways are subsystems of the full metabolic network. When constructing a model of a pathway this fact has to be taken into account and it requires some choices: the pathway can be represented in isolation or as a subsystem of the full network, interacting with its environment through some common compounds. In [9, 6] we proposed to use Petri net models for pathway comparisons based on reaction homology and functional aspects as captured by T-invariants. In this paper we consider the two modelling alternatives, isolated or open to any interaction with the environment, and conclude that neither of them is definitively better than the other. An isolated PN model guarantees correctness, namely minimal T-invariants of the pathway are minimal T-invariants of the full network, and it works well in most cases, but it captures only internal fluxes. Sometimes this is not sufficient to characterise the potential behaviours of the pathway, for example when a pathway has few internal cycles. In a fully

open PN model all potentially open places, namely the ones linking the pathway to the rest of the metabolic network and the ones which are sources or sinks, are indeed open. This approach may loose the correctness of T-invariants and in general it increases the size of the model without guaranteeing a better characterisation of the potential behaviours of the pathway. The information on the links among pathways becomes very relevant, even when they are not so important for distinguishing the functionalities associated to the pathway and, unfortunately, link informations is sometimes imprecise in KEGG. The most useful approach seems to be an intermediate one, in which a pathway is considered as an open subsystem, but the compounds on which the interaction takes place can be selected by the user. Our experience suggests a canonical choice of the open places which seems to produce the best results even in the absence of specific knowledge on a pathway, i.e., opening in input source places and opening in output sink places. We presented an extension of CoMeta, a tool for comparing metabolic pathways in different organisms, implementing our proposal. The tool retrieves the information about each selected pathway from the KEGG database, it determines the compounds which may potentially interact with the environment of the pathway and it offers to the user the possibility to select the interactions of interest, discriminating between input and output metabolites. CoMeta proposes the canonical choice in the selection (sources open in input and sinks open in output), but the user can freely add and delete open compounds. We presented some experiments which, although the work is still in a preliminary stage, suggest the appropriateness of this approach.

References

1. Kegg Markup Language manual. http://www.genome.ad.jp/kegg/docs/xml.
2. KEGG pathway database - Kyoto University Bioinformatics Centre. http://www.genome.jp/kegg/pathway.html.
3. Petri Net Markup Language. http://www.pnml.org.
4. Petri net tools. http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools.
5. 4ti2 team. 4ti2—a software package for algebraic, geometric and combinatorial problems on linear spaces. Available at www.4ti2.de.
6. P. Baldan, N. Cocco, F. Giummolé, and M Simeoni. Comparing Metabolic Pathways through Reactions and Potential Fluxes. In W. van der Aalst and A. Yakovlev, editors, *Special Issue of ToPNoC(Workshops and Tutorials)*, LNCS. Springer, 2013. Accepted for publication.
7. P. Baldan, N. Cocco, A. Marin, and M Simeoni. Petri nets for modelling metabolic pathways: a survey. *Natural Computing*, 9(4):955–989, 2010.
8. P. Baldan, N. Cocco, F. De Nes, M. Llabrés Segura, and M. Simeoni. MPath2PN - Translating metabolic pathways into Petri nets. In M. Heiner and H. Matsuno, editors, *BioPPN2011 Int. Workshop on Biological Processes and Petri Nets*, CEUR Workshop Proceedings, pages 102–116, 2011.
9. P. Baldan, N. Cocco, and M Simeoni. Comparison of metabolic pathways by considering potential fluxes. In M. Heiner and R. Hofestädt, editors, *BioPPN2012 - 3rd International Workshop on Biological Processes and Petri Nets, satellite event*

of Petri Nets 2012, Hamburg, Germany, June 25, 2012, CEUR Workshop Proceedings - Vol-852, pages 2–17. ceur-ws.org, 2012. ISSN 1613-0073, online: http://ceur-ws.org/Vol-852.

10. J. Esparza and M. Nielsen. Decidability issues for Petri Nets - a survey. *Journal Inform. Process. Cybernet. EIK*, 30(3):143–160, 1994.

11. E. Grafahrend-Belau, F. Schreiber, M. Heiner, A. Sackmann, B. H. Junker, S. Grunwald, A. Speer, K. Winder, and I. Koch. Modularization of biochemical networks based on classification of Petri net t-invariants. *BMC Bioinformatics*, 9(90), 2008.

12. S. Hardy and P.N. Robillard. Petri net-based method for the analysis of the dynamics of signal propagation in signaling pathways. *Bioinformatics*, 24(2):209–217, 2008.

13. M. Heiner and I. Koch. Petri Net Based Model Validation in Systems Biology. In *Petri Nets and Other Models of Concurrency - ICATPN 2004*, volume 3099 of *LNCS*, pages 216–237. Springer, 2004.

14. R. Hofestädt. A Petri net application of metbolic processes. *Journal of System Analysis, Modelling and Simulation*, 16:113–122, 1994.

15. I. Koch and M. Heiner. Petri nets. In B. H. Junker and F. Schreiber, editors, *Analysis of Biological Networks*, Book Series in Bioinformatics, pages 139–179. Wiley & Sons, 2008.

16. T. Murata. Petri Nets: Properties, Analysis, and Applications. *Proceedings of IEEE*, 77(4):541–580, 1989.

17. Saitou N. and Nei M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 1987.

18. M. Pedersen. Compositional definitions of minimal flows in Petri nets. In M. Heiner and A. M. Uhrmacher, editors, *Proceeedings of CMSB'08*, volume 5307 of *Lecture Notes in Computer Science*, pages 288–307. Springer, 2008.

19. V. N. Reddy, M. L. Mavrovouniotis, and M. N. Liebman. Petri net representations in metabolic pathways. In *ISMB93: First Int. Conf. on Intelligent Systems for Molecular Biology*, pages 328–336. AAAI press, 1993.

20. A. Schrijver. *Theory of linear and integer programming*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, 1999.

21. S. Schuster and C. Hilgetag. On elementary flux modes in biochemical reaction systems at steady state. *Journal of Biological Systems*, 2:165–182, 1994.

22. P. Sestoft. Programs for biosequence analysis. http://www.itu.dk/people/sestoft/bsa.html.

23. R. Sokal and Michener C. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:14091438, 1958.

24. T. Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biologiske Skrifter / Kongelige Danske Videnskabernes Selskabg*, 5(4):1–34, 1948.

25. T.T. Tanimoto. Technical report, IBM Internal Report, November 17 1957.

26. E. C. Webb. *Enzyme nomenclature 1992: recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the nomenclature and classification of enzymes.* San Diego: Published for the International Union of Biochemistry and Molecular Biology by Academic Press, 1992.

Semi-quantitative modelling of biological systems with extended Fuzzy 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. State of the art approaches for modelling biological systems can be classified as qualitative or quantitative. Qualitative models are relatively simple and require only basic knowledge of the system, but may only be used to get a rough image about the system's dynamics. On the other hand, quantitative models are able to mimic accurate dynamical properties of observed system, but require accurate kinetic data, which is often lacking. Existing fuzzy Petri net approaches only consider qualitative modelling. We propose a new, semi-quantitative approach based on fuzzy logic and extended Petri nets (PNs). Fuzzy logic allows us to linguistically describe biological processes, even if kinetic data are unknown. We present the details of fuzzification, defuzzification and the definition of IF-THEN rules on a model of degradation from a repressilator. We demonstrate how our approach circumvents the problem of missing kinetic parameters and how it can be used to augment existing quantitative methods such as models based on ordinary differential equations. By using fuzzy logic we were able to obtain comparable results to those of existing methods while using only rough estimation of kinetic parameter values, showing our method can be used for sufficient system analysis that helps with designing a novel biological system even when accurate kinetic data are unknown.

Keywords: biological switching systems, modelling biological systems, fuzzy logic, Petri nets, Fuzzy Petri nets

Qualitative modelling and analysis of Photosystem II[*]

Luboš Brim, Vilém Děd and David Šafránek

Systems Biology Laboratory, Faculty of Informatics, Masaryk University, Botanická 68a,
602 00 Brno, Czech Republic
{`brim,xded,xsafran1`}`@fi.muni.cz`

Abstract. In this paper, a work-in-progress aiming at qualitative modelling of photosynthesis at the mechanistic cellular level by means of Petri nets is described. Presented preliminary results concentrate on modelling and analysis of photosystem II, a crucial component of photosynthesis. By employing qualitative model checking combined with invariant analysis we obtain new insights into electron transfer mechanisms studied in photosystem II.

1 Introduction

Photosynthetic reactions of green plants and some bacteria take place on a membrane of specialized organelles called thylakoids where a lot of protein complexes reside. The most important are photosystem I, photosystem II and cytochrome complex b_6f. These complexes are responsible for absorbing light and they bind the light energy that is further used and transformed in later phases of photosynthesis. Products of these reactions are oxygen and energy bound in ATP and NADPH. The first complex in the whole reaction cascade is photosystem II which is responsible for absorbing light, splitting molecule of water and exciting electrons that further reduce subsequent complexes. [15]. Photosystem II is also considered to have main influence to measured data (especially light-induced fluorescence)[9].

Photosynthesis is often modelled using photochemical and redox reactions reflecting only some of measurable values, e.g., chlorophyll fluorescence, concentration of carbon dioxide or oxygen [15]. In the case of chlorophyll, fluorescence absorbance/emission is driven by femtosecond-scale reactions considered to be transitions of electrons. For these reactions, determination of kinetic rates is very difficult even impossible. Moreover, the reactions are influenced by many other factors such as value of pH or environment temperature.

Recently, some estimating methods have been already used for creation of quantitative models of photosystem II using differential equations [11]. There are also many studies trying to understand precise function of each component of photosystem II. The measured data are moreover commonly explained by very different theories [9]. Some models were analysed qualitatively [10], nevertheless, their development has been still based on incomplete quantitative knowledge.

The potential of using qualitative modelling in system biology is commonly under-rated because of the high level of abstraction. On the other hand, it has been shown many times that qualitative modelling is very powerful and contributive in cases of lack or incompleteness of data [21].

Petri nets are very powerful formalism which has been proved useful for modelling of complex biological systems and for providing some interesting hypothesis about their behaviour only on bases of knowledge of interactions between its components. Notable examples include discovery of various metabolite functions and components with very specific behaviour, e.g., cycling or accumulation [18]. For example, Petri nets have been successfully applied to models of metabolic cascade [6–8]. Though based on a purely qualitative model, the analyses have brought new insights. Other applications show the advantages of using Petri nets for high level modelling of multi-cellular organisms [3].

To the best of our knowledge, there exist only a few applications of Petri nets to modelling of photosynthesis. A general photosynthesis reaction is used as a simple example in [16] while a more serious case study is provided in [14] targeting non-photochemical quenching by means of metabolic P systems.

This paper describes a work-in-progress aiming at qualitative modelling of photo-synthesis at the mechanistic cellular level by means of Petri nets. Our first results concentrate on modelling and analysis of photosystem II, a crucial component of photosynthesis. For model development and analysis, we have employed the tools Snoopy [19] and Charlie [5]. We show that qualitative model checking is able to verify some commonly accepted theories about photosystem II.

2   Background and Problem Formulation
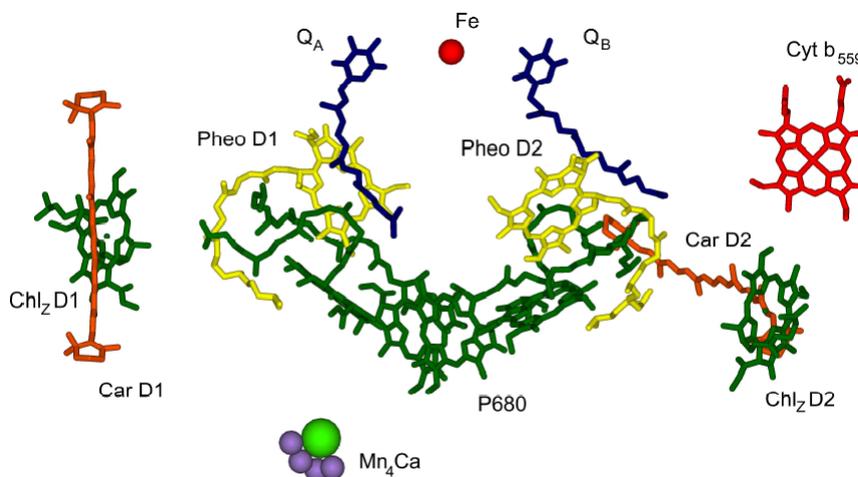
Fig. 1: Structure of photosystem II. Taken from [12].

Photosystem II can be divided into reaction center, oxygen evolving complex (OEC) and light harvesting complex (LHC). Reaction center is composed of *cytochrome* b_{559}, *chlorin complex* P680 and two other proteins very similar to each other(D_1 and D_2). Each of them binds other redox active components: *pheophytin* (*Pheo*), *quinone* (Q_a for D_1 , Q_b for D_2), *chlorophyll* (*Chl$_z$*), *carotene* (*Car*) and *tyrosine* (Y_Z for D_1, Y_D for D_1) [17, 20]. Because literature does not describe precise transitions inside complex P680, in this paper we neglect the fact that this complex consists of more than a single molecule. Very important is *oxygen-evolving complex* (OEC) bounded to P680. This complex is responsible for splitting the molecule of water and can be non-functional under some conditions [22].

The photosynthetic process begins with impact of photon to the LHC. Its energy is transited to the reaction center where it causes excitation of electron [13].

2.1 Basic electron path

The whole electron path begins with oxidation of complex of chlorins P680. Its electron reduces pheophytin Pheo. Then the electron continues to quinone Q_A and finally to quinone Q_B which is capable of carrying two electrons [20]. After receiving a second electron, quinone Q_B (now Q_B^{-2}) is now neutralized by two hydrogen cations from the stroma (the outside of thylakoid) to quinol $Q_B^{-2}H_2$ and leaves PSII into PQ-pool(space between PSII and complex b_6f containing 7-10 molecules of quinone/quinol). Into its place there comes a new neutral molecule of quinone Q_B. After leaving PSII, quinol transports electrons to the next protein complex (b_6f) and releases hydrogen into the thylakoid inner space turning again into quinone that is able to recycle back to PSII [9].
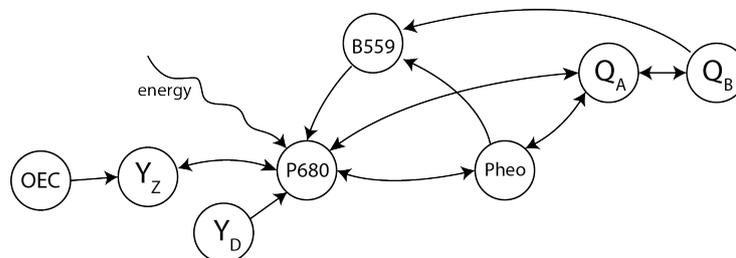
Fig. 2: Possible electron paths in Photosystem II.

When the P680 complex is oxidised, it receives one electron from *tyrosine Y_Z* (protein is situated between OEC and P680) which is subsequently reduced by one of the four electrons of OEC. After the fourth excitation of the electron from P680, the last electron is taken from OEC which is now capable to split two molecules of water. This reaction provides needed electrons and releases four hydrogen cations and two atoms of oxygen [20].

2.2   Alternative electron paths

Literature presents also other possible transitions of electron. In particular, there are possibilities of reverse transitions. Existence of those transitions is known between molecules of quinone Q_A and Q_B[11], Pheo and P680[23], Q_A and Pheo, P680 and Y_D, and finally Y_D and OEC[17]. There might be also a reverse reaction of binding (unbinding) of quinone Q_B(quinol $Q_B^{-2}H_2$) [11].

Except for those reverse transitions, literature describes also another direct transition of electrons. There is a possible transition from reduced Q_A^{-1} to oxidised P680 [20]. Other path is reduction of oxidised P680 by an electron from cyt b559 which then can receive an electron from a molecule of Qb reduced at least once [20]. Oxidised cyt b559 can receive an electron also from reduced pheophytin Pheo [1]. Such paths through cyt b559 are important during inhibition of OEC function (e.g., by low temperature or damage). As long as P680 is oxidised, it becomes a strong oxidation agent and may cause destruction of the whole system. The alternative path through cyt b559 prevents such events by periodic reduction of P680 [1, 22]. Finally, another possible transitions are oxidation of tyrosine Y_D (a protein analogous to Y_Z bound to D_2) and reduction of oxidised P680 where Y_D serves as the electron donor only once and during permanent illumination it stays in the oxidised state [22].

2.3   Existing models of photosystem II

It is worth noting that the literature shows only model structures created to better understand its content or those derived from quantitative models. There exist more models (described at structural level) differing in composition and interpretation of components and related transitions. This is the case mainly for OEC, pheophytin, cytochrome b559, and both molecules of tyrosine. OEC is modelled by four transitions between states S_i where $i \in \{0, 1, 2, 3\}$ refers to number of missing electrons. In some cases, a free electron state S_4 is used which is connected to S0 signifying the split of a molecule of water.

Structures are always described by a network of electron transitions, state transitions of a system, or by a reaction network of individual subunits.

Combination of these approaches was presented by Zhu et al. [23]. He divided photosystem into two subunits (P680/Pheo a Q_A/Q_B) and presented their state graphs. Moreover, his model contains OEC with S_i states and a molecule of tyrosine. All mentioned parts are connected by electron transitions.

Nedbal [15] showed very comprehensive structure composed by many state graphs of the subunit Y_Z/P680/Chl$_D$/Pheo/Q_A. These graphs were connected by transitions referring to change of four states of Q_B (Q_B, Q_B^{-1}, Q_B^{-2} and E representing absence of non-presence of Q_B). Finally, these graphs were obtained by interconnecting the subgraphs with reactions representing changes among S_i states of OEC.

Lazar [10] discussed several structures of quantitative models and the possibility of using first-order and second-order kinetics for modelling reactions of (un)binding Q_B and reactions between OEC and subunit $P680/Q_A/Q_B$. Subsequent analysis of these two approaches was focused mainly on quantitative features.

Other important model structures can be found also in [2, 20].

2.4 Problem specification

As mentioned above, existing models differ in the level of detail or in the modelling approach employed. Naturally, there is an inevitable need to unify the notation of models. There are also many biological questions to be solved. For example:

– Does the behaviour and features of models depend on presence/absence of particular photosystem II subunits?
– Are all the current theories about photosystem II valid?
– Can the model reach some final (stable) state?
– Are there any possible electron cycles which have not yet been observed?
– Are the existing models correct?

3 Results

For answering these questions we need to create a bunch of models reflecting the absence/presence of particular components. We perform static analysis (discovering invariants and their interpretation) and dynamic analysis using computational tree logic (CTL).

3.1 Model development

There exist several approaches to modeling photosystem II using Petri nets depending on interpretation of places and transitions. Some of them can be seen in Fig. 3. In this paper, we employ places to represent states of observed components of photosystem II and transitions to represent electron transitions. Only exception is the case of Q_B where places refer to states of binding site or bounded molecule of Q_B and transitions represent electron transfer ($Q_A \leftrightarrow Q_B^-$, $Q_B^- \leftrightarrow Q_B^{-2}$) or (un)binding of Q_B or Q_B^{-2}.

We consider a basic model including only irreversible reactions and quinone Q_B modeled without E state. Refining the model by all possible states and related reactions results in a combinatorial explosion. Number of possible models created on a basis of knowledge mentioned in Sec. 2 is 4320. For this work, we have selected 11 models:

A is basic model of PSII (see Fig. 4). It does not contain any irreversible reaction while it contains only complex P680, Q_A and Q_B without E state. This model reflects only the basic forward path of electron as mentioned in [20].
B is an extension of the model A. There are considered molecules of pheophytin and tyrosine in the basic path. Moreover, Q_B is modelled with state E.
C represents the model B extended by the reverse electron flow from quinone Q_A to P680.
D represents model B extended by molecule of tyrosine Y_D.
E refers to model C extended by molecule of cytochrome B599 and related transitions.
F is a model created by the union of models E and D. It makes the complete model without any reversible transitions.
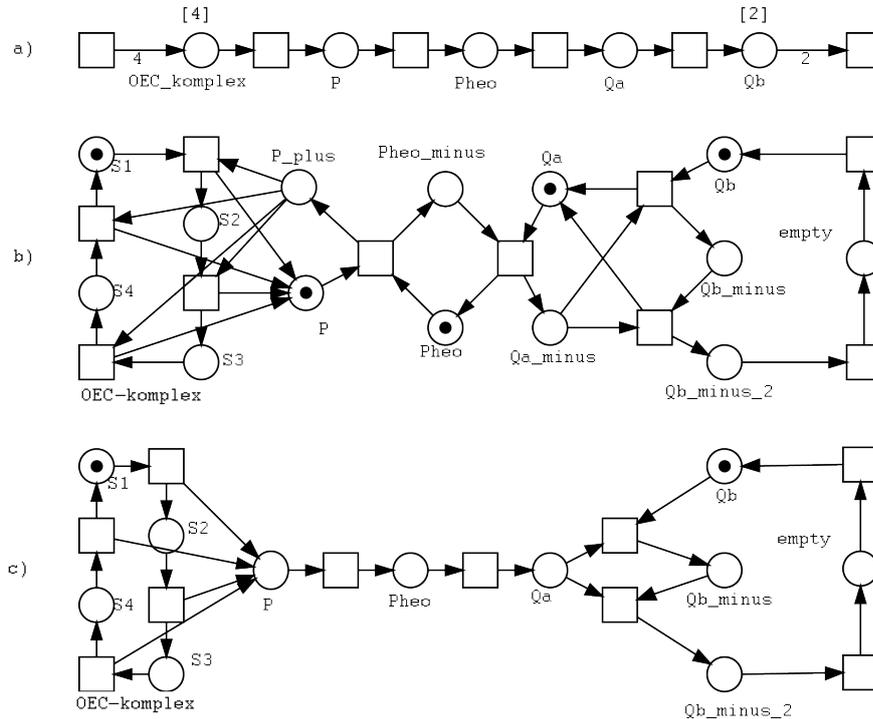
Fig. 3: The figure shows different approaches to modelling of photosystem II using Petri nets. In (a), places represent the component itself. In this case, the maximal number of tokens per place must be specified (normally it is 1 but Q_B can carry 2 and OEC up to 4 electrons). In (b), places represent states of individual components. Model c is created as combination of models a and b.

G represents model B extended by reverse transitions between Y_Z and P680, P680 and Pheo, Pheo and Q_A.

H represents model F extended by reverse transitions between Q_A and Q_B.

I is a model created by the union of models D and H.

J refers to model H extended by reverse transitions between states Q_B^{-2} and E (E and Q_B) which represents possible reverse binding of Q_B^{-2} (reverse unbinding of Q_B).

K is the complete and the most complex model containing all components and transitions mentioned in literature (see Fig. 4).

Moreover, we have created all models in two forms. Variant (a) (e.g., K_a) denotes models containing OEC. Models marked (b) refer to the system with a non-functional OEC.
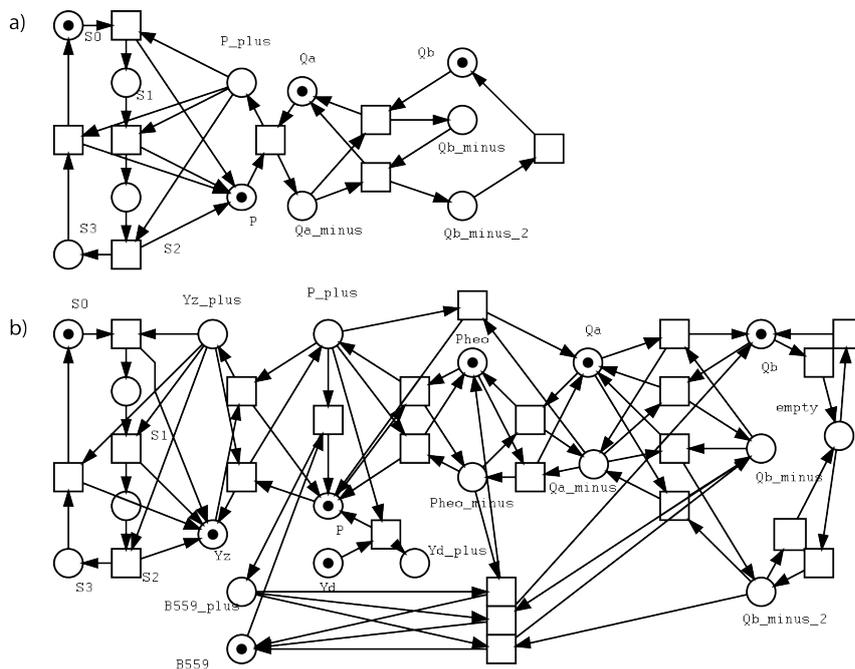
Fig. 4: The figure shows the simplest (a) and the most complex (b) model with the respective initial marking.

3.2 Initial marking construction

Before the analysis we need to specify what is the initial state (marking) of our system. It would not be necessary for models with the state space making a single strongly connected component. In this case, all states are reachable regardless of current state of model. Nevertheless, there exist models that do not satisfy that. The connected component in the state space can be interpreted as a permanent outflow/inflow of the electron from/into some system subunit. For the whole analysis we expect all components in neutral state and OEC in the state S0 (with four electrons present). Employing these constraints, the initial marking M_0 is the following:

$P680$	$P680^+$	$Pheo$	$Pheo^-$	Q_A	Q_A^-	Q_B	Q_B^-	Q_B^{-2}	E	Y_Z	Y_Z^-	S_0	S_1	S_2	S_3	Y_D	Y_z^-	B_{559}	B_{559}^-
1	0	1	0	1	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0

3.3   Invariants

P-invariants determine sets of places where the total number of tokens is conserved. For our models this can be interpreted as preserving all of redox active components in the model, e.g., P-invariants show that all components were present in the model during the process execution.

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
$P \rightarrow Pheo$	1	0	0	0	0	0	0	1	1	1	1	4
$Pheo \rightarrow P$	1	0	0	0	0	0	0	0	0	0	0	0
$Pheo \rightarrow Q_A$	0	1	0	0	0	0	0	0	1	1	1	4
$Q_A \rightarrow Pheo$	0	1	0	0	0	0	0	0	0	0	0	0
$Q_A \rightarrow Q_B$	0	0	1	0	0	0	0	0	0	1	0	2
$Q_A \rightarrow Q_B^-$	0	0	0	1	0	0	0	0	0	0	1	0
$Q_B^- \rightarrow Q_A$	0	0	1	0	0	0	0	0	0	0	0	0
$Q_B^{-2} \rightarrow Q_A$	0	0	0	1	0	0	0	0	0	0	0	0
$Q_B^{-2} \rightarrow E$	0	0	0	0	0	1	0	0	0	0	0	2
$E \rightarrow Q_B^{-2}$	0	0	0	0	0	1	0	0	0	0	0	0
$E \rightarrow Q_B$	0	0	0	0	0	0	1	0	0	0	0	2
$Q_B \rightarrow E$	0	0	0	0	0	0	1	0	0	0	0	0
$Y_Z \rightarrow P$	0	0	0	0	1	0	0	0	0	0	0	4
$P \rightarrow Y_Z$	0	0	0	0	1	0	0	0	0	0	0	0
$S0 \rightarrow Y_Z$	0	0	0	0	0	0	0	0	0	0	0	1
$S1 \rightarrow Y_Z$	0	0	0	0	0	0	0	0	0	0	0	1
$S2 \rightarrow Y_Z$	0	0	0	0	0	0	0	0	0	0	0	1
$S3 \rightarrow Y_Z$	0	0	0	0	0	0	0	0	0	0	0	1
$Y_D \rightarrow P$	0	0	0	0	0	0	0	0	0	0	0	0
$Q_A \rightarrow P$	0	0	0	0	0	0	0	0	1	0	0	0
$Pheo \rightarrow B559^+$	0	0	0	0	0	0	0	1	0	0	0	0
$Q_B^- \rightarrow B559^+$	0	0	0	0	0	0	0	0	0	1	0	0
$Q_B^{-2} \rightarrow B559^+$	0	0	0	0	0	0	0	0	0	0	1	0
$B559 \rightarrow P$	0	0	0	0	0	0	0	1	0	1	1	0

Table 1: Table shows which T-invariants were (1) or were not (0) found in particular models. All related models of model group *b* contain the same invariants with the only exception of t_{12}.

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
A	0	0	0	0	0	0	0	0	0	0	0	1
B	0	0	0	0	0	0	0	0	0	0	0	1
C	0	0	0	0	0	0	0	0	0	1	0	1
D	0	0	0	0	0	0	0	0	0	0	0	1
E	0	0	0	0	0	0	0	1	1	1	1	1
F	1	1	1	1	1	0	0	1	1	1	1	1
G	1	1	0	0	1	0	0	0	0	0	0	1
H	1	1	1	1	1	0	0	0	0	0	0	1
I	1	1	1	1	1	0	0	0	0	0	0	1
J	1	1	1	1	1	1	1	0	0	0	0	1
K	1	1	1	1	1	1	1	1	1	1	1	1

Table 2: Rows refer to the models described in the first column. Numbers stand for numbers of occurrences of the transition required to revisit the original state. T-invariants are specified in columns.

T-invariants refer to multi sets of reactions which can be interpreted as sequences of reactions the triggering of which will not change the system state. All T-invariants found are shown in Tab. 1. Invariants t_1 - t_7 refer respectively to triggering single reactions and their related reverse counterparts. These t-invariants are usually called trivial invariants. Other invariants (except t_{12}) describe more complicated cycles of electrons inside photosystem II. The most complex is invariant t_{12} expressing the whole set of reactions needed for a sequence of two subsequent unbinding events of Q_B^{-2}, e.g., sending 4

electrons to the next photosynthetic system. In Tab. 2, T-invariants are shown mapped to the respective models.

3.4 Analysis by CTL

We employed model checking to study individual dynamic phenomena described in terms of formulae of the computation tree logic (CTL) [4]. The most important result we have obtained is given in Fig. 5. It explains relations between created models and shows the validity of individual CTL formulae.
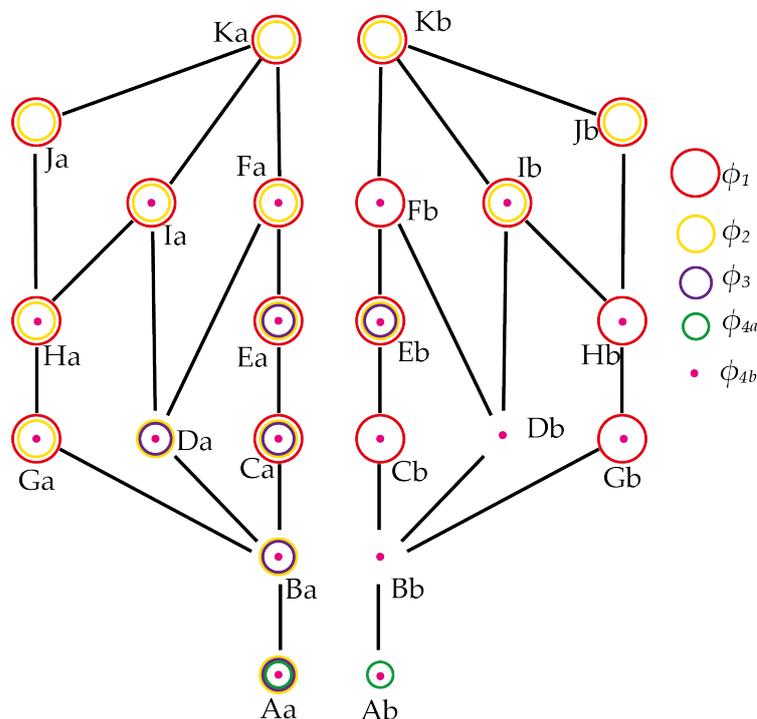
Fig. 5: The graphs show an overview of the model checking results of the models described in Section 3.1. Relation between models X and Y (considering X positioned above Y) naturally means that X is a refinement of Y (it contains a superset of all components in X). Generally speaking the models on the top of the graph are the most detailed and complex. On the other hand, the models on the bottom are the simplest ones. Transitive and reflexive connections are missing. For simplification, there are missing also connections between models containing OEC (group *a*) and models without OEC (group *b*). Colours of individual circles appearing in nodes refer to validity of respective CTL formulae. The formula $\varphi = 1$ for a model X $\Leftrightarrow$ the respective node is coloured by the color assigned to φ.

Formula ϕ_1 describes the possibility of quinone Q_B reduction that allows unbinding of Q_B from the system thus making other parts of photosynthetic reactions possible.

$$\phi_1 = EG(EF(P \wedge Q_A \wedge Q_B) \wedge !(Q_B^{-2})) \tag{1}$$

Formally, ϕ_1 requires there exists a path in the reachability graph such that P/Qa/Qb is always reachable and Qb never reaches the twice reduced state. Validity of ϕ_1 means that Qb is never left unbound from PSII.

The formula has the same truth value for all corresponding models from both groups. Thus we can assume that the expressed property is independent of OEC function.

The formula is false only for models where the electron can cycle through reversible transitions without reaching Q_B^{-2}. In other words, presence of any other T-invariant than t_{12} violates the formula.

Formula ϕ_2 verifies if there exists for every state a possibility to reach a state where P680 is neutralized. This can be interpreted as the possibility to avoid the irreversible photosystem damage caused by long time oxidation of P680.

$$\phi_2 = AG(P^+ \rightarrow EF(P)) \tag{2}$$

Formula ϕ_2 is true for all models from group *a* where P680 can be always reduced by an electron from OEC. For models A_b, B_b a D_b the formula is false because there are no reverse transition capable of reducing P680. In terms of T-invariants, the models do not contain any other T-invariant than only t_{12}.

Despite of the fact that models C_b, G_b and H_b enable the excited electron to return, a specific state can be reached where the electron cannot be returned back. If is P680 after first excitation reduced by electron from Y_Z and this electron is also excited and transited to quinone Q_B, both of these electrons can be taken away and P680 will stay oxidised. Model E_b (I_b) contains in addition cyt b559 (tyrosine Y_D) capable of donating one electron to the system. The number of electrons is therefore 3 and although two of them can leave the system, one will be always able to cycle and periodically reduce P680.

Interesting is model F_b which contains both cyt b559 and tyrosine Y_D. Thus double oxidation of Q_B can be performed two times which would lead to exhaustion of system and permanently oxidised P680.

Formula is also true for models J_b and K_b where it is always possible to reduce P680 by a sequence of reverse transitions even from the PQ-pool.

Formula ϕ_3 describes cyclic behaviour of P680. Verifying the assumption that P680 is neutralized or oxidized, it will always get to the state where is oxidised or neutralized, respectively. In other words, the property ensures that the system does not stack with P680 permanently oxidised or neutralized which would lead to its disfunction caused by the absence of electron excitation events.

$$\phi_3 = AG((P \rightarrow AF(P^+)) \wedge (P^+ \rightarrow AF(P))) \tag{3}$$

The formula is false for models containing any other reverse reaction than the one related with complex P680, in other words, it is false for all models containing any

t-invariant not related with P680. System in these reaction can cycle. This holds also for group *b*. Moreover this formula is true only in case when ϕ_2 holds, i.e., a final state satisfying P680 cannot be never reduced can be reached. Only model E_b does not have any final state and thus the formula is true. The reason is that the model contain only reverse reactions related to P680 ensuring its periodic reduction.

Formula ϕ_4 expresses the condition that if P680 is oxidised and Q_A and Q_B are neutralized, there is only one possible action - neutralization of $P680^+$.

$$\phi_{4a} = AG((P^+ \wedge Q_A \wedge Q_B) \to AX((P \wedge Q_A \wedge Q_B))) \tag{4}$$

To make the property clearer we define another formula ϕ_{4a} varying for different models in order to ensure that all oxidizable (reducible) components are oxidized (reduced), respectively. This state displays an absolute lack of electrons in the system. For example, for the model including molecules Y_Z and Y_D the formula has the following form:

$$\begin{aligned}\phi_{4b} = &AG((P^+ \wedge Q_A \wedge Q_B \wedge Y_D{}^+ \wedge Y_Z{}^+) \to \\ &AX((P^+ \wedge Q_A \wedge Q_B \wedge Y_D{}^+ \wedge Y_Z)))\end{aligned} \tag{5}$$

Formula ϕ_{4a} is true only for basic models A_a and A_b. In other models, there is included some other component capable of another step than reducing oxidised P680. More specific formula ϕ_{4b} ensures that it will not happen. It is false only for models J_a, K_a, J_b and K_b which contains t-invariant t_7 that means unbinding of neutral quinone Q_B ($Q_B \to E$).

3.5 Discussion

Petri nets showed up as a feasible modelling formalism suitable for our problem. All models, even the most complex ones, are simple and transparent.

Considered binary interpretation of places and transitions seems to be optimal, because other interpretations would not be so explicit about states of components and it would be necessary to define maximal number of tokens present in every place.

T-invariants provided a simple way for representation and specification of non-trivial transitions needed for revisiting the initial state.

Formulas used for expressing observed features were also very transparent and easily understandable. Nevertheless, they proved some behaviour of system is possible despite it is actually very improbable, even impossible. An example is infinite cycling of electron using a forward reaction and its respective reverse direction.

An interesting result is that some formulas have the same truth value for corresponding model variants (*a*) and (*b*). Thus we can assume that expressed features do not dependent on the functionality of OEC. A special attention should be paid to models without OEC because some of them, especially the very comprehensive model F_b, can reach critical state with permanently oxidised P680. Special attention is needed during formulas formulation, e.g., formula ϕ_{4a}. It is valid only for the simplest models in the hierarchy. An alternative formula ϕ_{4b} has to be employed for other models.

Details on models, formulas and (im)possibility of verifying the requested features should be an objective of further discussions with scientists interested in photosystem II.

4   Conclusions and Future Work

In this paper, known facts about photosystem II have been summarized and we have shown preliminary results on its qualitative modelling and analysis. Created models have been compared wrt validity of CTL formulas representing crucial features of the photosystem. It has been shown that some features do not depend on the functionality of OEC. Moreover, it has been shown that some models can reach the final state in which the complex P680 is oxidised. Such a scenario would lead to disfunction and irreversible damage of photosystem II.

For future work we consider addition of formulas reflecting other features of the system. Furthermore, our research aims at targeting other protein complexes in the photosynthesis chain. Their models can be analysed individually and also integrated into a complex photosynthesis model. Obtained information about models can be also validated in a real system (if it is possible) or compared to existing quantitative models. Other direction could be an implementation of software capable of creation and analysis of all 4320 possible models of photosystem II and visualization of validity of specified formulas in similar way as in the graph shown in Fig. 5.

References

1. Barber, J., Rivas, J.D.L.: A functional model for the role of cytochrome b559 in the protection against donor and acceptor side photoinhibition. Biochemistry 90(23), 42– 46 (1993)
2. Belyaeva, N.E., Pahchenko, V.Z., Renger, G., Riznichenko, G.Y., Rubin, A.B.: Application of a photosystem II model for analysis of fluorescence induction curves in the 100 ns to 10 s time domain after excitation with a saturating light pulse. Biophysics 51, 860–872 (2006)
3. Bonzanni, N., Krepska, E., Feenstra, K.A., Fokkink, W., Kielmann, T., Bal, H., Heringa, J.: Executing multicellular differentiation: Quantitative predictive modelling of *C.elegans* vulval development. Bioinformatics 25(16), 2049–2056 (2009)
4. Clarke, E.M., Emerson, E.A., Sistla, A.P.: Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Trans. Program. Lang. Syst. 8(2), 244– 263 (Apr 1986), http://doi.acm.org/10.1145/5397.5399
5. Franzke, A.: Charlie 2.0 – a multi-threaded Petri net analyzer. Diploma thesis, BTU Cottbus, Dep. of CS (December 2009)
6. Gilbert, D., Heiner, M.: From petri nets to differential equations - an integrative approach for biochemical network analysis. In: ICATPN'06. pp. 181–200. LNCS, Springer (2006)
7. Hardy, S., Robillard, P.N.: Pn: Modeling and simulation of molecular biology systems using petri nets: modeling goals of various approaches. J Bioinform Comput Biol 2004, 2–4 (2004)
8. Heiner, M., Gilbert, D., Donaldson, R.: Petri nets for systems and synthetic biology. In: Proceedings of the Formal methods for the design of computer, communication, and software systems 8th international conference on Formal methods for computational systems biology. pp. 215–264. SFM'08, Springer (2008)
9. Lazár, D.: Modelling of light-induced chlorophyll fluorescence rise (O-J-I-P transient) and changes in 820 nm-transmittance signal of photosynthesis. Photosynthetica 47, 483–498 (2009)
10. Lazár, D., Jablonský, J.: On the approaches applied in formulation of a kinetic model of photosystem II: Different approaches lead to different simulations of the chlorophyll a fluorescence transients. J Theor Biol. 257 (2009)
11. Lazár, D., Schansker, G.: Models of chlorophyll a fluorescence transients. In: Photosynthesis in silico: understanding complexity from molecules to ecosystems, Advances in Photosynthesis and Respiration, vol. 29, pp. 85–123. Springer (2009)

12. Litvín, R.: Photochemistry of photosystem II reaction centre: Alternative electron pathways. Ph.D. thesis, University of South Bohemia, Faculty of Science (2009)
13. Lokstein, H., Hrtel, H., Hoffmann, P., Woitke, P., Renger, G.: The role of light-harvestingcomplexii in excessexcitationenergy dissipation: An in-vivofluorescencestudy on the origin of high-energyquenching. Journal of Photochemistry and Photobiology 26(2), 175–184 (1994)
14. Manca, V., Pagliarini, R., Zorzan, S.: A photosynthetic process modelled by a metabolic p system. Natural Computing 8(4), 847–864 (Dec 2009)
15. Nedbal, L., vCervený, J., Schmidt, H.: Scaling and integration of kinetic models of photosynthesis: Towards comprehensive e-photosynthesis. In: Laisk, A., Nedbal, L., Govindjee (eds.) Photosynthesis in silico: understanding complexity from molecules to ecosystems, Advances in Photosynthesis and Respiration, vol. 29, pp. 17–29. Springer (2009)
16. Pedersen, M.: Compositional definitions of minimal flows in petri nets. In: Proceedings of the 6th International Conference on Computational Methods in Systems Biology. pp. 288–307. CMSB '08, Springer-Verlag (2008)
17. Rappaport, F., Guergova-Kuras, M., Nixon, P.J., Diner, B.A., Lavergne, J.: Kinetics and pathways of charge recombination in photosystem II. Biochemistry 41(26), 8518–8527 (2002)
18. Reddy, V.N., Liebman, M.N., Mavrovouniotis, M.L.: Qualitative analysis of biochemical reaction systems. Computers in Biology and Medicine 26(1), 9 – 24 (1996)
19. Rohr, C., Marwan, W., Heiner, M.: Snoopy–a unifying petri net framework to investigate biomolecular networks. Bioinformatics 26(7), 974–975 (2010)
20. Ruffle, S., Sayre, R.: Functional analysis of photosystem II. In: Rochaix, J., Goldschmidt-Clermont, M., Merchant, S. (eds.) The Molecular Biology of Chloroplasts and Mitochondria in Chlamydomonas, Advances in Photosynthesis and Respiration, vol. 7, pp. 287–322. Springer (2004)
21. Schaub, M., Henzinger, T., Fisher, J.: Qualitative networks: a symbolic approach to analyze biological signaling networks. BMC Systems Biology 1(1), 1–21 (2007), http://dx.doi.org/10.1186/1752-0509-1-4
22. Shinopoulos, K.E., Brudvig, G.W.: Cytochrome b559 and cyclic electron transfer within photosystem II. Biochimica et Biophysica Acta (BBA) - Bioenergetics 1817(1), 66 – 75 (2012)
23. Zhu, X.G., Govindjee, Baker, N., deSturler, E., Ort, D., Long, S.: Chlorophyll a fluorescence induction kinetics in leaves predicted from a model describing each discrete step of excitation energy and electron transfer associated with photosystem II. Planta 223, 114–133 (2005)

Petri net based modelling and simulation of p16-Cdk4/6-Rb pathway

Nimet İlke Çetin,[1] Rza Bashirov,[1] Şükrü Tüzmen[2]

[1] Department of Applied Mathematics and Computer Science
[2] Department of Biological Sciences,
Eastern Mediterranean University, Famagusta, North Cyprus, Mersin-10, Turkey
{ilke.cetin;rza.bashirov;sukru.tuzmen}@emu.edu.tr

Abstract. Tumor suppressor gene p16 is of utmost interest in investigation of signal transduction pathways due to its gatekeeper role at the G1/S checkpoint of the cell cycle. Defects in p16 result in uncontrolled cell division which leads to progression of malignancy in an organism. In the present research we focus on p16-Cdk4/6-Rb pathway which is a cornerstone of G1 phase of the cell cycle. We implement Pet net formalism and Cell Illustrator software tool to create model of p16-Cdk4/6-Rb pathway and perform a series of simulations to validate the model.

Keywords: Replicative senescence, Cell cycle, p16-Cdk4/6-Rb pathway, Hybrid functional Petri net, Cell Illustrator

1 Introduction

1.1 Biological context

Cell division is a fundamental biological process that is essential to continuity of all living organisms. Cell replication or growth is controlled by a complex network of signals, that control the cell cycle. During the cell cycle cells grow to twice their size, copy their chromosomes, and divide into two new cells. The cell cycle is composed of four distinct phases: G1-phase (gap 1), S-phase (synthesis), G2-phase (gap 2) and M-phase (mythosis) [14]. Cell cycle checkpoints are used between neighboring phases to monitor and regulate the progress of the cell cycle. A cell cannot proceed to the next phase until otherwise checkpoint requirements have been met.

Tumor suppressor gene p16 plays important role in regulating cell grows and division at checkpoint G1/S [34]. The p16 gene is major tumor suppressor gene that is responsible for replicative senescence. Cell division is not an infinitely continuous process as cells undergo a finite number of cumulative population doublings [17]. Most human normal cells permanently stop dividing after a 50-75 cell divisions and enter a state termed cellular or replicative senescence [17]. Most tumors contain cells that appear to have bypassed this limit and evaded replicative senescence. Immortality, or even an extended replicative lifespan, greatly increases susceptibility to malignant progression because it permits

the extensive cell divisions needed to acquire successive mutations. Thus, cellular senescence may act as a barrier to cancer and play an important role in tumor suppression [8]. Inactivation of tumor suppressor gene p16, which in fact keeps track of replicative senescence, results in uncontrolled cell division, which leads to cancer [4].

During G1 phase, proteins Cdk4 and Cdk6 form complex with protein CycD, which in turn phosphorylates the Rb protein family. When Rb is phosphorylated by Cdk4/6 it loses its function and releases its target, the E2F family transcription factors, resulting in the initiation of DNA replication [31, 32]. Otherwise Rb inhibits transcription factor E2F [36]. E2F is a transcription factor which initiates transcription of genes required for S phase [5]. In the case of malignant progression action of p16 inhibits binding of Cdk4/6 with CycD which leaves Rb, and other Rb related proteins [25, 35]. The p16 targets Cdk4 and Cdk6, rather than the CycD, and actually competes with CycD for Cdk binding. Binding of p16 results in changes in conformation of Cdk proteins so that they can no longer bind CycD [29]. The p16 may also deactivate preassembled Cdk4/6_CycD complex blocking their function [29].

The proteins and their complexes are involved in natural degradation. In addition, the CycD protein is also tightly regulated by ubiquitin-dependent degradation [2, 13, 23].

1.2   Related work

Over the past two decades considerable efforts have been directed towards Petri net based investigation of biological systems. A series of biological phenomena modelled and simulated in terms of Hybrid Functional Petri Net (HFPN) include molecular interactions in the flower developmental network of Arabidopsis thaliana [19], lac operon gene regulatory mechanism in the glycolytic pathway of Escherichia coli [9], cell fate specification during Caenorhabditis elegans vulval development [21], antifolate inhibition of folate metabolism [3], validation of transcriptional activity of the p53 [12], glycolytic pathway controlled by the lac operon gene [11], apoptosis signalling pathway [26], circadian rhythms of Drosophila [26], switching mechanism of λ phage [26].

In [16] the authors proposed a hybrid Petri net model of cell cycle. The model comprises both stochastic and deterministic approaches. In this model, stochasticity is used to capture change of the cell size and effect of noises. This model is centered upon interactions between complexes CycB-Cdk1, Cdh1-APC, and monomers Cdc14 and Cdc20 [33]. The study expands macro-level understanding of cell cycle control. However, this study does not provide any insights into understanding quantitative behavior of biological components involved in the cell cycle regulation. Indeed, cell cycle regulation is a complex biological mechanism that consists of hundreds of biological components, processes and pathways. It is hard if not impossible to perform quantitative analysis of cell cycle regulation based on modest size model.

1.3   Contributions

The present research exploits HFPN to create a model of p16-Cdk4/6-Rb pathway, which is a cornerstone of cell cycle regulation at G1/S checkpoint. We combine biological facts described in Subsection 1.1 and quantitative knowledge on reaction rates provided in [11, 12] in a HFPN model. Then we use Cell Illustrator software to perform simulation-based model checking to validate the HFPN model. Simulation-based model checking in general provides interesting biological insights which could be used for future wet-lab experiments [22]. Once the model validated it can be used for obtaining broader understanding of cell cycle regulation.

The manuscript is organized as follows. Section 2 provides a succinct background on HFPN. In Section 3 we develop a HFPN model of p16-Cdk4/6-Rb pathway, and explain relationship between HFPN objects and their biological counterparts. Section 4 presents and analyzes the simulation results. Finally, conclusions are outlined in Section 5.

2   Hybrid Functional Petri Net

Biological systems are characterized by interaction of different structured processes. A continuous process is used to represent a biological reaction, at which a real number called the reaction speed or reaction rate is assigned as a parameter. Concentration change of the biological components or substrates after the biological reaction is completed is also represented as a real number. Promotion/inhibition mechanisms and checking for presence of this or that biological component or phenomenon are typical discrete processes. Change of quantity in a discrete process is usually expressed by integers or Boolean values.

When modelling biological pathways it is desirable to use a modelling framework that combines both continuous and discrete processes. Related software tools are consequently expected to comprise different structured data types including real numbers, integers, Boolean, etc. HFPN [26, 21] was originally proposed for modelling and simulating biological systems employing hybrid structure and dedicated software Cell Illustrator [11, 27] provides suitable platform for visualization and simulation of HFPN models.

While modelling with HFPN, the researchers prefer to use terminology that is slightly different than the traditional one [28]. In order to ensure compliance with the biological content Petri net objects such as place, transition, arc and token are respectively renamed as entity, process, connector and quantity. To increase the readability of the paper below we provide a brief description of HFPN model elements. For more detailed information on this issue the readers are referred to [10].

In context of HFPN an *entity* is an abstract object that represents biological component or substrate such as DNA, mRNA, protein, enzyme, complex of proteins, etc. Each entity is assigned a numeric value called *quantity*, which stands for concentration of related substrate. Variables are used to carry concentration values. A *process* is another abstract object that is used to model

biological reaction or phenomenon like transcription, translation, binding, nuclear export/import, ubiquitination and natural degradation. A process defines the change rate of entity value and establishes interactions among entities. Rate of change is expressed as a formula.

Table 1. Correspondence between biological components and HFPN entities.

Entity name	Entity type	Variable	Initial value	Value type
p16mRNA	Continuous	$m1$	0	Double
p16(C)	Continuous	$m2$	0	Double
p16(N)	Continuous	$m3$	0	Double
CDK4mRNA	Continuous	$m4$	0	Double
CDK4(C)	Continuous	$m5$	0	Double
CDK4(N)	Continuous	$m6$	0	Double
CDK6mRNA	Continuous	$m7$	0	Double
CDK6(C)	Continuous	$m8$	0	Double
CDK6(N)	Continuous	$m9$	0	Double
CycDmRNA	Continuous	$m10$	0	Double
CycD(C)	Continuous	$m11$	0	Double
CycD(N)	Continuous	$m12$	0	Double
CDK4_CDK6	Continuous	$m13$	0	Double
CDK4_CDK6_CycD	Continuous	$m14$	0	Double
Phosphate	Continuous	$m15$	100	Double
RB_DP_E2F	Continuous	$m16$	100	Double
nr_div	Discrete	$m17$	0	Integer
RB_P	Continuous	$m18$	0	Double
DP_E2F	Continuous	$m19$	0	Double
Mutation	Generic	$m20$	true/false	Boolean
p16mutated	Continuous	$m21$	0	Double
G1-dysfunction	Generic	$m22$	true/false	Boolean
p16_CDK4/6(N)	Continuous	$m23$	0	Double
p16_CDK4/6(C)	Continuous	$m24$	0	Double
Ubiquitin	Continuous	$m25$	100	Double
CycD[Ub]	Continuous	$m26$	0	Double
S_phase_genes	Continuous	$m27$	0	Double

The entities and processes are classified as being discrete, continuous and generic. A *discrete entity* is quantified by integers. A *discrete process* causes integer-valued change of a quantity. A *continuous entity* is quantified by real numbers, and consequently *continuous process* causes change of a quantity according to reaction rate formula, which is also represented by real numbers. A *generic entity* contains structured data type composed of different structured data such as Boolean, double and integer. A *generic process* handles structured data assigned to associated entities. In HFPN we distinguish between process connector, inhibitory connector and association connector. A *process connector* is adjacent

from input entity to a process or from process to its output entity. Weight parameter is used to specify an activation threshold. Process connectors ensure flow of tokens in the model. An *inhibitory connector* is used to inhibit a process. Inhibitory connectors are integral elements of biological models with competing processes. An *association connector* establishes adjacency relation between specified entity and process under circumstance that occurrence of related process does not cause concentration change. An association connector is often used in modelling of enzymatic and catalytic reactions.

Table 2. Correspondence between biological phenomena and HFPN processes.

Biological phenomenon	Process	Process type	Process rate
Transcription of p16mRNA	$T1$	Continuous	1
Translation of p16	$T2$	Continuous	m1*0.1
Nuclear import of p16	$T3$	Continuous	m2*0.1
Transcription of CDK4mRNA	$T4$	Continuous	1
Translation of CDK4	$T5$	Continuous	m4*0.1
Nuclear import of CDK4	$T6$	Continuous	m5*0.1
Transcription of CDK6mRNA	$T7$	Continuous	1
Translation of CDK6	$T8$	Continuous	m7*0.1
Nuclear import of CDK6	$T9$	Continuous	m8*0.1
Transcription of CycDmRNA	$T10$	Continuous	1
Translation of CylinD	$T11$	Continuous	m10*0.1
Nuclear import of CycD	$T12$	Continuous	m11*0.1
Binding of CDK4 and CDK6	$T13$	Continuous	m6*m9*0.01
Binding of CDK4_CDK6 and CycD	$T14$	Continuous	m12*m13*0.01
Phosphorylation of RB	$T15$	Continuous	m14*m15*m16*0.1
Mutation of p16	$T16$	Generic	m2*0.1
Binding of p16(N) and CDK4_CDK6	$T17$	Continuous	m3*m13*0.01
Nuclear export of p16_CDK4_CDK6	$T18$	Continuous	m23*0.1
Ubiquitination of CycD	$T19$	Continuous	m11*m25*0.01
Degradation of CycD[Ub]	$T20$	Continuous	m26*0.5
Transcription of S phase genes	$T21$	Continuous	m19*1

Table 3. Natural degradations in the HFPN model.

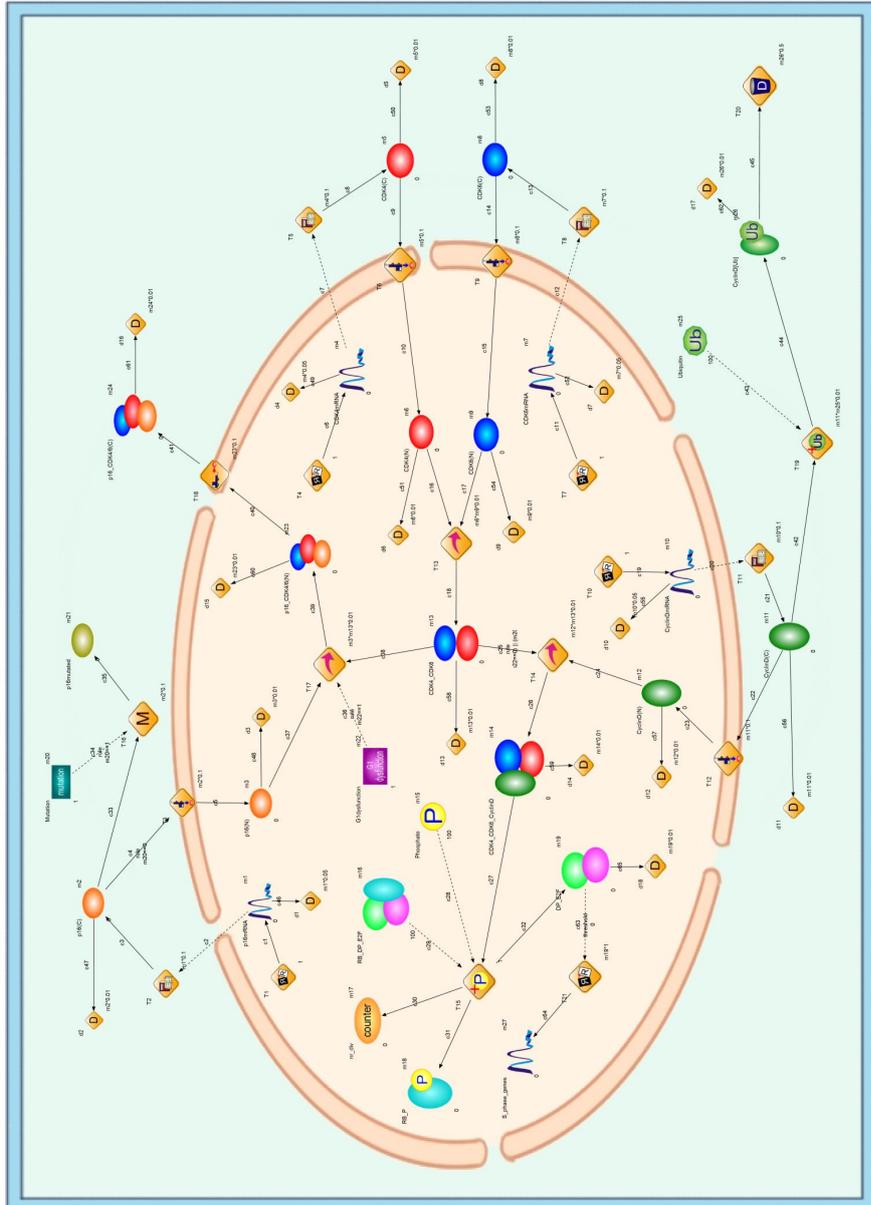| Biological phenomenon	Process	Process type	Process rate
Degradation of proteins	$d2, d3, d5, d6, d8, d9,$ $d11 - d18$	Continuous	mi*0.01
Degradation of mRNAs	$d1, d4, d7, d10$	Continuous	mi*0.05

Fig. 1. A Cell Illustrator screen snapshot of p16-CDK4/6-RB HFPN model.

Fig. 2. A Petri net fragment illustrating phosphorylation of RB and natural degradation of the components involved into the process.

Table 4. Connectors in the HFPN model.

Connector	Firing style	Firing script	Connector type
c4	Rule	m20==1	Input process
c25	Rule	(m20==0 && m22==0) ||	Input process
		(m20==1 && m22==0) ||	
		(m20==1 && m22==1)	
c34	Rule	m22==1	Input association
c36	Rule	m20==0	Input process
c2,c7,c12,c20,c28,c29	threshold	0	Input association
c43,c63			
c9,c14,c16,c17,c22,c24	threshold	0	Input process
c25,c27,c33,c37,c38,c40			
c42,c45-c62,c65			
c1,c3,c5,c6,c8,c10	threshold	0	Output process
c11,c13,c15,c18,c19,c21			
c23,c26,c30,c31,c32,c35			
c39,c41,c44,c64			

3   Model Development

In this section we provide step-by-step explanation on how HFPN model of
p16-Cdk4/6-Rb pathway is created according to the biological facts provided
in Subsection 1.1, and describe relationship between HFPN objects and their
biological counterparts.

The entities used in the model are detailed in Table 1. The entities represent
mRNAs, nuclear and cytoplasmic proteins, protein complexes, phosphate, ubiq-
uitin, mutation and G1-dysfunction. A variable associated with a continuous
entity quantifies concentration of specified substrate. To ensure continual phos-
phorylation of **Rb** we assume that there exist sufficient amount of phosphate and
Rb_DP_E2F concentration. This is why variables **m15** and **m16** are initially
set to 100. Likewise, **m25** is set to 100 to guarantee continual ubiquitination of
CycD. The initial concentration of mRNAs and consequently protein monomers
and their complexes are set to 0 since simulation starts with transcription of re-
lated mRNAs. The entities **G1-dysfunction** and **mutation** are used to indicate
boolean status or presence/absence of corresponding events. The entity **nr_div**
counts the number of cell divisions.

The processes used in the present research include transcription, transla-
tion, nuclear import/export, binding, ubiquitination, phosphorylation and nat-
ural degradation. Relationship between processes and biological phenomena are
illustrated in Table 2 and Table 3. It was reported that mutations in the p16
binding site result in diminished capability of p16 binding to Cdk4/6. This par-
ticularly leads to loss of function of p16 as an inhibitor of Cdk4/6-CycD complex.
In this model, boolean status of mutation is controlled by **T16** and **m20**. As-
signment **m20==1** constitutes presence of mutation, consequently leading to

occurrence of **T16** which in deed arrests p16 in cytoplasm. Otherwise **T3** occurs generating nuclear import of p16. Likewise, the presence/absence of dysfunction in the G1 phase is controlled by entity **G1-dysfunction** and variable **m22**. Assignment **m22==1** indicates the presence of dysfunction in the G1 phase. Next p16 acts as inhibitor of Cdk4/6-CycD complex. We use two Boolean variables with total of four distinct combinations. The rules set for associated connectors and processes depend on four distinct combinations of two Boolean variables **m20**, which represents the mutation in p16, and **m22**, which stands for the dysfunction in G1 phase. Occurrence of transitions **T3**, **T14**, **T16**, and **T17** respectively depend on the rules on connectors **c4**, **c25**, **c34**, and **c36**. For instance, **T3**, nuclear import of p16, occurs if there is no mutation in p16. That is, **T3** can fire only if **m20==0**. All connectors together with their firing styles, firing scripts, and connector types are described in Table 4. A snapshot of HFPN model is illustrated in Fig. 1.

A net fragment bound to **T15** is shown in Fig. 2. This fragment reveals the structural basis for phosphorylation of Rb. Other than connector rules, the phosphorylation of Rb (**T15**) has its activity rule as: (m20==0 && m22==0 || (m20==1 && m22==0) || (m20==1 && m22 ==1). Here, the first statement part is for the case when p16 is not mutated, and there is no dysfunction in the G1 phase. It is known that replicative senescence should occur if there is no mutation and dysfunction in a cell, which means that the cell stops dividing after 50 divisions [17]. In our model, the **m17** is defined as a counter which keeps track the number of divisions, and in the case of no mutation and no dysfunction, it is checked whether the counter is less than 50 or not. If it is not, the cell should stop dividing, which means that RB should not be phosphorylated after 50 divisions. The other two statement parts in the activity rule of **T15** are the cases when p16 is mutated. If p16 is mutated, then the replicative senescence will not occur and the cell will divide continually leading to progression of malignancy.

Process rates are chosen in accordance with the reaction speeds for specific reaction types adopted in [11, 12]. Process rate for transcription is set to 1 to ensure continual mRNA production. The process rates are listed in Table 2.

4   Simulations and Results

In this research, simulations were carried out using Cell Illustrator 5.0 (professional version) that is licensed to Eastern Mediterranean University. Simulation results for concentration behaviour of nuclear and cytoplasmic proteins and their complexes are illustrated in Fig. 3-5. We performed simulations for the following four cases:

1. The p16 is not mutated and there is no dysfunction in the G1 phase (m20==0 && m22==0).
2. The p16 is not mutated and there is dysfunction in the G1 phase (m20==0 && m22==1).
3. The p16 is mutated and there is no dysfunction in the G1 phase (m20==1 && m22==0).

4. The p16 is mutated and there is dysfunction in the G1 phase (m20==1 && m22==1).

It is generally assumed that p16 is transported to the nucleus and acts as a CKI to regulate the G1/S cell cycle checkpoint. This phenomenon has been reported in normal cells where the protein was mainly found in the nucleus but not in the cytoplasm [5]. This fact is supported by the simulation results that are illustrated in Fig.3. For all four cases the concentration of p16(C) is at level 17.5 after almost 50 pt (Petri net time), at which the steady state starts. On the other hand, if there is no dysfunction in the G1 phase (m22==0) and if p16 is not mutated (m20==0) p16(N) is at level 175, that is, almost 10 times more than that of in cytoplasm. It should be noticed that small oscillations in the p16(C) graphs are result of natural degradation which is 10 times slower than the translation process. Mutation in p16 arrests it in cytoplasm. This is why when p16 is mutated its concentration in nucleus is constantly 0.

Healthy and functioning p16 protein forms a complex with Cdk4/6 if it detects a dysfunction. Simulation results, that are illustrated in Fig. 3 and Fig. 4, have shown that p16_Cdk4/6 concentration in cytoplasm and nucleus are respectively at level 125 and 15, i.e. p16_Cdk4/6 concentration in cytoplasm is almost 8 times more than that in nucleus, indicating that p16_Cdk4/6 is accumulated in cytoplasm rather than in nucleus. We were not able to find an experimental result to compare this finding with. The reasonable explanation for this fact however could be the difference between reaction rates of nuclear export and binding, i.e., the former is 10 times faster than the latter.

It was reported in [24] that levels of Cdk proteins in cells vary little throughout the cell cycle. Simulation results for change of Cdk4 and Cdk6 concentrations in nucleus and cytoplasm are shown in Fig. 4-5. These results fully agree with this fact, in sense that concentration of Cdk4 and Cdk6 in nucleus and cytoplasm are respectively at the level 12 and 17 throughout the simulations. This fact remains true even for Cdk4/6 (Fig. 5).

5   Conclusions

The present research explores interaction between HFPN and biological processes, to the benefit of both fields. On the one hand we adopt HFPN for modelling and simulation of specific biological pathways, and consequently expand the list of HFPN applications. On the other hand, through modelling and simulating with HFPN we obtain broader understanding of cell cycle regulation.

The fact that in normal cells p16 protein is mainly accumulated in the nucleus but not in the cytoplasm [5] is confirmed by simulation results. The simulation results have shown that the p16_CDK4/6 protein complex is accumulated in cytoplasm rather than in nucleus. We were not able to find an experimental result to compare this finding with. The simulation results are in agreement with the fact that levels of Cdk proteins in cells vary little throughout the cell cycle [24].
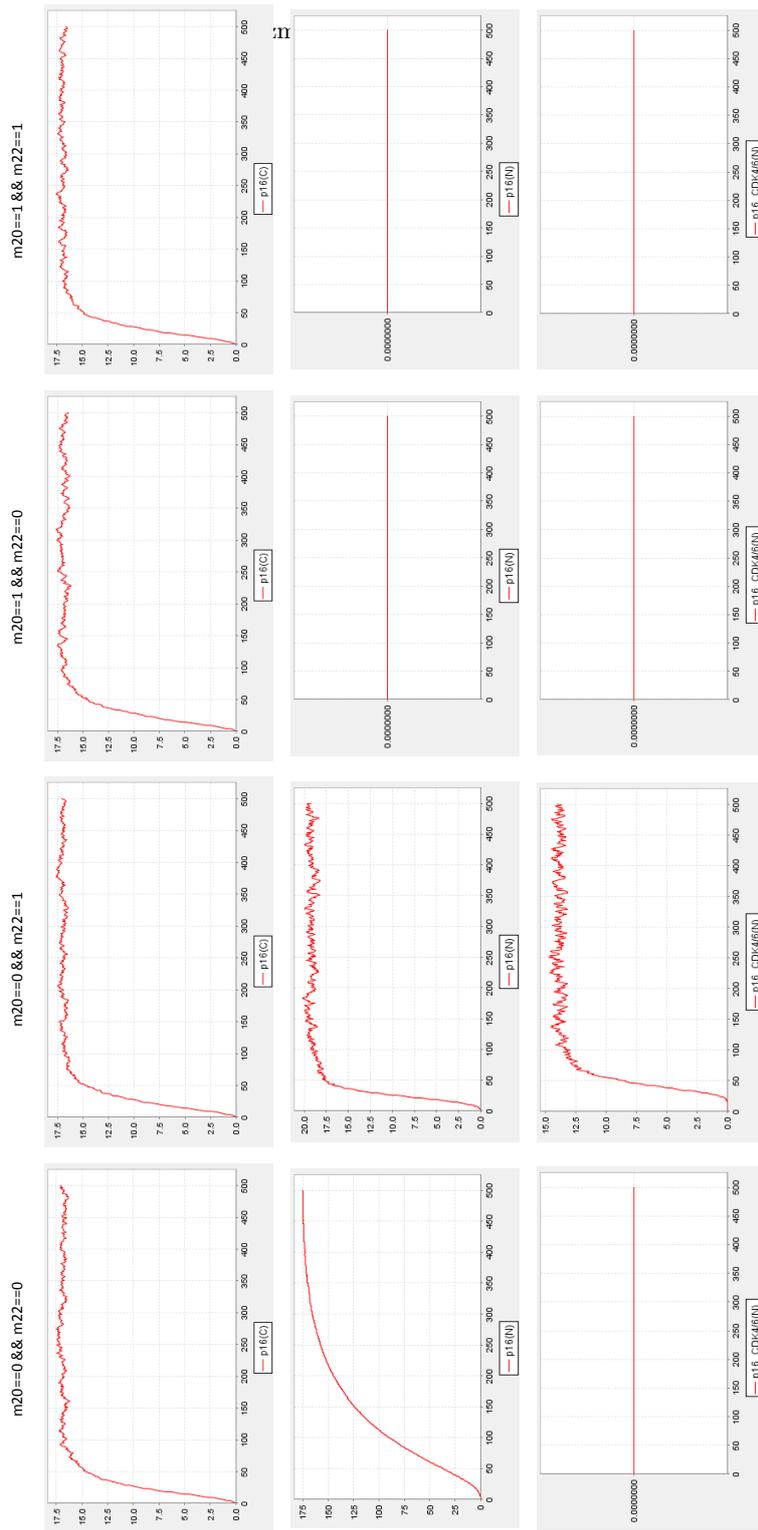
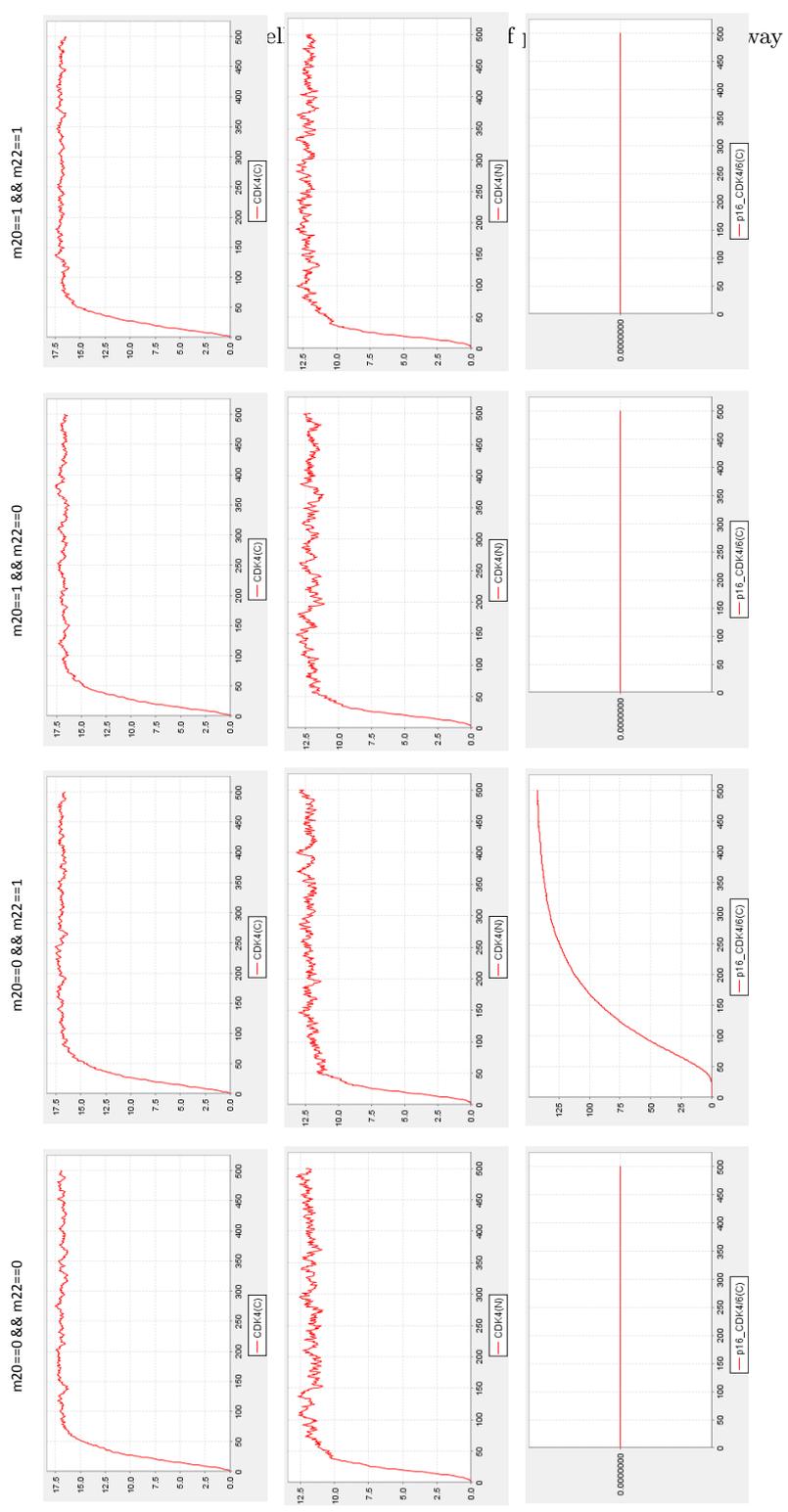**Fig. 3.** Simulation results for p16(C), p16(N) and p16-CDK4/6.

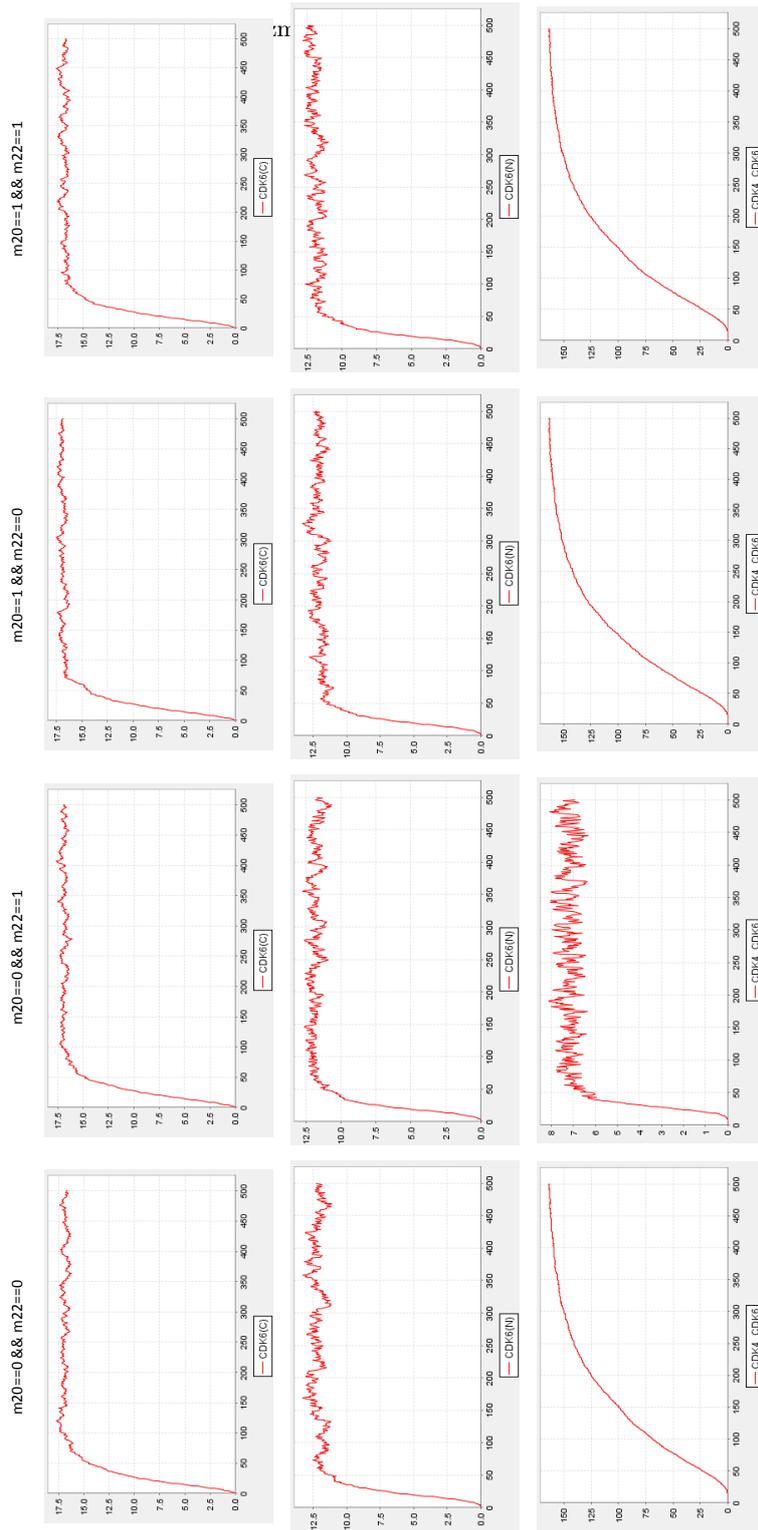**Fig. 4.** Simulation results for CDK4(C) and CDK4(N).

Fig. 5. Simulation results for CDK6(C), CDK6(N) and CDK4/6.

References

1. Agherbi, H., Gaussmann-Wenger, A., Verthuy, C., Chasson, L., Serrano, M., Djabali, M.: Polycomb mediated epigenetic silencing and replication timing at the INK4a/ARF locus during senescence. PLoS One **4**(5), e5622 (2009).
2. Alao, J.P.: The regulation of cyclin D1 degradation: roles in cancer development and the potential for therapeutic invention. Molecular Cancer **6**(24) (2007).
3. Assaraf, Y. G., Ifergana, I., Kadryb, W. N., PinteRb R. Y.: Computer modelling of antifolate inhibition of folate metabolism using hybrid functional petri nets. Journal of Theoretical Biology **240**, 637-647 (2006).
4. Baker, D.J., Perez-Terzic, C., Jin, F., Pitel, K., Niederlander, N.J., Jeganathan, K., Yamada, S., Reyes, S., Rowe, L., Hiddinga, H.J., Eberhardt, N.L., Terzic, A., van Deursen, J.M.: Opposing roles for p16Ink4a and p19Arf in senescence and ageing caused by BubR1 insufficiency. Nat Cell Biol **10**(7), 825-836 (2008).
5. Bartkova, J., Lukas, J., Guldberg, P., Alsner, J., Kirkin, A.F., Zeuthen, J., Bartek, J.: The p16-cyclin D/Cdk4-pRb pathway as a functional unit frequently altered in melanoma pathogenesis. Cancer Res, **56**(23), 5475-5483 (1996).
6. Batt, G., Ropers, D., de Jong, H., Geiselmann, J., Mateescu, R., Page, M., Schneider, D.: Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in Escherichia coli. Bioinformatics 21(Suppl 1) 19-28 (2005).
7. Calder, M., Vyshemirsky, V., Gilbert, D., Orton, R.: Analysis of signalling pathways using the prism model checker. In: Proceeding of Computational Methods in Systems Biology, 3-5 April, Edinburgh, pp.179-190 (2005).
8. Campisi, J.: Cellular senescence as a tumorsuppressor mechanism. Trends Cell Biol **10**, S27-S31 (2001).
9. Castellini, A., Franco, G., Manca, V.: Hybrid functional Petri nets as MP systems. Nat Comput **9**(1), 61-81 (2010).
10. Cell Illustrator^{TM}: User Guide. The University of Tokyo, 2002-2010 Human Genome Center, Institute of Medical Science.
11. Doi, A., Fujita, S., Matsuno, H., Nagasaki, M., Miyano, S.: Constructing biological pathway models with Hybrid Functional Petri Nets. In Silico Biol **4**(3), 271-291 (2004).
12. Doi, A., Nagasaki, M., Matsuno, H.,, Miyano, S.: Simulation-based validation of the p53 transcriptional activity with hybrid functional Petri net. In Silico Biology **6**(1-2), 1-13 (2006).
13. Germain, D., Russell, A., Thompson, A., Hendley, J.: Ubiquitination of free cyclin D1 is independent of phosphorylation on threonine 286. J Biol Chem **275**(16), 12074-12079 (2000).
14. Enoch, T., Nurse, P.: Coupling M phase and S phase: controls maintaining the dependence of mitosis on chromosome replication. Cell **65**(6), 921-923 (1991).
15. Fisher, J., Piterman, N., Hajnal, A., Henzinger, T.A.: Predictive modeling of signaling crosstalk during C. elegans vulval development. PLoS Comput Biol **3**(5), e92 (2007).
16. Herajy, M., Schwarick, M.: A hybrid Petri net model of the eukaryotic cell cycle. In: Proc. 3th International Workshop on Biological Processes and Petri Nets, 29-43, 2012.
17. Hayflick, L.: The Limited in Vitro Lifetime of Human Diploid Cell Strains. Exp Cell Res **37**, 614-636 (1965).

18. Heath, J., Kwiatkowska, M., Norman, G., Parker, D., Tymchyshyn, O.: Probabilistic model checking of complex biological pathways. Theor Comput Sci **391**(3), 239-257 (2008).
19. Kaufmann, K., Nagasaki, M., Jauregui, R.: Modelling the molecular interactions in the flower developmental network of Arabidopsis thaliana. Annals of Botany **107**(9), 1545-1556 (2011).
20. Kwiatkowska, M.Z., Norman, G., Parker, D.: Using probabilistic model checking in systems biology. SIGMETRICS Performance Evaluation Review, **35**(4), 14-21 (2008).
21. Li, C., Nagasaki, M., Ueno, K., Miyano, S.: Simulation-based model checking approach to cell fate specification during Caenorhabditis elegans vulval development by hybrid functional Petri net with extension. BMC Systems Biology **3**(42) (2009).
22. Li, C., NagasakiM̃., Koh, C.H., Miyano, S.: Online model checking approach based parameter estimation to a neuronal fate decision simulation model in Caenorhabditis elegans with hybrid functional Petri net with extension. Mol Biosyst. **7**(5), 1576-1592, (2011).
23. Lin, D.I., Barbash, O., Kumar, K.G., Weber, J.D., Harper, J.W., Klein-Szanto, A.J., Rustgi, A., Fuchs, S.Y., Diehl, J.A.: Phosphorylation-dependent ubiquitination of cyclin D1 by the SCF(FBX4-alphaB crystallin) complex. Mol Cell **24**(3), 355-366 (2006).
24. Malumbres, M., BaRbacid, M.: Cell cycle, CDKs and cancer: a changing paradigm. Nat Rev Cancer, **9**(3), 153-166 (2009).
25. Matheu, A., Maraver, A., Collado, M., Garcia-Cao, I., Canamero, M., Borras, C., Flores, J.M., Klatt, P., Vina, J., Serrano, M., Anti-aging activity of the Ink4/Arf locus. Aging Cell **8**(2), 152-161 (2009).
26. Matsuno, H,, Tanaka, Y., Aoshima, H., Doi, A., Matsui, M., Miyano, S.: Biopathways representation and simulation on Hybrid Functional Petri Nets. In Silico Biol **3**(3), 389-404 (2003).
27. Nagasaki, M., Doi, A., Matsuno, H., Miyano, S.: Genomic object net: I. A platform for modelling and simulating biopathways. Appl Bioinformatics **2**(3), 181-184 (2004).
28. Reisig, W.: Petri Nets: an introduction. EATCS, Monographs on Theoretical Computer Science. Springer, New York, 1985.
29. Russo A.A., Tong, L., Lee, J.O., Jeffrey, P.D., Pavletich, N.P.: Structural basis for inhibition of the cyclin-dependent kinase Cdk6 by the tumour suppressor p16INK4a. Nature **395**(6699), 237-243 (1998).
30. Sherr C.J.: Cancer cell cycle. Science **274**, 1672-1677 (1996).
31. Stevaux O., Dyson N.J.: A revised picture of the E2F transcriptional network and RB function. Curr Opin Cell Biol **14**, 684-691 (2002).
32. Trimarchi J.M., Lees J.A.: Sibling rivalry in the E2F family. Nat Rev Mol Cell Biol **3**, 11-20 (2002).
33. Tyson, J., Novak, B.: A Systems Biology View of the Cell Cycle Control Mechanisms. Elsevier, San Diego, CA, (2011).
34. Vidal A., Koff A.: Cell-cycle inhibitors: three families united by a common cause. Gene **247**(1-2), 1-15 (2000).
35. Walkley, C.R., Orkin, S.H.: RB is dispensable for self-renewal and multilineage differentiation of adult hematopoietic stem cells. Proc Natl Acad Sci USA, **103**(24), 9057-9062 (2006).
36. Weinberg, R.A., The retinoblastoma protein and cell cycle control. Cell **81**(3), 323-330 (1995).

Reconstructing $\mathcal{X}'$-deterministic extended Petri nets from experimental time-series data $\mathcal{X}'$

Marie C.F. Favre[⋆], Annegret K. Wagler

Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes
(LIMOS, UMR CNRS 6158), Université Blaise Pascal (Clermont-Ferrand II)
BP 10125, 63173 Aubière Cedex, France
`marie.favre@isima.fr`   `wagler@isima.fr`

Abstract. This work aims at reconstructing Petri net models for biological systems from experimental time-series data $\mathcal{X}'$. The reconstructed models shall reproduce the experimentally observed dynamic behavior in a simulation. For that, we consider Petri nets with priority relations among the transitions and control-arcs, to obtain additional activation rules for transitions to control the dynamic behavior. The contribution of this paper is to present an integrative reconstruction method, taking both concepts, priority relations and control-arcs, into account. Our approach is based on previous works for special cases and shows how these known steps have to be modified and combined to generate the desired integrative models, called $\mathcal{X}'$-deterministic extended Petri nets.

1 Introduction

The overall aim of systems biology is to analyze biological systems and to understand different phenomena therein as, e.g., responses of cells to environmental changes, host-pathogen interactions, or effects of gene defects. To gain the required insight into the underlying biological processes, experiments are performed and the resulting experimental data are interpreted in terms of models. Depending on the biological aim and the type and quality of the available data, different types of mathematical models are used and corresponding methods for their reconstruction have been developed. Our work is dedicated to Petri nets, a framework which turned out to coherently model static interactions in terms of networks and dynamic processes in terms of state changes, see e.g. [5,9]. A network $(P, T, \mathcal{A}, w)$ reflects the involved system components by places $p \in P$ and their interactions by transitions $t \in T$, linked by weighted directed arcs. Each place $p \in P$ can be marked with an integral number of tokens defining a system state $\boldsymbol{x} \in \mathbb{Z}_+^{|P|}$, dynamic processes are represented by sequences of state changes, performed by switching or firing enabled transitions (see Section 2).

Our central question is to reconstruct models of this type from experimental time-series data by means of an exact, exclusively data-driven approach. We base our method on earlier results from [1,2,3,4,8,12]. This approach takes as input a set P of places and discrete time-series data $\mathcal{X}'$ 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 $(P, T, \mathcal{A}, w)$ 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 different alternatives is not taken randomly, but a specific transition is selected. For that, (standard) Petri nets have to be equipped with additional activation rules to force the switching or firing of special transitions (to reach $\boldsymbol{x}^{j+1}$ from $\boldsymbol{x}^j$), and to prevent all others from switching. Analogously, the reconstruction approach needs to be extended accordingly. In previous works, we considered two possible types of additional activation rules.

On the one hand, in [8,11,12] the concept of priority relations among the transitions of a network was introduced in order to allow the modelization of deterministic systems (see Section 2 for more details). This leads to the notion of $\mathcal{X}'$-deterministic Petri nets, which show a prescribed behavior on the experimentally observed subset $\mathcal{X}'$ of states: the reconstructed Petri nets $(P, T, \mathcal{A}, w)$ do not only contain enough transitions to reach the experimentally observed successors $\boldsymbol{x}^{j+1}$ from $\boldsymbol{x}^j$, but exactly this transition will be selected among all enabled ones in $\boldsymbol{x}^j$ which is necessary to reach $\boldsymbol{x}^{j+1}$.

On the other hand, in [1,2] the concept of control-arcs was used to represent catalytic or inhibitory dependencies. Here, an enabled transition $t \in T$ coupled with a read-arc (resp. an inhibitory-arc) to a place $p \in P$ can switch only if a token (resp. no token) is present in p (see Section 2). This leads to the reconstruction of extended Petri nets which are catalytic conformal with $\mathcal{X}'$.

For consistently integrating both concepts, priority relations and control-arcs, into the modeling framework, the difficulty is that both are concurrent concepts to force or prevent the switching of enabled transitions. In [13], the notion of $\mathcal{X}'$-deterministic extended Petri nets is introduced as the desired output of an integrative reconstruction method. The contribution of this paper is to present the steps of such an approach, based on previous reconstruction methods for special cases [1,2,3,4,8] , and to show how these known steps have to be modified and combined to generate the desired integrative models (see Section 3).

2 Petri nets and extensions

A standard or simple *Petri net* $\mathcal{P} = (P, T, A, w)$ is a weighted directed bipartite graph with two kinds of nodes, places and transitions. The places $p \in P$ represent the system components (e.g. proteins, enzymes, genes, receptors or their conformational states) and the transitions $t \in T$ stand for their interactions (e.g., chemical reactions, activations or causal dependencies). The arcs in

$A \subset (P \times T) \cup (T \times P)$ link places and transitions, and the arc weights $w : A \to \mathbb{N}$ reflect stoichiometric coefficients of the corresponding reactions.

Each place $p \in P$ can be marked with an integral number x_p of tokens, and any marking defines a state $\boldsymbol{x} \in \mathbb{N}^{|P|}$ of the system. In biological systems, all components can be considered to be bounded, as the value x_p of any state refers to the concentration of the studied component $p \in P$, which can only increase up to a certain maximum $\operatorname{cap}(p)$. This leads to a *capacitated Petri net* $(\mathcal{P}, \operatorname{cap})$, i.e., a Petri net $\mathcal{P} = (P, T, A, w)$ together with a capacity function $\operatorname{cap} : P \to \mathbb{N}$, whose set of potential states is $\mathcal{X} := \{\boldsymbol{x} \in \mathbb{N}^{|P|} \mid x_p \le \operatorname{cap}(p)\}$. A transition $t \in T$ is *enabled* in a state $\boldsymbol{x} \in \mathcal{X}$ of a capacitated Petri net if

E1 $x_p \ge w(p, t)$ for all p with $(p, t) \in A$, and,
E2 $x_p + w(t, p) \le \operatorname{cap}(p)$ for all p with $(t, p) \in A$

and we define $T(\boldsymbol{x}) := \{t \in T : t \text{ satisfies E1, E2 in } \boldsymbol{x}\}$.
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 control-arcs by $A_C = A_R \cup A_I$, and the set of all arcs by $\mathcal{A} = A \cup A_R \cup A_I$.

In a capacitated extended Petri net, switching of transitions is additionally controlled by read- and inhibitor-arcs; a transition t satisfying E1 and E2 can switch only if also the following conditions hold:

E3 $x_p \ge w(p, t)$ for all p with $(p, t) \in A_R$, and,
E4 $x_p < w(p, t)$ for all p with $(p, t) \in A_I$.

In an extended Petri net, a transition is *enabled* in a state $x \in \mathcal{X}$ if it satisfies E1, $\dots$, E4 (otherwise, it is *disabled*). The switch of a transition t enabled in x leads to a successor state $\operatorname{succ}_{\mathcal{X}}(\boldsymbol{x}) = \boldsymbol{x}' \in \mathcal{X}$ whose marking is obtained by

$$
x'_p := \begin{cases} x_p - w(p, t), & \text{for all } p \text{ with } (p, t) \in A, \\ x_p + w(t, p), & \text{for all } p \text{ with } (t, p) \in A, \\ x_p, & \text{otherwise.} \end{cases}
$$

In general, there can be more than one transition satisfying E1, $\dots$, E4 in a state $\boldsymbol{x} \in \mathcal{X}$ and we define $T_{\mathcal{A}}(\boldsymbol{x}) := \{t \in T : t \text{ satisfies E1, }\dots\text{, E4 in } \boldsymbol{x}\}$. The decision which transition switches is typically taken randomly (and the dynamic behavior is analyzed in terms of reachability, starting from a certain initial state). This is not appropriate for modeling biological systems which show a deterministic behavior, e.g., where a certain stimulation always results in the same response. In this case, additional activation rules are required in order to force the switch from a state $\boldsymbol{x}$ to a specific successor state $\operatorname{succ}_{\mathcal{X}}(\boldsymbol{x})$. For this purpose, priorities between the transitions of the network can be used to determine which of the transitions in $T_{\mathcal{A}}(\boldsymbol{x})$ has to be taken. Note that these priorities typically reflect the rate of the corresponding reactions where the fastest reaction has highest priority. In Marwan et al. [8] it is proposed to model such priorities with the help of partial orders on the set T of transitions of the network $\mathcal{P}$. Here, a *partial order* $\mathcal{O}$ on T is a relation $\le$ between pairs of elements of T respecting

- reflexivity (i.e., $t \leq t$ holds for all $t \in T$),
- transitivity (i.e., from $t \leq t'$ and $t' \leq t''$ follows $t \leq t''$ for all $t, t', t'' \in T$),
- anti-symmetry (i.e., $t \leq t'$ and $t' \leq t$ implies $t = t'$).

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.

Note that priorities can prevent enabled transitions from switching: for a state $\boldsymbol{x} \in \mathcal{X}$, only a transition $t \in T_{\mathcal{A}}(\boldsymbol{x})$ is *allowed to switch* or *can switch* if

E5 there is no other transition $t' \in T_{\mathcal{A}}(\boldsymbol{x})$ with $(t \leq t') \in \mathcal{O}$.

The set of all transitions that are allowed to switch in $\boldsymbol{x}$ is denoted by

$$T_{\mathcal{A},\mathcal{O}}(\boldsymbol{x}) := \{t \in T : t \text{ satisfies E1}, \ldots, \text{E5 in } \boldsymbol{x}\}.$$

To enforce a deterministic behavior, $T_{\mathcal{A},\mathcal{O}}(\boldsymbol{x})$ must contain at most one element for each $\boldsymbol{x} \in \mathcal{X}$ to enforce that $\boldsymbol{x}$ has a unique successor $\text{succ}_{\mathcal{X}}(\boldsymbol{x})$, see [11] for more details. Extended Petri nets with priorities satisfying this property are said to be $\mathcal{X}$-deterministic. For our purpose, we consider a relaxed condition, namely that $T_{\mathcal{A},\mathcal{O}}(\boldsymbol{x})$ contains at most one element for each experimentally observed state $\boldsymbol{x} \in \mathcal{X}'$, but $T_{\mathcal{A},\mathcal{O}}(\boldsymbol{x})$ may contain several elements for non-observed states $\boldsymbol{x} \in \mathcal{X} \setminus \mathcal{X}'$. We call such Petri nets $\mathcal{X}'$-*deterministic*.

In this paper we consider *capacitated extended Petri nets with priorities* $(\mathcal{P}, \text{cap}, \mathcal{O})$: extended Petri nets $\mathcal{P} = (P, T, \mathcal{A}, w)$ with a capacity function cap : $P \to \mathbb{N}$ on their places and a partial order $\mathcal{O} \subset T \times T$ on their transitions. Our goal is to reconstruct $\mathcal{X}'$-deterministic extended Petri nets from given experimental data $\mathcal{X}'$.

3   Reconstructing $\mathcal{X}'$-deterministic extended Petri nets

In this section, we describe the input, the main ideas, and the generated output of our integrative reconstruction approach.

3.1   Input

A set of components P (later represented by the set of places) is chosen which is expected to be crucial for the studied phenomenon. All known P-invariants [1] of the system (e.g., different conformational stages of a cell, a receptor, a protein) shall be 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 change of nutrient concentrations or the exposition to some pathogens), 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

---

[1] Laxly said, a P-invariant is a set $P' \subseteq P$ of places (components) where the sum of the number of all tokens on all the places in P' is constant. P-invariants are not computed by the algorithm but must be known a priori by a biologist.

for all 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

$$\mathcal{X}'(\boldsymbol{x}^1, \boldsymbol{x}^k) = (\boldsymbol{x}^0; \boldsymbol{x}^1, \ldots, \boldsymbol{x}^k).$$

Several experiments starting from different initial states in a set $\mathcal{X}'_{ini} \subseteq \mathcal{X}'$, reporting the observed state changes for all components $p \in P$ at certain time points, 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.* The developmental decision of starving *P. polycephalum* plasmodia to exit the vegetative plasmodial stage and to enter the sporulation pathway is controlled by environmental factors like visible light [10]. One of the photoreceptors involved in the control of sporulation *Spo* is a phytochrome-like photoreversible photoreceptor protein which occurs in two stages P_{FR} and P_R. 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 [6]. If P_R is exposed to red light R, it is photoconverted back to the initial stage P_{FR}, which prevents sporulation. Note that the changes between the stages P_{FR} and P_R can be experimentally observed due to a change of color. The experimental setting consists of

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

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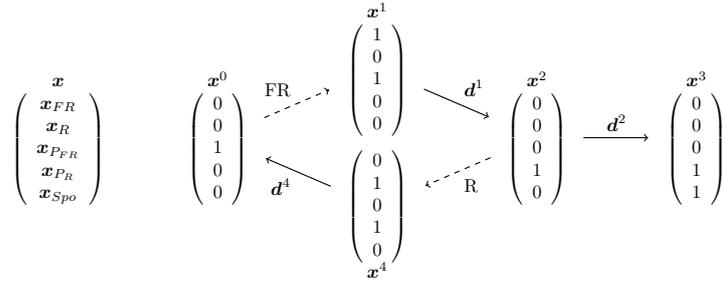

Fig. 1. A scheme illustrating the experimental time-series data described in Exp. 1 concerning the light-induced sporulation of *Physarum polycephalum*, where the entries of the state vectors are interpreted as shown on the left (dashed arrows represent stimulations, solid arrows responses).

In the best case, two consecutively measured states $\boldsymbol{x}^j, \boldsymbol{x}^{j+1} \in \mathcal{X}'$ are also consecutive system states, i.e., $\boldsymbol{x}^{j+1}$ can be obtained from $\boldsymbol{x}^j$ by switching a single transition in T. This is, however, in general not the case (and depends on the chosen time points to measure the states in $\mathcal{X}'$), but $\boldsymbol{x}^{j+1}$ is obtained from $\boldsymbol{x}^j$ by a switching sequence of some length, where the intermediate states are not reported in $\mathcal{X}'$.

For a successful reconstruction approach, 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 $\mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) = \boldsymbol{x}^{j+1} \in \mathcal{X}'$. Moreover, the data $\mathcal{X}'$ are *monotone* if for each pair $(\boldsymbol{x}^j, \boldsymbol{x}^{j+1}) \in \mathcal{X}'$, the possible intermediate states $\boldsymbol{x}^j = \boldsymbol{y}^1, \boldsymbol{y}^2, ..., \boldsymbol{y}^{m+1} = \boldsymbol{x}^{j+1}$ satisfy

$$y_p^1 \leq y_p^2 \leq \ldots \leq y_p^m \leq y_p^{m+1} \text{ for all } p \in P \text{ with } x_p^j \leq x_p^{j+1} \text{ and}$$
$$y_p^1 \geq y_p^2 \geq \ldots \geq y_p^m \geq y_p^{m+1} \text{ for all } p \in P \text{ with } x_p^j \geq x_p^{j+1}.$$

Whereas reproducibility is obviously necessary, it was shown in [3] that monotonicity [2] has to be required too. Due to monotonicity, a capacity $\mathrm{cap}(p)$ can be determined from $\mathcal{X}'$ for each $p \in P$ by $\mathrm{cap}(p) = \max\{x_p : \boldsymbol{x} \in \mathcal{X}'\}$, but is not required for the reconstruction.

3.2   Output

A capacitated extended Petri net with priorities $(\mathcal{P}, \mathrm{cap}, \mathcal{O})$ 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 $\mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) = \boldsymbol{x}^{j+1} \in \mathcal{X}'$. This can be interpreted as follows. With $\mathcal{P}$, an *incidence matrix* $M(\mathcal{P}) \in \mathbb{Z}^{|P| \times |T|}$ is associated, where each row corresponds to a place $p \in P$ of the network, and each column $M(\mathcal{P})_{\cdot t}$ to the *update vector* $\boldsymbol{r}^t$ of a transition $t \in T$:

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

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 the state vector $\boldsymbol{x}^{j+1}$ from $\boldsymbol{x}^j$ by adding the corresponding columns $M(\mathcal{P})_{\cdot t}$ of $M(\mathcal{P})$ for all $t \in T'$:

$$\boldsymbol{x}^j + \sum_{t \in T'} M(\mathcal{P})_{\cdot t} = \boldsymbol{x}^{j+1}.$$

Hence, T has to contain enough transitions to perform all experimentally observed switching sequences. The underlying standard network $\mathcal{P} = (P, T, A, w)$ is

---

[2] This is equivalent to say that system states in $\mathcal{X}'$ have been measured to appropriate time points such that the values of their components do not oscillate between two measured states or, equivalently, that all essential responses are indeed reported in the experiments.

conformal with $\mathcal{X}'$ if, for any two consecutive states $\boldsymbol{x}^j, \mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) = \boldsymbol{x}^{j+1} \in \mathcal{X}'$, the linear equation system

$$\boldsymbol{x}^{j+1} - \boldsymbol{x}^j = M(\mathcal{P})\boldsymbol{\lambda}$$

has an integral solution $\boldsymbol{\lambda} \in \mathbb{N}^{|T|}$ such that $\boldsymbol{\lambda}$ is the incidence vector of 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 $\boldsymbol{y}^l + M(\mathcal{P})_{.t^l} = \boldsymbol{y}^{l+1}$ 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 of the reconstruction approach consists of the set of all $\mathcal{X}'$-deterministic extended Petri nets $(\mathcal{P}, \mathrm{cap}, \mathcal{O})$ (all having the same set P of places and the same capacities cap deduced from $\mathcal{X}'$).

3.3   Representation of the observed responses

To solve the problem of representing the observed responses by switching sequences, we propose the following approach, based on previous works in [4,8].

Extraction of difference vectors. As initial step, extract the observed changes of states from the experimental data. For that, define the set

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

Example 2. From our running example in Fig. 1 we obtain $\mathcal{D} = \{d^1, d^2, d^4\}$ with $\boldsymbol{d}^1 = \boldsymbol{x}^2 - \boldsymbol{x}^1 = (-1, 0, -1, 1, 0)^T$, $\boldsymbol{d}^2 = \boldsymbol{x}^3 - \boldsymbol{x}^2 = (0, 0, 0, 0, 1)^T$ and $\boldsymbol{d}^4 = \boldsymbol{x}^0 - \boldsymbol{x}^4 = (0, -1, 1, -1, 0)^T$.

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 $M \in \mathbb{Z}^{|P| \times |T|}$.

The first step is to describe the set of potential update vectors which might constitute the columns of M.

Representation of difference vectors. Recall that two consecutively measured states $\boldsymbol{x}^j, \boldsymbol{x}^{j+1} \in \mathcal{X}'$ are not necessarily consecutive system states, i.e., $\boldsymbol{x}^{j+1}$ may be obtained from $\boldsymbol{x}^j$ by a switching sequence of some length, where the intermediate states are not reported in $\mathcal{X}'$. Due to monotonicity, the values of the elements cannot oscillate in the intermediate states between $\boldsymbol{x}^j$ and $\boldsymbol{x}^{j+1}$.

Moreover, for any P-invariant $P' \in \mathcal{I}_P$, all suitable update vectors have to satisfy $\sum_{p \in P'} r_p = 0$. Hence, it suffices to represent any $\boldsymbol{d}^j \in \mathcal{D}$ using only vectors from the following set

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

Remark 1. In previous approaches [4], none of the reconstructed (standard) networks must contain a transition enabled at any of the observed terminal states $\boldsymbol{x}^k \in \mathcal{X}'_{term}$; hence all such vectors in Box($\boldsymbol{d}^j$) could be removed. This is not the case for extended Petri nets as desired output of the reconstruction, since the corresponding transitions can be disabled due to control-arcs. Here, we only exclude the zero vector $\boldsymbol{0}$ as trivial update vector.

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

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

By construction, Box($\boldsymbol{d}^j$) and $\Lambda(\boldsymbol{d}^j)$ are always non-empty since $\boldsymbol{d}^j$ itself is always a solution due to the required reproducibility of the input data $\mathcal{X}'$ (which particularly includes $\boldsymbol{d}^j \neq \boldsymbol{0}$ for all $\boldsymbol{d}^j \in \mathcal{D}$). For each $\boldsymbol{\lambda} \in \Lambda(\boldsymbol{d}^j)$, construct the (multi-)set

$$\mathcal{R}(\boldsymbol{d}^j, \boldsymbol{\lambda}) = \{r^t \in \text{Box}(\boldsymbol{d}^j) : \lambda_t \neq 0\}$$

of update vectors used for this solution $\boldsymbol{\lambda}$.

Example 3. For $\mathcal{D}$ from Exp. 2, the update vectors for a decomposition are Box($\boldsymbol{d}^1$) = $\{\boldsymbol{d}^1, \boldsymbol{r}^1, \boldsymbol{r}^2\}$, Box($\boldsymbol{d}^2$) = $\{\boldsymbol{d}^2\}$ and Box($\boldsymbol{d}^4$) = $\{\boldsymbol{d}^4, \boldsymbol{r}^3, \boldsymbol{r}^4\}$ with vectors $\boldsymbol{r}^1 = (-1, 0, 0, 0, 0)^T$, $\boldsymbol{r}^2 = (0, 0, -1, 1, 0)^T$, $\boldsymbol{r}^3 = (0, -1, 0, 0, 0)^T$ and $\boldsymbol{r}^4 = (0, 0, 1, -1, 0)^T$. Hence, the possible decomposition of the responses are $\boldsymbol{d}^1 = \boldsymbol{d}^1 = \boldsymbol{r}^1 + \boldsymbol{r}^2$, $\boldsymbol{d}^2 = \boldsymbol{d}^2$ and $\boldsymbol{d}^4 = \boldsymbol{d}^4 = \boldsymbol{r}^3 + \boldsymbol{r}^4$ and the resulting sets are

$$\begin{aligned}
\mathcal{R}(\boldsymbol{d}^1, \boldsymbol{\lambda}^1) &= \{\boldsymbol{d}^1\}, \mathcal{R}(\boldsymbol{d}^1, \boldsymbol{\lambda}^2) = \{\boldsymbol{r}^1, \boldsymbol{r}^2\}, \\
\mathcal{R}(\boldsymbol{d}^2, \boldsymbol{\lambda}) &= \{\boldsymbol{d}^2\}, \\
\mathcal{R}(\boldsymbol{d}^4, \boldsymbol{\lambda}^1) &= \{\boldsymbol{d}^4\}, \mathcal{R}(\boldsymbol{d}^4, \boldsymbol{\lambda}^2) = \{\boldsymbol{r}^3, \boldsymbol{r}^4\}.
\end{aligned}$$

3.4 Priority conflicts.

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.

Sequences and their conflicts. Every permutation $\pi = (\boldsymbol{r}^{t_1}, \dots, \boldsymbol{r}^{t_m})$ of the elements of a set $\mathcal{R}(\boldsymbol{d}^j, \boldsymbol{\lambda})$ 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

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

By construction, every such sequence σ respects monotonicity and 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 priority 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 priority conflict* (WPC) if $y \neq y'$ and a *strong priority conflict* (SPC) if $y = y'$. Note that a WPC can be resolved by adding appropriate control-arcs, whereas a SPC cannot be resolved that way (see section 3.5).

Note that we have a strong priority conflict between the trivial sequence $\sigma(\boldsymbol{x}^k, \boldsymbol{0})$ for any terminal state $\boldsymbol{x}^k \in \mathcal{X}'_{term}$ and any sequence σ containing $\boldsymbol{x}^k$ as intermediate state. Such sequences σ are not catalytic conformal due to [2].

Example 4. From the running example, we obtain the following sequences

$$\sigma_1(\boldsymbol{x}^1, \boldsymbol{d}^1) = ((\boldsymbol{x}^1, \boldsymbol{d}^1)) \qquad\qquad \sigma_1(\boldsymbol{x}^4, \boldsymbol{d}^4) = ((\boldsymbol{x}^4, \boldsymbol{d}^4))$$
$$\sigma_2(\boldsymbol{x}^1, \boldsymbol{d}^1) = ((\boldsymbol{x}^1, \boldsymbol{r}^1), (\boldsymbol{x}^0, \boldsymbol{r}^2)) \qquad \sigma_2(\boldsymbol{x}^4, \boldsymbol{d}^4) = ((\boldsymbol{x}^4, \boldsymbol{r}^3), (\boldsymbol{x}^2, \boldsymbol{r}^4))$$
$$\sigma_3(\boldsymbol{x}^1, \boldsymbol{d}^1) = ((\boldsymbol{x}^1, \boldsymbol{r}^2), (\boldsymbol{x}^5, \boldsymbol{r}^1)) \qquad \sigma_3(\boldsymbol{x}^4, \boldsymbol{d}^4) = ((\boldsymbol{x}^4, \boldsymbol{r}^4), (\boldsymbol{x}^6, \boldsymbol{r}^3))$$
$$\sigma(\boldsymbol{x}^2, \boldsymbol{d}^2) = ((\boldsymbol{x}^2, \boldsymbol{d}^2)) \qquad\qquad \sigma(\boldsymbol{x}^3, 0) \text{ and } \sigma(\boldsymbol{x}^0, 0)$$

where $\boldsymbol{x}^5 = (1, 0, 0, 1, 0)^T$ and $\boldsymbol{x}^6 = (0, 1, 1, 0, 0)^T$. Between these sequences, we have SPCs and WPCs as indicated in Fig. 2.

Construction of the priority conflict graph. To reflect the weak and strong priority conflicts between all possible sequences resulting from $\mathcal{X}'$, we construct a *priority conflict graph* $\mathcal{G} = (V_D \cup V_{term}, E_D \cup E_W \cup E_S)$ where the nodes correspond to sequences and the edges to priority conflicts:

- V_D contains for all $\boldsymbol{x}^j \in \mathcal{X}' \backslash \mathcal{X}'_{term}$ and the difference vector $\boldsymbol{d}^j = \mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}^i) - \boldsymbol{x}^i$, for all $\boldsymbol{\lambda} \in \Lambda(\boldsymbol{d}^j)$ and all permutations π of $\mathcal{R}(\boldsymbol{d}^j, \boldsymbol{\lambda})$ the sequence $\sigma_{\pi, \boldsymbol{\lambda}}(\boldsymbol{x}^j, \boldsymbol{d}^j)$.
- 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 reflects all strong priority conflicts between sequences σ, σ' stemming from distinct difference vectors.
- E_W reflects all weak priority conflicts between sequences σ, σ' stemming from distinct difference vectors.

The edges in E_D induce a clique partition $\mathcal{Q}$ of $\mathcal{G}$ in as many cliques [3] as there are observed states in $\mathcal{X}' \setminus \mathcal{X}'_{term}$ resp. difference vectors in $\mathcal{D}$: $V_\mathcal{D} = Q_1 \cup \ldots \cup Q_{|\mathcal{D}|}$. Moreover, each node in V_{term} corresponds to a clique of size 1, so that $\mathcal{G}$ is partitioned into $|\mathcal{X}'|$ many cliques.

Example 5. The resulting priority conflict graph $\mathcal{G}$ of the running example is shown in Fig. 2.

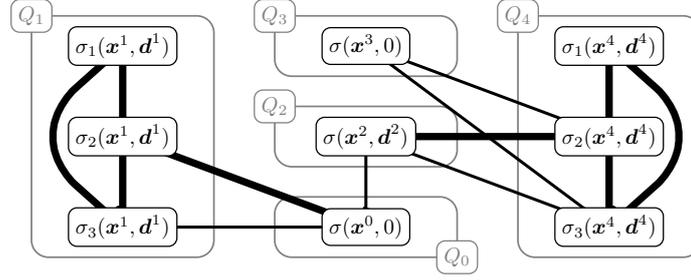[3] A clique is a subset of mutually adjacent nodes.

Fig. 2. The conflict graph resulting from the sequences listed in Exp. 4, where bold edges indicate SPCs, thin edges WPCs and gray boxes the clique partition.

Selection of suitable sequences. To obtain a network explaining all observations, we have to select one sequence per difference vector $\boldsymbol{d}^j$, i.e., exactly one node from each clique $Q_j \in \mathcal{Q}$. To encode the priority conflicts involving terminal states, we require also to select all trivial sequences $\sigma(\boldsymbol{x}^k, \boldsymbol{0})$, i.e., all nodes from V_{term}. Thus, we are interested in subsets $S \subseteq V_D$ of cardinality $|\mathcal{D}|$ such that $|S \cap Q_j| = 1$ for each $Q_j \in \mathcal{Q}$, and no SPCs occur in $S \subseteq V_{term}$.
The set of all such solutions $S \cup V_{term}$ can be encoded by all vectors $\boldsymbol{x} \in \{0,1\}^{|V_D \cup V_{term}|}$ satisfying

$$
\begin{aligned}
\sum_{\sigma \in Q_j} \boldsymbol{x}_\sigma &= 1 & \forall Q_j \in \mathcal{Q} \\
\boldsymbol{x}_\sigma &= 1 & \forall \sigma \in V_{term} \\
\boldsymbol{x}_\sigma + \boldsymbol{x}_{\sigma'} &\leq 1 & \forall \sigma\sigma' \in E_S \\
\boldsymbol{x}_\sigma &\in \{0,1\} & \forall \sigma \in V_D \cup V_{term}.
\end{aligned}
$$

Example 6. From $\mathcal{G}$ in Exp. 5, we obtain the following feasible subsets $S_i \cup V_{term}$

$S_1 = \{\sigma_1(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma(\boldsymbol{x}^2, \boldsymbol{d}^2), \sigma_1(\boldsymbol{x}^4, \boldsymbol{d}^4)\},\ S_3 = \{\sigma_1(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma(\boldsymbol{x}^2, \boldsymbol{d}^2), \sigma_3(\boldsymbol{x}^4, \boldsymbol{d}^4)\},$
$S_2 = \{\sigma_3(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma(\boldsymbol{x}^2, \boldsymbol{d}^2), \sigma_1(\boldsymbol{x}^4, \boldsymbol{d}^4)\},\ S_4 = \{\sigma_3(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma(\boldsymbol{x}^2, \boldsymbol{d}^2), \sigma_3(\boldsymbol{x}^4, \boldsymbol{d}^4)\}.$

Composition of conformal networks. Every selected subset $S \cup V_{term}$ corresponds to a standard network $\mathcal{P}_S = (P, T_S, A_S, w)$ which is conformal with $\mathcal{X}'$ (but not yet necessarily $\mathcal{X}'$-deterministic):

- 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 $\sigma \in S$;
- there might be weak priority conflicts $\sigma\sigma' \in E_W$ for nodes $\sigma, \sigma' \in S \cup V_{term}$ which have to be resolved subsequently by inserting appropriate control-arcs.

Example 7. We apply the method only to the feasible set $S_1 \cup V_{term}$ from Exp. 6 (all solutions for $S_2 \cup V_{term}$, $S_3 \cup V_{term}$ and $S_4 \cup V_{term}$ are presented in Exp. 9). We construct the standard network presented in Fig. 3 with $T_{S_1} = \{\boldsymbol{d}^1, \boldsymbol{d}^2, \boldsymbol{d}^4\}$. There is a priority conflict between $\sigma(\boldsymbol{x}^2, \boldsymbol{d}^2)$ and $\sigma(\boldsymbol{x}^0, 0)$ due to $\boldsymbol{d}^2, 0 \in T(\boldsymbol{x}^2) \cap T(\boldsymbol{x}^0)$.
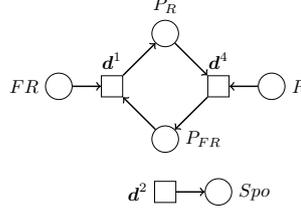
Fig. 3. Standard network $\mathcal{P}_{S_1} = (P, T_{S_1}, A_{S_1}, w)$ from solution S_1 (Exp. 6)

3.5   Inserting control-arcs.

For each of the yet reconstructed standard networks $\mathcal{P}_S = (P, T_S, A_S, w)$ resulting from a selected subset S from the previous reconstruction step, we have to determine appropriate control-arcs in order to resolve weak priority conflicts corresponding to edges $\sigma\sigma' \in E_W$ with $\sigma, \sigma' \in S \cup V_{term}$ (if any), in order to turn $\mathcal{P}_S$ into a catalytic conformal extended Petri net $\mathcal{P}_S = (P, T_S, \mathcal{A}_S, w)$.

Recall that we have a weak priority conflict between two sequences σ and σ' if there are 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 $(\boldsymbol{y}', \boldsymbol{r}^{t'}) \in \sigma'$. This weak priority conflict has to be resolved by adding appropriate control-arcs such that

- the update vector $\boldsymbol{r}^t$ becomes a transition t with $t \in T_{\mathcal{A}}(\boldsymbol{y})$ but $t \notin T_{\mathcal{A}}(\boldsymbol{y}')$ (or vice versa) if $\boldsymbol{y}, \boldsymbol{y}' \notin \mathcal{X}'_{term}$ or
- the update vector $\boldsymbol{r}^t$ becomes a transition t with $t \in T_{\mathcal{A}}(\boldsymbol{y})$ which is disabled by control-arcs in $\boldsymbol{y}'$ if $\boldsymbol{y}' \in \mathcal{X}'_{term}$.

Inserting control-arcs This task can be done by using similar techniques as proposed in [1,2]. Let $P(\boldsymbol{y}, \boldsymbol{y}')$ be the set of places where $\boldsymbol{y}$ and $\boldsymbol{y}'$ differ, i.e., $P(\boldsymbol{y}, \boldsymbol{y}') = \{p \in P : y_p \neq y'_p\}$. In order to disable transition t resulting from $\boldsymbol{r}^t$ at $\boldsymbol{y}'$, we can include either

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

Each of the so-determined control-arcs defines a transition t with the desired properties (inheriting the standard arcs from $\boldsymbol{r}^t$ and having either a read-arc or an inhibitor-arc as described above).

Remark 2. In case of a SPC involving states $\boldsymbol{y} = \boldsymbol{y}'$, the set $P(\boldsymbol{y}, \boldsymbol{y}')$ becomes empty and it is, therefore, not possible to resolve a SPC by control-arcs.

For every reconstructed standard network $\mathcal{P}_S = (P, T_S, A_S, w)$ and any subset $P' \subseteq P$ containing exactly one place from $P(\boldsymbol{y}, \boldsymbol{y}')$ for every weak priority conflict, we get a catalytic conformal extended Petri net $\mathcal{P}_{S,P'} = (P, T_S, \mathcal{A}_{S,P'}, w)$ by inserting the respective control-arcs for all $p \in P'$.

Example 8. We define control-arcs to resolve the WPC between $\sigma(\boldsymbol{x}^2, \boldsymbol{d}^2)$ and $\sigma(\boldsymbol{x}^0, 0)$ for the network $\mathcal{P}_{S_1}$. We obtain

$$P(\boldsymbol{x}^2, \boldsymbol{x}^0) = \{P_{FR}, P_R\} \text{ by } \boldsymbol{x}^2 = (0,0,\boldsymbol{0},\boldsymbol{1},0)^T \text{ and } \boldsymbol{x}^0 = (0,0,\boldsymbol{1},\boldsymbol{0},0)^T.$$

Any non-empty subset of $P(\boldsymbol{x}^2, \boldsymbol{x}^0)$ can be used to disable $\boldsymbol{d}^2$ at $\boldsymbol{x}^0 \in \mathcal{X}'_{term}$. For P_{FR}, $\boldsymbol{x}^2_{P_{FR}} < \boldsymbol{x}^0_{P_{FR}}$ holds, leading to an inhibitor-arc $(P_{FR}, \boldsymbol{d}^2) \in \mathcal{A}_{S_1, P'}$, and for P_R, $\boldsymbol{x}^2_{P_R} > \boldsymbol{x}^0_{P_R}$ holds, leading to a read-arc $(P_R, \boldsymbol{d}^2) \in \mathcal{A}_{S_1, P'}$ both with weight 1. The two possible alternatives are presented in Fig. 4.
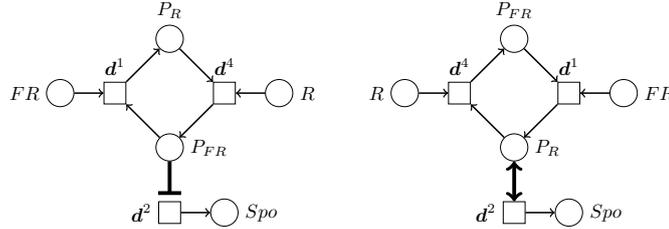

Fig. 4. The two catalytic conformal networks resulting from $\mathcal{P}_{S_1}$ in Exp. 8

3.6 Determining priority relations

To generate the required priorities for each of the yet reconstructed extended networks $\mathcal{P}_{S, P'} = (P, T_S, \mathcal{A}_{S, P'}, w)$, we only need to set the priorities among all the transitions in T_S according to the sequences selected for S.

Recall that every $\sigma \in S$ stands for a sequence

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

which induces a priority relation $\mathcal{O}_\sigma$ indicating that the transition t_i resulting from $\boldsymbol{r}^{t_i}$ is supposed to have highest priority at $\boldsymbol{y}^i$. That is, $\mathcal{O}_\sigma$ is defined by

$$\mathcal{O}_\sigma = \left\{ t_i > t : t \in T_{\mathcal{A}_{S, P'}}(\boldsymbol{y}^i) \setminus t_i, 1 \le i \le m \right\}.$$

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 implies finally that every extended network $\mathcal{P}_{S, P'} = (P, T_S, \mathcal{A}_{S, P'}, w)$ together with the partial order $\mathcal{O}_{S, P'}$ constitutes an $\mathcal{X}'$-deterministic extended Petri net fitting the given data $\mathcal{X}'$.

Example 9. For the running example, it is left to determine the priority relations. For the extended Petri nets $\mathcal{P}_{S_1,P'}$, we can easily verify that $T_{\mathcal{A}_{S_1,P'}}(\boldsymbol{x})$ contains exactly one transition for all $\boldsymbol{x} \in \mathcal{X}'$, so no priorities are implied and $\mathcal{O}_{S_1,P'} = \emptyset$ follows. For the Petri nets coming from the other sets S_2, S_3, S_4, all possible minimal $\mathcal{X}'$-deterministic extended Petri nets are depicted in Fig. 5, 6 and 7.
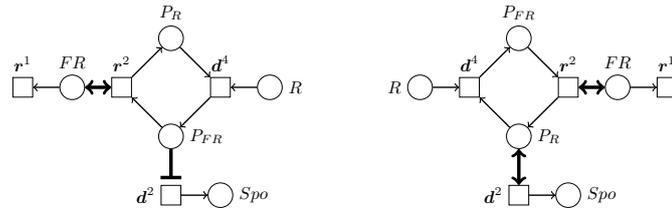
Fig. 5. From S_2, two catalytic conformal networks $\mathcal{P}_{S_2}$ result, both with priority relations $\mathcal{O}_2 = \{(\boldsymbol{r}^2 > \boldsymbol{r}^1)\}$.

Fig. 6. From S_3, four minimal catalytic conformal networks $\mathcal{P}_{S_3}$ result, all with priority relations $\mathcal{O}_3 = \{(\boldsymbol{r}^4 > \boldsymbol{r}^3)\}$.
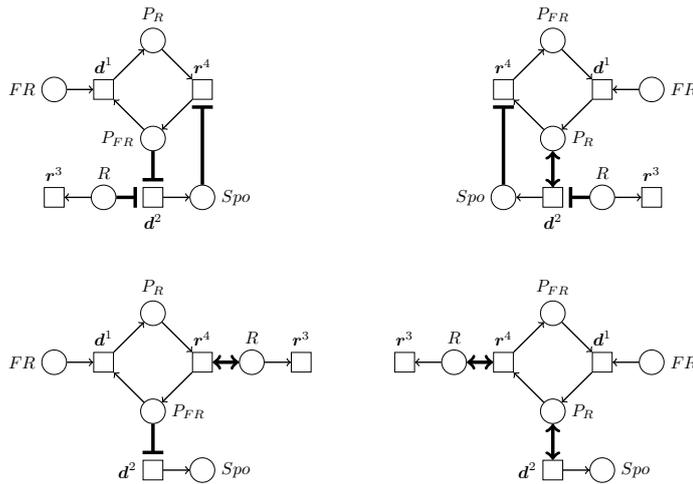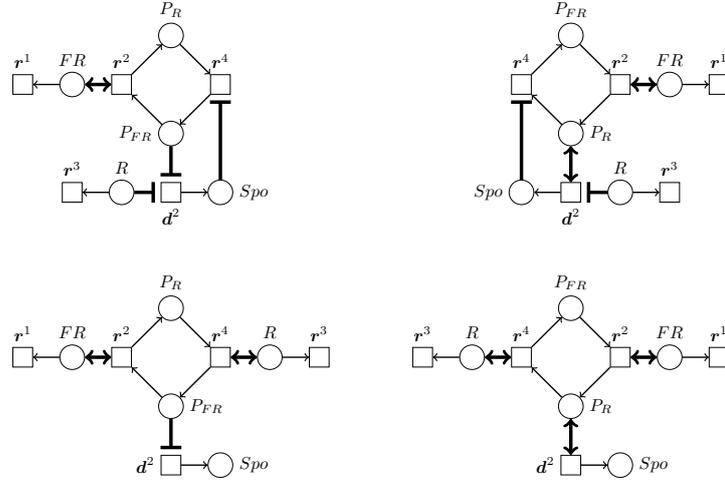
Fig. 7. From S_4, four minimal catalytic conformal networks $\mathcal{P}_{S_4}$ result, all with priority relations $\mathcal{O}_4 = \{(r^2 > r^1), (r^4 > r^3)\}$.

4   Concluding remarks

To summarize, we present in this paper the steps of an integrative reconstruction method to generate all possible $\mathcal{X}'$-deterministic extended Petri nets from monotone and reproducible experimental time-series data $\mathcal{X}'$.

This approach is based on previous works for special cases: the reconstruction of standard networks [4], standard networks with priorities [8] and extended Petri nets [1,2]. Here, we modify and generalize the previous methods by

- adjusting the representation of the observed difference vectors d^j to the case of extended networks with priorities (where d^j might be enabled at a terminal state in $\mathcal{X}'_{term}$),
- refining the idea from [8] to construct a priority conflict graph by distinguishing weak and strong priority conflicts (where only strong conflicts affect the selection),
- generalizing the method from [1,2] such that weak priority conflicts can be resolved by inserting control-arcs (where arbitrary arcs weights can occur).

Note that a preprocessing (to test the experimental data $\mathcal{X}'$ for reproducibility and, if necessary, to handle infeasible situations) can be handled similar as in [4] and a postprocessing (to keep only "minimal" solutions in the sense that all other $\mathcal{X}'$-deterministic extended Petri nets fitting the data contain the returned ones) is presented in [13].

In total, this integrative approach is promising for the reconstruction of networks fully fitting the experimentally observed phenomena.

Our further goal is to make the new approach accessible by a suitable implementation, e.g., using Answer Set Programming as in the case of the reconstruc-

tion of standard networks with priorities [7] and to apply it to new biological experimental data.

References

1. 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.
2. 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.
3. M. Durzinsky, A. K. Wagler, and R. Weismantel. A combinatorial approach to reconstruct Petri nets from experimental data. In Monika Heiner and Adelinde M. Uhrmacher, editors, *CMSB*, volume 5307 of *Lecture Notes in Computer Science*, pages 328–346. Springer, 2008.
4. M. Durzinsky, A. K. Wagler, and R. Weismantel. An algorithmic framework for network reconstruction. *Journal of Theoretical Computer Science*, 412(26):2800–2815, 2011.
5. I. Koch and M. Heiner. Petri nets. In B. H. Junker, F. Schreiber, editors, *Biological Network Analysis*, Wiley Book Series on Bioinformatics, pages 139–179, 2007.
6. 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.
7. M. Durzinsky M. Ostrowski, T. Schaub and W. Marwan. Automatic network reconstruction using ASP. *Th. and Practice of Logic Progr.*, 11:749–766, 2011.
8. 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.
9. J. W. Pinney, R. D. Westhead, and G. A. McConkey. Petri net representations in systems biology. *Biochem. Soc. Tarns.*, 31:1513–1515, 2003.
10. 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.
11. L. M. Torres and A. K. Wagler. Encoding the dynamics of deterministic systems. *Math. Methods of Operations Research*, 73:281–300, 2011.
12. A. K. Wagler. *Prediction of network structure*, In: Modeling and System Biology, I. Koch, F. Schreiber, W. Reisig (eds.) *Computational Biology 16*. Springer London, pages 309–338 2010.
13. A. K. Wagler and J.-T. Wegener. On minimality and equivalence of Petri nets. *Proceedings of Concurrency, Specification and Programming CS&P'2012 Workshop*, 2:382–393, 2012.

The Foundation of Evolutionary Petri Nets⋆

Marco S. Nobile[1], Daniela Besozzi[2], Paolo Cazzaniga[3], Giancarlo Mauri[1]

[1] Università degli Studi di Milano-Bicocca, Dipartimento di Informatica,
Sistemistica e Comunicazione
Viale Sarca 336, 20126 Milano (Italy)
E-mail: {nobile,mauri}@disco.unimib.it
[2] Università degli Studi di Milano, Dipartimento di Informatica
Via Comelico 39, 20135 Milano (Italy)
E-mail: besozzi@di.unimi.it
[3] Università degli Studi di Bergamo, Dipartimento di Scienze Umane e Sociali
Piazzale S. Agostino 2, 24129 Bergamo (Italy)
E-mail: paolo.cazzaniga@unibg.it

Abstract. Evolutionary Computation (EC) mimics evolution processes
to solve burdensome computational problems, like the design, optimiza-
tion and reverse engineering of complex systems, and its effectiveness is
tied to a proper formalization of the candidate solutions. Petri Net (PN)
formalism is extensively exploited for the modeling, simulation and anal-
ysis of the structural and behavioral properties of complex systems. Here
we introduce a novel evolutionary algorithm inspired by EC, the Evolu-
tionary Petri Net (EPN), which is based on an extended class of PNs,
called Resizable Petri Net (RPN), provided with two genetic operators:
mutation and crossover. RPN includes the new concept of hidden places
and transitions, that are used by the genetic operators for the optimiza-
tion of PN-based models. We present a potential application of EPNs to
face one of the most challenging problems in Systems Biology, the reverse
engineering of biochemical reaction networks.

1 Introduction

Evolution is the process whereby the inherited characteristics of a population
are modified over successive generations; these modifications help the offspring
of the best fitting individuals in their competition for sustenance resources and
survival. In other words, the evolution process is the way a population adapts
itself in order to better survive in a complex and ever changing environment.

Evolutionary Computation (EC) is the application of these principles to solve
complex computational problems [1], when classic deterministic approaches may
results unfeasible due to a prohibitive computational effort, for instance in the
case of high-dimensional combinatorial problems. In the EC field, the individuals
forming a population correspond to a set of candidate solutions (smaller than
the whole set of possible solutions) which are iteratively improved by means of
an evolutive pressure, driven by a fitness function that quantifies the quality of

---

each individual. The improvement is determined by the application of genetic operators (e.g., mutation, crossover), which modify the structure of each individual, allowing the exploration of the search space and the convergence toward an optimal solution.

Intuitively, the structure of the individuals, and the way genetic operators are implemented, have a deep impact on the evolutionary process. In the case of distributed, asynchronous and concurrent systems, one of the most common modeling approach consists in the use of Petri Nets (PNs), that is, bipartite graphs that allow the analysis of the structural properties of the system and of its dynamic behavior [2]. PNs have been coupled to EC to solve problems in a plethora of different fields, e.g., job scheduling optimization [3], development of robust and flexible manufacturing systems [4], learning and classification [5]. PNs have also been exploited to solve complex biological problems, such as the discovery of PN models of biochemical systems consistent with population level genetic models [6] or the automatic reverse engineering of kinetic metabolic pathways [7], that is, the automatic creation of a model of a biological system able to explain some experimentally observed phenomena. EC have been widely applied to the latter problem [8–11]. Nevertheless, only a few works have exploited a PN representation of individuals in EC methods [12], probably because of the difficulty in defining proper genetic operators and the intrinsic limitations of standard PNs.

In this work we introduce the *Evolutionary Petri Net* (EPN), which is based on an extended class of PNs, called *Resizable Petri Net*, and two genetic operators, crossover and mutation, that altogether give a foundation for the development of a robust evolutionary algorithm for the optimization, the automatic simplification and the reverse engineering of PNs. Due to space limits, we briefly introduce a potential application of EPNs to face one of the most challenging problems in Systems Biology, i.e., the reverse engineering of biochemical reaction networks.

2 Evolutionary Computation

Evolutionary Computation (EC) exploits the Darwinian evolution theory to solve complex problems [1]. Many bio-inspired EC methods have been proposed (e.g., Genetic Algorithms (GA) [13], Particle Swarm Optimization [14]), all sharing the following common traits: *(i)* they exploit a population $\mathcal{P}$ of randomly generated individuals, i.e., the candidate solutions; *(ii)* $\mathcal{P}$ evolves, generation after generation, thanks to an iterative process that employs random modifications of the individuals; *(iii)* the individuals able to solve the problem better than the other candidate solutions are conserved and promoted during the evolutive process; *(iv)* to the aim of discriminating the best solutions in $\mathcal{P}$, a *fitness function* quantifies the quality of each individual; *(v)* the process is executed iteratively until a termination criterion is met.

In the particular case of GAs, the individuals are encoded as fixed-length strings (the *genome*), composed by concatenating symbols from a finite alpha-

bet, and the evolutionary process is driven by three operators: selection, crossover and mutation. The selection mechanism introduces an evolutionary pressure on the population, in order to make the individuals compete and adapt according to the fitness function that characterizes the problem in hand; selection methods are beyond the scope of the present paper, a comprehensive review is available in [15]. The crossover operator simulates the exchange of genetic material occurring during biological reproduction, by creating two offspring from the union of substructures belonging to two selected parents, aiming at the recombination of two promising parents, in order to improve the average quality of the population. The individuals undergo the crossover mechanism with a certain probability; alternatively, individuals are identically copied into $\mathcal{P}$. In both cases, the mutation operator can randomly modify these individuals – with a low probability – by changing, for instance, symbols in its genome. The mutation mechanism is used for the exploration of the search space, being the only operator able to introduce new genetic material in the population.

GAs have shown the ability of efficiently solving a plethora of complex problems, but all these results represent specific solutions to particular instances of a generic problem. In contrast, Genetic Programming (GP) [16] is an extension of GA in which the individuals are computer programs, generally encoded as tree data structures where the inner nodes are functions, and the leaves (or terminal nodes) are constant values or variables. Because of this peculiar representation, GP can identify the optimal solution for whole classes of problems, instead of specific instances. The functioning of GP is conceptually identical to GA and shares all the common traits of evolutionary algorithms, but it presents some slight difference with respect to GAs. Since the individuals are represented as trees, they are not fixed in height and they are exposed to *bloating*, that is, the uncontrolled growth of individuals, generation after generation. A maximum height D for the individuals can be defined, but this might negatively affect the evolution of the population. Moreover, the (unknown) actual height of the tree corresponding to the optimum might be greater than D.

GP has been extended to support individuals encoded as graphs [17], leading to the definition of many crossover operators, which are supposed to work with individuals whose number of nodes is different. For some applications, the individuals of GP may be represented by even more complex data structures like bipartite graphs, which describe in a more natural way the elements of a complex system and their mutual interactions. Anyway, the development of robust genetic operators, able to perform the evolution of such graphs, is challenging.

The set of terminal nodes of GP is generally defined at the beginning of the evolutionary process, meaning that the interacting elements of the complex system under investigation are well known: a condition that, sometimes, is not realistic. In this work we propose a novel methodology, inspired by GP, in which individuals are encoded as multi-partite graphs which, at the end of the evolutionary process, represent putative bipartite graphs, namely Petri Nets. Our methodology is able to automatically determine the best fitting size for the in-

ferred network, by modifying the number of nodes of the candidate solutions during the evolutionary process by means of a special mutation operator.

3   Evolutionary Petri Nets

Petri Net (PN) is a modeling formalism for distributed, asynchronous and concurrent systems [2], which is defined as a weighted, directed, bipartite graph consisting of two kinds of nodes: the nodes representing the state (or conditions) of the system, called *places* and denoted by circles, and the nodes representing *transitions* (or events) between places, denoted by bars. PNs are extensively exploited for the simulation and analysis of the structural and behavioral properties of complex systems. Basic notions and notations of PNs can be found in [18]. To the aim of developing an evolutionary methodology whose candidate solutions are based on PNs, we hereby propose an extension of the conventional PN formalism, called *Evolutionary Petri Net* (EPN). EPNs provide a conceptual framework for the representation of candidate solutions, and embed robust and consistent genetic operators.

Before introducing the EPN formalism, we define a *Resizable Petri Net* (RPN) as a 9-tuple $\xi = (P, P^h, T, T^h, F, W, M_0, O_{pre}, O_{post})$ where:

- $P = \{p_1, \ldots, p_m\}$ is a finite set of places;
- P^h is the set of hidden places, such that $P \cap P^h = \emptyset$;
- $T = \{t_1, \ldots, t_n\}$ is a finite set of transitions;
- T^h is the set of hidden transitions, such that $T \cap T^h = \emptyset$ and $(P \cup P^h) \cap (T \cup T^h) = \emptyset$;
- $F \subseteq \big((P \cup P^h) \times (T \cup T^h)\big) \bigcup \big((T \cup T^h) \times (P \cup P^h)\big)$ is the set of arcs;
- $W : F \to \mathbb{N}$ is a weight function, which associates a non-negative integer value to each arc;
- $M_0 : P \to \mathbb{N}$ is the initial marking of non-hidden places of the net (all hidden places have zero tokens, initially);
- $O_{pre} \in \mathbb{N}$ is the maximum pre-order allowed in the RPN, that is, for each $t \in (T \cup T^h)$

$$\sum_{\substack{p \in {}^\bullet t \\ p \in (P \cup P^h)}} W(p, t) \leq O_{pre}; \tag{1}$$

- $O_{post} \in \mathbb{N}$ is the maximum post-order allowed in the RPN, that is, for each $t \in (T \cup T^h)$

$$\sum_{\substack{p \in t^\bullet \\ p \in (P \cup P^h)}} W(t, p) \leq O_{post}. \tag{2}$$

Differently from a traditional PN, a RPN is composed of a fixed number of places ($|P| = m$) and transitions ($|T| = n$), together with a variable number of hidden places and transitions in the sets P^h and T^h, respectively, whose cardinalities can change during the evolutionary process because of the application of the genetic operators. Two examples of RPN are depicted in Figure 1a; unlike

other similar works on dynamically reconfigurable PNs [18], or on virtual PNs [19], RPNs are modified by "exogenous" mutations that can arbitrarily introduce new hidden places and new hidden transitions, where hidden transitions can represent events involving both elements from P and P^h. In RPNs we explicitly separate the sets P and T from the hidden (modifiable) sets P^h and T^h in order to make use of the available, consolidated domain knowledge of the system under investigation as static (i.e., non modifiable) places and transitions. In the case of zero-knowledge on the interaction of the elements of the net, the set P is populated with those elements that are known to be part of the system and whose data can be exploited by the fitness function, while $T = \emptyset$.

Conversely, hidden places and transitions are exploited by the genetic operators of EPN in order to explore alternative and more complex topologies of the net. Since hidden (i.e., modifiable) sets are dynamically modified and evaluated by the EC algorithm, EPNs allow the optimization of an existent PN according to some given constraints; moreover, they allow the automatic discovery of simplified or more efficient alternative models. These tasks can be accomplished by initializing the system of interest as a "fully-hidden" RPN (that is, $P = T = \emptyset$), and letting the evolutionary algorithm explore the space of alternative models.

The RPN formalism includes the pre- and post-order conditions (Equations 1 and 2) for multiple reasons. First, they help to reduce the bloating phenomenon of GP by limiting the number of arcs. Secondly, they avoid the possibility of a convergence to degenerate or overfitting solutions, represented by completely connected PNs, by limiting the weights of in- and out-going arcs of transitions. This can also be used to limit the search space, by excluding *a priori* unfeasible topologies. However, both pre- and post-order conditions are optional and can be excluded from the RPN by setting $O_{pre} = O_{post} = \infty$.

The modifiers of RPNs, exploited by the evolutionary algorithm, are two classic genetic operators: crossover and mutation. Given the space Ξ of all possible RPN topologies, we can define an *Evolutionary Petri Net* (EPN) as a triple $E = (\xi, \chi, \mu)$ where:

- $\xi \in \Xi$;
- $\chi : (\Xi \times \Xi) \to (\Xi \times \Xi)$ is the crossover operator which modifies two RPNs ξ and $\bar{\xi}$, where ξ and $\bar{\xi}$ are such that $P_\xi = P_{\bar{\xi}}$;
- $\mu : \Xi \cup \{p_{in}, t, p_{out}\} \to \Xi$ is the mutation operator, where $\{p_{in}, t, p_{out}\}$ is a triple consisting of two places p_{in} and p_{out} and a transition t, namely $p_{in}, p_{out} \in (P \cup P^h) \cup P^\infty$ and $t \in (T \cup T^h) \cup T^\infty$, where P^∞ and T^∞ are infinite sets of places and transitions, such that $P^\infty \cap (P \cup P^h) = \emptyset$ and $T^\infty \cap (T \cup T^h) = \emptyset$.

The functioning of χ and μ is described in the next sections, where we denote by $\xi(\tau)$, $\tau \in \mathbb{N}$, the RPN at the τ-th generation of the evolutionary process.

3.1   The Crossover Operator

The crossover mechanism implements the exchange of genetic material between two RPNs, in order to generate new offspring which inherit the best substructures

of the parents. Various crossover mechanisms specifically designed for graphs have been proposed in literature [20, 21], in particular to tackle the complex case of two networks with a different number of nodes [17] but, to the best of our knowledge, no specific work exists about the crossover between bipartite graphs or, more specifically, PNs. Indeed, the crossover between two PNs is supposed to identify some substructures in each graph, "detach" them from one parent graph and "attach" them into the other, and viceversa, while keeping the consistency of both graphs. The difficulties arise in: *(i)* how to characterize a substructure; *(ii)* how to "detach" it from a graph; *(iii)* how to "attach" it to a new graph, considering that there is not a direct correspondence between the elements belonging to different sets of hidden places. The last issue is extremely relevant, because it determines the ability of EPNs to transferring a precise functionality from a RPN to its offspring.

Our proposal for a crossover mechanism of two RPNs $\xi(\tau), \bar{\xi}(\tau) \in \Xi$, is named *sticky crossover* (SC). SC works on hidden transitions as follows: a transition $t_\chi \in T^h$ ($\bar{t}_\chi \in \bar{T}^h$, respectively) is randomly selected in the RPN ξ ($\bar{\xi}$) (the red nodes in Figure 1b), and the substructures consisting of the preset and postset of t_χ ($\bar{t}_\chi$) (dotted lines) are exchanged between ξ and $\bar{\xi}$. We denote by $P^+ = ({}^\bullet t_\chi \cup t_\chi^\bullet) \cap P^h$ ($\bar{P}^+$, respectively) the set of hidden places contained in the substructure connected to t_χ ($\bar{t}_\chi$), that is added to $\bar{\xi}$ (ξ); $F^+ = \{(t_\chi, p), (p, t_\chi) \in F \mid p \in P^+\}$ ($\bar{F}^+$) denotes the set of arcs belonging to this substructure.

To the aim of determining the attachment points for the moving substructures, we randomly select two input places $p_b \in {}^\bullet t_\chi$ and $\bar{p}_b \in {}^\bullet \bar{t}_\chi$, and two output places $p_e \in t_\chi^\bullet$ and $\bar{p}_e \in \bar{t}_\chi^\bullet$ (Figure 1c); p_b, p_e, $\bar{p}_b$, and $\bar{p}_e$ will be used as "attachment" points for the incoming substructure, that is, the substructure identified by transition t_χ is attached to $\bar{\xi}$, in such a way that $\bar{p}_b$ is replaced by p_b, $\bar{p}_e$ is replaced by p_e, and viceversa (see Figure 1d).

Formally, the crossover mechanism acts on the sets of places, transitions and arcs of the RPN as follows:

$$\bar{P}^h = \bar{P}^h \cup P^+ \setminus \{\bar{p}_b, \bar{p}_e\}; \quad P^h = P^h \cup \bar{P}^+ \setminus \{p_b, p_e\};$$
$$\bar{T}^h = \bar{T}^h \cup \{t_\chi\} \setminus \{\bar{t}_\chi\}; \quad T^h = T^h \cup \{\bar{t}_\chi\} \setminus \{t_\chi\};$$
$$\bar{F} = \bar{F} \cup F^+ \setminus \bar{F}^+ \cup \{(\bar{t}, p_b) \mid \bar{t} \in {}^\bullet \bar{p}_b\} \cup \{(p_e, \bar{t}) \mid \bar{t} \in \bar{p}_e^\bullet\} \cup$$
$$\cup \{(p_b, \bar{t}) \mid \bar{t} \in \bar{p}_b^\bullet\} \cup \{(\bar{t}, p_e) \mid \bar{t} \in {}^\bullet \bar{p}_e\} \setminus \{(\bar{t}, \bar{p}_b) \mid \bar{t} \in {}^\bullet \bar{p}_b\} \setminus$$
$$\setminus \{(\bar{p}_e, \bar{t}) \mid \bar{t} \in \bar{p}_e^\bullet\} \setminus \{(\bar{p}_b, \bar{t}) \mid \bar{t} \in \bar{p}_b^\bullet\} \setminus \{(\bar{t}, \bar{p}_e) \mid \bar{t} \in {}^\bullet \bar{p}_e\};$$
$$F = F \cup \bar{F}^+ \setminus F^+ \cup \{(t, \bar{p}_b) \mid t \in {}^\bullet p_b\} \cup \{(\bar{p}_e, t) \mid t \in p_e^\bullet\} \cup$$
$$\cup \{(\bar{p}_b, t) \mid t \in p_b^\bullet\} \cup \{(t, \bar{p}_e) \mid t \in {}^\bullet p_e\} \setminus \{(t, p_b) \mid t \in {}^\bullet p_b\} \setminus$$
$$\setminus \{(p_e, t) \mid t \in p_e^\bullet\} \setminus \{(p_b, t) \mid t \in p_b^\bullet\} \setminus \{(t, p_e) \mid t \in {}^\bullet p_e\}.$$

The weights of the arcs exchanged during the crossover process are not modified. Only hidden places are moved between RPNs during the crossover process, because each RPN contains an identical set of fixed places, so that there is a biunivocal correspondence between the elements in P and P^h. For this reason, for each place $p \in ({}^\bullet t_\chi \cup t_\chi^\bullet) \cap P$, where t_χ is the transition selected for crossover,

(a) Parents RPNs

(b) Step #1: selection of the transitions t_χ and $\bar{t}_\chi$

(c) Step #2: random selection of p_b, p_e, $\bar{p}_b$ and $\bar{p}_e$ nodes

(d) Step #3: exchange of substructures
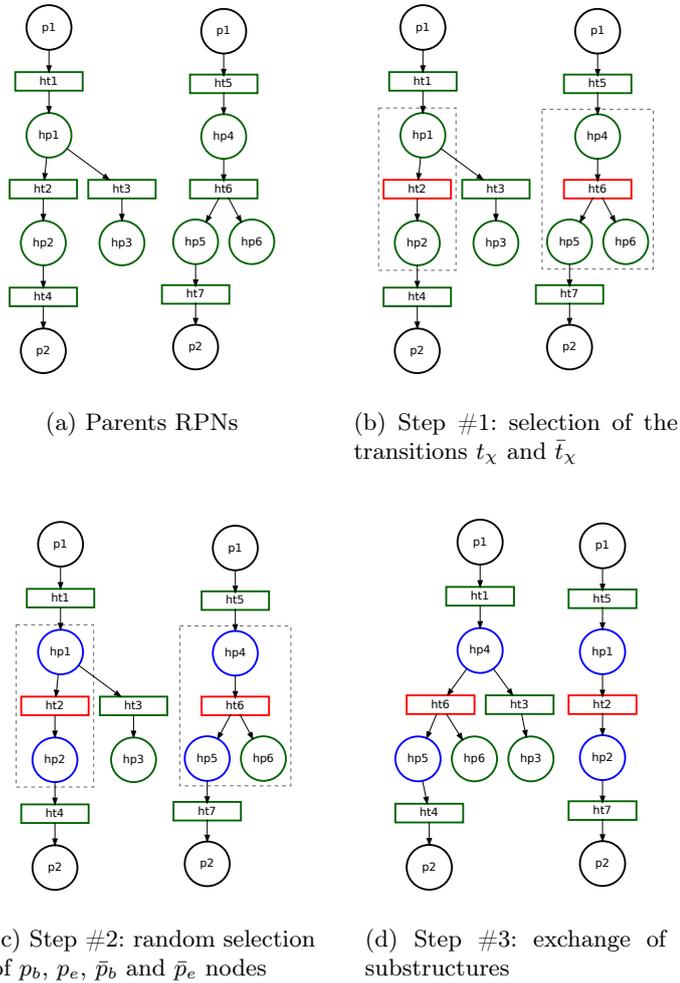
Fig. 1. Example of *sticky crossover* between two RPNs. (a) From each parent RPN (b) a hidden transition is randomly selected (red nodes ht_2 and ht_6), identifying the substructures that will be exchanged between the RPNs (dotted gray line). (c) One random place from the preset and one from the postset of ht_2 and ht_6 are selected (the blue nodes hp_1, hp_2, hp_4 and hp_5 that correspond to p_b, p_e, $\bar{p}_b$ and $\bar{p}_e$, respectively). (d) The substructures are exchanged between the RPNs and attached by replacing the respective blue nodes, thus yielding the two offspring. For the sake of compactness we do not report the weights of arcs when they are equal to 1.

we let $\bar{F} = \bar{F} \cup \{(\bar{p}, t_\chi)\} \cup \{(t_\chi, \bar{p})\}$ and viceversa. It is important to clarify that the elements in P^h and T^h are "anonymous", that is, they do not share any semantics between different RPNs: if a hidden element is transferred from a RPN to another during a crossover, it is considered a new unknown element with respect to the already existing elements.

(a) Parents RPNs   (b) RPNs after crossover

Fig. 2. Example of *sticky crossover* breaking up one RPN, thus creating a separate component (the hidden transition ht_8 in Subfigure (b), and its pre- and postsets).
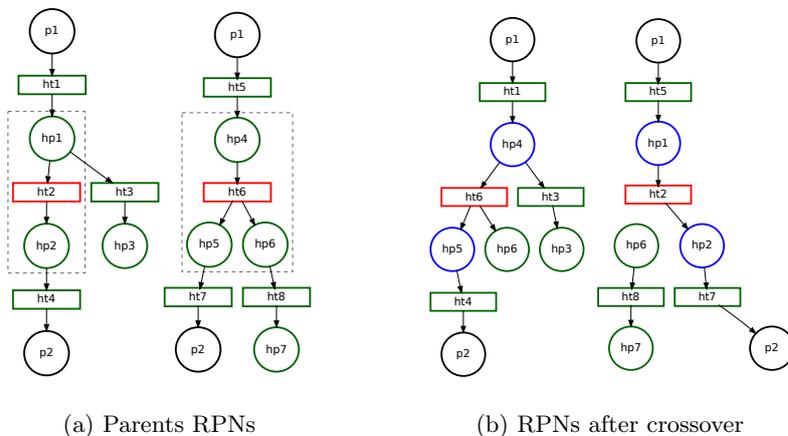
If the SC involves $p \in P^h$ and $\bar{p} \in \bar{P}$ as the selected "attachment" places of the crossover then, in the exchange of the substructure from ξ to $\bar{\xi}$, place $\bar{p} \in \bar{P}$ in $\bar{\xi}$ remains unchanged, while in the exchange from $\bar{\xi}$ to ξ, place $p \in P^h$ in ξ is replaced by $\bar{p}$. So doing, ξ will result in a consistent RPN, since $P = \bar{P}$.

SC is a convenient crossover operator for many reasons: *(i)* it allows the crossover between two arbitrary RPNs, regardless the cardinality of the sets $P^h, \bar{P}^h$, which can vary during the genetic evolution; *(ii)* the pre- and post-order of the transitions of the offspring (as defined in Equations 1 and 2) are automatically conserved; *(iii)* the "directionality" of transitions is preserved, since SC swaps substructures whose presets are still presets, and elements of postsets are still postsets. Nevertheless, the SC has two drawbacks: *(i)* considering a substructure identified by a transition t_χ, if $\tilde{t} \neq t_\chi$ is connected to a place $p \in ({}^\bullet t_\chi \cup t_\chi^\bullet), p \notin \{p_e, p_b\}$, then the SC breaks the RPN leaving a separate component (see Figure 2); *(ii)* the SC allows the exchange of a single transition between two RPNs, which could have a little impact on large networks.

The second issue can be solved with two strategies: the first is to exploit some graph visiting algorithm (i.e., breadth- or depth-first) and extend the substructure accordingly; the second strategy is to determine a value $\rho \in \mathbb{N}$ such that $1 \leq \rho \leq \min\{|T^h|, |\bar{T}^h|\}$, and to repeat the crossover on ρ different transitions, that is, to randomly create two vectors of indexes $i_1, \ldots, i_\rho$ and $j_1, \ldots, j_\rho$ (with $i_k, j_k \in \mathbb{N}$, $k = 1, \ldots, \rho$), and then apply the crossover operator on each couple $(t_{i_k}, \bar{t}_{j_k})$. The first strategy allows to exchange multiple transitions that have been causally connected according to the chosen graph visiting algorithm, whilst the second one allows to exchange multiple independent transitions.

In the implementation of EPNs for real case applications, the computational complexity of applying the crossover operator is, in the worst case, $O(|F|^2 \cdot |P|)$. However, the use of a hash function might yield a reduction of the computational cost, in the average case, to $O(|P|)$.

3.2   The Mutation Operator

The mutation operator modifies the structure of a RPN $\xi(\tau)$ in Ξ, or the properties of its places and arcs (i.e., capacity and weight), by acting on a single, randomly chosen hidden transition $t \in T^h$. In particular, the mutation operator associates $\xi(\tau)$ to a new consistent RPN $\xi(\tau + 1)$, according to a specified triple $\{p_{in}, t, p_{out}\}$. The rationale behind this triple is to provide new genetic material, that is, to modify the topology of the RPN; the functioning of the mutation operator for all the possible cases of $\{p_{in}, t, p_{out}\}$ is summarized in Table 1. After the application of any mutation case (except #8), we set $F = F \cup (p_{in}, t) \cup (t, p_{out})$ and $W(p_{in}, t) = W(t, p_{out}) = 1$. Cases #7 and #8 are particular and deserve a detailed explanation.

Case #7 introduces a brand new transition which is "disconnected" from the rest of the network, since it exploits new places that are not used by any other transition. So doing, the dynamics of these places, that is, the succession of their markings as a consequence of the firings, is independent from the rest of the RPN. In other words, Case #7 is a "silent" modification of the topology of the RPN, that will not have a direct impact on its behavior. Nevertheless, a further application of a genetic operator to the mutated RPN may connect this latent transition to the main net component, thus conditioning the behavior of the whole RPN.

Case #8 does not introduce any new genetic material. This operator is used to change the capacities $K(p)$ of randomly selected places $p \in P$, i.e. $K(p) = rnd$, where rnd is a random number sampled from the uniform distribution $(0, K_{max}]$ (where K_{max} is the maximum capacity for the places in the RPN). Alternatively, the mutation can modify the weight of an arc; for instance, given the arc weight $W(p, t)$, its value can be updated as $W(p, t) = W(p, t) \pm 1$. It is worth noting that also Case #8 can even lead to a modification in the structure of the RPN, whenever an arc weight is set to zero (i.e., the arc is removed). As a consequence, isolated hidden places or hidden transitions, which represent sources or sinks, can be introduced in the RPN after Case #8 is applied and must be removed. Therefore, after the application of this operator, the consistency of the RPN must be verified, regarding each place p and transition t involved in the mutation: *(i)* if $p \in P^h$ and $\nexists \{(p, t), (t, p)\} \ \forall t \in T \cup T^h$, then $P^h = P^h \setminus \{p\}$; *(ii)* if $t \in T^h$ and $\nexists \{(t, p), (p, t)\} \ \forall p \in P \cup P^h$, then $T^h = T^h \setminus \{t\}$.

As a final step of the evolution process, pre- and post-order conditions of the offspring need to be verified, since the described mutations might produce a putative RPN whose topology does not satisfy Equations 1 and 2. In such a case, a further modification of the weights of the ingoing and/or outgoing arcs of a transition t is required. In particular, a randomly selected input (output, respectively) place of transition t is modified so that $W(\tilde{p}, t) = W(\tilde{p}, t) - 1$ (and/or $W(t, \tilde{p}) = W(t, \tilde{p}) - 1$), where $\tilde{p} \in {}^{\bullet}t$ ($\tilde{p} \in t^{\bullet}$, respectively). It is clear that if $W(x, y) = 0$, then the corresponding arc (x, y) is removed from F. This operation is repeated until the RPN respects all the pre- and post-order conditions.

It is worth noting that these two mechanisms implicitly allow mutation to delete places, thus reducing the size of the RPN.

Table 1. Effect of the mutation operator on a Resizable Petri Net $\xi(\tau)$.

No.	Condition	$\xi(\tau + 1)$	Semantics of the mutation
1.	$p_{in} \notin P \cup P^h$ $t \in T \cup T^h$ $p_{out} \in P \cup P^h$	$P^h = P^h \cup \{p_{in}\}$ $P^\infty = P^\infty \setminus \{p_{in}\}$	A new hidden place p_{in} is created and added to P^h; transition t is extended to have a new input place p_{in}
2.	$p_{in} \in P \cup P^h$ $t \in T \cup T^h$ $p_{out} \notin P \cup P^h$	$P^h = P^h \cup \{p_{out}\}$ $P^\infty = P^\infty \setminus \{p_{out}\}$	A new hidden place p_{out} is created and added to P^h; transition t is extended to have a new output place p_{out}
3.	$p_{in} \in P \cup P^h$ $t \notin T \cup T^h$ $p_{out} \in P \cup P^h$	$T^h = T^h \cup \{t\}$ $T^\infty = T^\infty \setminus \{t\}$	A new hidden transition t is created and added to T^h; transition t is connected to the existing input and output places p_{in} and p_{out}, respectively
4.	$p_{in} \notin P \cup P^h$ $t \notin T \cup T^h$ $p_{out} \in P \cup P^h$	$P^h = P^h \cup \{p_{in}\}$ $T^h = T^h \cup \{t\}$ $P^\infty = P^\infty \setminus \{p_{in}\}$ $T^\infty = T^\infty \setminus \{t\}$	A new hidden transition t is created and added to T^h; a new hidden place p_{in} is created and added to P^h; transition t is connected to new input place p_{in} and to an existing output place p_{out}
5.	$p_{in} \in P \cup P^h$ $t \notin T \cup T^h$ $p_{out} \notin P \cup P^h$	$T^h = T^h \cup \{t\}$ $P^h = P^h \cup \{p_{out}\}$ $P^\infty = P^\infty \setminus \{p_{out}\}$ $T^\infty = T^\infty \setminus \{t\}$	A new hidden transition t is created and added to T^h; a new hidden place p_{out} is created and added to P^h; transition t is connected to an existing input place p_{in} and to the new output place p_{out}
6.	$p_{in} \notin P \cup P^h$ $t \in T \cup T^h$ $p_{out} \notin P \cup P^h$	$P^h = P^h \cup \{p_{in}, p_{out}\}$ $P^\infty = P^\infty \setminus \{p_{in}, p_{out}\}$	Two new hidden places p_{in} and p_{out} are created and added to P^h; transition t is connected to p_{in} as input place and to p_{out} as output place
7.	$p_{in} \notin P \cup P^h$ $t \notin T \cup T^h$ $p_{out} \notin P \cup P^h$	$P^h = P^h \cup \{p_{in}, p_{out}\}$ $T^h = T^h \cup \{t\}$ $P^\infty = P^\infty \setminus \{p_{in}, p_{out}\}$ $T^\infty = T^\infty \setminus \{t\}$	A new hidden transition t is created and added to T^h; two new hidden places p_{in} and p_{out} are created and added to P^h; transition t is connected to the new input place p_{in} and to the new output place p_{out}
8.	$p_{in} \in P \cup P^h$ $t \in T \cup T^h$ $p_{out} \in P \cup P^h$	$P^h = P^h$ $T^h = T^h$ $P^\infty = P^\infty, T^\infty = T^\infty$	No new genetic material is introduced. Either p_{in} or p_{out} is randomly chosen; then, its capacity or the weight of the arc connecting it to t is modified

The computational complexity of the mutation operator is, for the worst case, $O(|P|)$. However, as in the case of crossover, the use of a hash function allows a reduction of the computational cost of this operator, in the average case, to $O(C)$ where, in general, $C < |P|$.

4   Toward the Application of EPNs for the Reverse Engineering of Biochemical Reaction Networks

In this section we sketch the theoretical basis of a potential application of EPNs in the context of Systems Biology, for the reverse engineering (RE) of biochemical interaction networks. This problem consists in the identification of the network

of reactions that describe the physical interactions among the chemical species occurring in the system (genes, proteins, metabolites, etc.). These networks can be defined thanks to human expertise, by relying on some pre-existing knowledge, although most of the times the exact molecular mechanisms occurring in living cells are not known and cannot be fully understood by means of laboratory experiments only. This problem therefore demands the development of automatic RE methods, so that a plausible network of biochemical reactions – able to reproduce some given experimental observations – can be determined in faster and inexpensive ways. Since these networks are usually kinetically parameterized, they can also be exploited to analyze the dynamics of the system under different conditions.

Here, we briefly describe the modeling of biochemical systems by means of PNs and then present a possible strategy based on EPNs for solving the RE problem, which shows the feasibility of our novel methodology. A biochemical network η can be modeled by means of a set of chemical species $\mathcal{S} = (S_1, \ldots, S_U)$, involved in a set of chemical reactions $\mathcal{R} = (R_1, \ldots, R_Z)$. Each reaction $R_\zeta \in \mathcal{R}, \zeta = 1, \ldots, Z$, is defined as $R_\zeta : a_{\zeta 1} \cdot S_1 + \ldots + a_{\zeta U} \cdot S_U \xrightarrow{k_\zeta} b_{\zeta 1} \cdot S_1 + \ldots + b_{\zeta U} \cdot S_U$, where $a_{\zeta i}, b_{\zeta i} \in \mathbb{N}$ are the stoichiometric coefficients of R_ζ, and $k_\zeta \in \mathbb{R}^+$ is the kinetic constant associated to R_ζ. The species occurring on the left-hand (right-hand) side of R_ζ are called reagents (products, respectively).

Because of their bipartite graph structure, PNs offer an ideal conceptual framework for the modeling of biochemical networks [22] defined as a set of reactions in the form of R_ζ. To this aim, a transformation of a network η into a corresponding PN (and viceversa) needs to be defined. Briefly, a mapping $f : \mathcal{S} \to P$ can be used to associate the species to the places of a PN, where the transitions represent the chemical reactions (i.e., $g : \mathcal{R} \to T$) and where the weights correspond to the stoichiometry of the reagents and products of each reaction (i.e., $v_r : a_{\zeta i} \to W(p_i, t_\zeta)$, $v_p : b_{\zeta i} \to W(t_\zeta, p_i)$). According to these mappings, it is straightforward to show that the PN on the left in Figure 1a represents the following set of reactions: $\{R_1^h : S_1 \to S_1^h, R_2^h : S_1^h \to S_2^h, R_3^h : S_1^h \to S_3^h, R_4^h : S_2^h \to S_2\}$. The number of tokens (given as discrete or continuous value) in each place represents the molecular amount (given as number of molecules or concentration, respectively) of the corresponding chemical species, so that M_0 represents the initial state of the biochemical system.

The PN representation of a biochemical reaction network allows the investigation of its structural features by means of theoretical approaches [22, 23]; in addition, PNs can be easily extended with timed delays [24], or to incorporate quantitative information (e.g., reaction rates, concentration levels) to the aim of investigating the dynamic evolution of biochemical systems [25]. Kinetic parameters can also be associated to the reactions, in order to derive the probability of each reaction to occur [26], or to convert the system into a set of coupled ordinary differential equations; in the latter case, places contain continuous values corresponding to the concentration of the chemical species associated to each place. Once that a model η of a biochemical network is defined in terms of a fully parameterized extended PN, according to the available domain knowledge, its

dynamic behavior can be investigated. Anyway, when the domain knowledge is incomplete, uncertain or completely missing, a (fully parameterized) PN model of a biochemical system cannot be constructed. In such a case, it is necessary to perform the RE of the biochemical reaction network, investigating the unknown reactions and chemical species that are responsible for the observed phenomena.

Many works perform the RE by means of evolutionary techniques [12, 27, 28]. One limitation of many of these techniques is that the individuals are modeled by means of data structures that are not ideal to describe reaction networks; in addition, many constraints and consistency controls must be introduced to control the quality and validity of the inferred network. The strongest limitation of all these methods is that the cardinality of $\mathcal{S}$, that is, the number of chemical species that are present in the system, is assumed to be known and kept fixed during the optimization. This is generally a strong assumption, that may be justified only if the biochemical system is very well known (but, in such a case, the network should be known as well) or when laboratory experiments can yield this information with a certain precision. As a matter of fact, in most cases the exact number and nature of the chemical species involved in the system – including the intermediate complexes formed by the chemical bonds among various molecules – is unknown; anyway, this is a fundamental information to properly carry out the RE of the system. Thanks to the flexibility of hidden places and transitions, the use of RPNs to represent the candidate solutions gives to EPN the possibility to explore much more possibilities than the traditional RE approaches that exploit a fixed number of chemical species, thus leading to the formulation of new hypotheses for the structure of the biochemical network that should then be validated with *ad hoc* laboratory experiments. In this context, hidden places in the RPN can be exploited to represent some molecular species or complexes which are necessary to reproduce the expected behavior of the biochemical system, but that have not been yet identified with experimental techniques; similar considerations hold for hidden transitions, which can represent molecular interactions that are not known from a biochemical point of view, but that might yield a better system functioning (in terms of the specified fitness function, and according to the available experimental data).

The application of an EPN-based methodology for the RE of biochemical systems is similar to GP. At first, a population $\mathcal{P}$ of RPNs is generated according to the available domain knowledge: when the information comes from established biological knowledge, the reactions are modeled using the sets P and T; on the contrary, when the information is uncertain, the network is modeled by means of hidden places and transitions. At generation $\tau = 0$, since the individuals are all identical, they undergo a preliminary mutation. During each generation, the best individuals are selected according to a specified fitness function, and the EPN genetic operators are applied to yield new offspring. The fitness function can be defined, for instance, by comparing a set of $\Delta \in \mathbf{N}$ experimental samples of the $|P| = m$ non-hidden chemical species (places in P) against the dynamics generated through simulations [29], exploiting the set of reactions that the RPN represents: $fitness(\eta) = \sum_{\delta=1}^{\Delta} \sum_{i=1}^{m} |\mathbf{Q}^{p_i}(t_\delta) - \mathbf{L}_\eta^{p_i}(t_\delta)|$, where $\mathbf{Q}^{p_i}(t_\delta)$ and

$\mathbf{L}^{p_i}_{\eta}(t_\delta)$ denote, respectively, the experimental and simulated quantity (concentration or molecular amount) of the p_i-th chemical species at time t_δ.

To the aim of facilitating the evolutionary process and helping the generation of biologically meaningful candidate solutions, in RE we can set $O_{pre} = 2$ to force the evolution of at most second-order reactions, since higher-order reactions have a probability to occur almost equal to zero, requiring the simultaneous collision of three or more reactant molecules. This choice helps the EPN functioning, since it strongly reduces the search space Ξ, but at the same time it does not pose any limitation to the practical applicability of the RE methodology, as higher-order reactions can be mimicked through a cascade of consecutive reactions of lower order. During the evolutionary process, the EPN explores this search space, and eventually a best individual $\mathcal{I} \in \mathcal{P}$ emerges, representing a consistent RPN that fits with all the observed phenomena: $\mathcal{I}$ is finally returned as the result of the RE. Due to space limits, we can only sketch how the crossover operator acts in determining a biochemical reaction network. Consider, for instance, the individuals ξ (left) and $\bar{\xi}$ (right) shown in Figure 1a: the crossover swaps the reactions R^h_2 in ξ and R^h_6 in $\bar{\xi}$, so that the chemical reaction R^h_2, renamed R^h_6 after the crossover, has an additional product (i.e., S^h_6, left side of Figure 1d). On the contrary, the reaction R^h_6 (renamed R^h_2 after the crossover) in $\bar{\xi}$ loses a product and becomes a simple transformation from one species to another. The consequence of these modifications is that the new reaction network might have a completely different dynamic behavior, and hence a different (hopefully, better) fitness value.

5   Conclusion and Future Work

In this work we presented an extension of the PN formalism, the Resizable Petri Net, which introduces the notion of hidden places and transitions. The RPN represents the basis for the development of an evolutionary algorithm, the Evolutionary Petri Net, whose crossover and mutation operators allow the exploration of the space of all possible RPN topologies, in order to provide a powerful tool for solving complex problems whose solutions can be encoded as PNs; in particular, EPNs has the capability to evolve consistent and meaningful solutions. We described here the theoretical foundations of EPNs; a thorough analysis of this novel evolutive framework will be presented in a future work, in order to introduce and discuss different strategies for the construction of the RPN individuals (according to various real case applications), as well as specific selection mechanisms. The probability distributions that need to be associated to the mutation and crossover operators, and which have a relevant impact on the convergence of the evolutive process of RPNs, will also be discussed in a forthcoming extension of this work.

To the purpose of grounding this framework to practical problems, we provided an example of a potential application of EPNs – the RE of biochemical reaction networks – which might represent a killer-application for our evolutionary methodology. In this context, the fitness evaluation that we proposed relies on the simulation of the candidate solutions, which is possible only if a proper parameterization of the candidate network in available. Hence, the RE problem

is further complicated by the need of a parameter estimation (PE) methodology for the inference of the missing kinetic parameters [30, 31]: to this aim, we are developing an integrated methodology that embeds the PE process in each generation of the RE. A further difficulty of the RE process is due to the fact that different networks can lead to the same dynamic behavior; this problem, known as *undistinguishability* [32], cannot be solved, in general, without additional knowledge. EPNs do not directly mitigate this drawback, even though pre- and post-order conditions allow the reduction of the possible topologies, thus permitting the derivation of meaningful networks that can be then discriminated by domain expertise. Finally, the initial marking of the hidden places was set to zero: a further extension of EPN, in which the initial marking co-evolves with the topology, is under investigation.

References

1. DeJong, K.: Evolutionary computation: a unified approach. The MIT Press (2006)
2. Murata, T.: Petri nets: Properties, analysis and applications. In: Proc. IEEE. Volume 77. (1989) 541–580
3. Thome, H., Nakamura, M., Hachiman, K.: Evolutionary Petri net approach to periodic job-shop scheduling. In: Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC'99). Volume 4., IEEE Computer Society Press (1999) 441–446
4. Saitou, K., Malpathak, S., Qvam, H.: Robust design of flexible manufacturing systems using colored Petri net and genetic algorithm. Journal of Intelligent Manufacturing **13**(5) (2002) 339–351
5. Mauch, H.: Evolving Petri nets with a genetic algorithm. In: Proc. of the 2003 conference on genetic and evolutionary computation (GECCO '03). Volume 2724 of LNCS., Springer (2003) 1810–1811
6. Moore, J., L.W.Hahn: Grammatical evolution for the discovery of Petri net models of complex genetic systems. In: Proc. of the 2003 conference on genetic and evolutionary computation (GECCO '03). Volume 2724 of LNCS., Springer (2003) 2412–2413
7. Kitagawa, J., Iba, H.: Identifying metabolic pathways and gene regulation networks with evolutionary algorithms. In: Evolutionary Computation in Bioinformatics. Morgan Kaufmann (2003) 255–278
8. Koza, J.R., Mydlowec, W., Lanza, G., Yu, J., Keane, M.A.: Reverse engineering of metabolic pathways from observed data using genetic programming. In: Pacific Symposium on Biocomputing. Volume 6. (2001) 434–445
9. Sugimoto, M., Kikuchi, S., Tomita, M.: Reverse engineering of biochemical equations from time-course data by means of genetic programming. BioSystems **80**(2) (2005) 155–164
10. Ando, S., Sakamoto, E., Iba, H.: Evolutionary modeling and inference of gene network. Information Sciences **145**(3) (2002) 237–259
11. Iba, H.: Inference of differential equation models by genetic programming. Information Sciences **178**(23) (2008) 4453–4468
12. Nummela, J., Julstrom, B.A.: Evolving petri nets to represent metabolic pathways. In: Proc. of the 2005 conference on genetic and evolutionary computation (GECCO '05), ACM (2005) 2133–2139
13. Holland, J.: Adaptation in Natural and Artificial Systems. The University of Michigan Press (1975)

14. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. IEEE International Conference on Neural Networks. Volume 4., IEEE (1995) 1942–1948
15. Bäck, T.: Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In: Proc. of the First IEEE Conference on Evolutionary Computation. Volume 1., IEEE (1994) 57–62
16. Koza, J.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press (1992)
17. Katagiri, H., Hirasawa, K., Hu, J., Murata, J.: Comparing some graph crossover in genetic network programming. In: Proc. of the 41st SICE Annual Conference. Volume 2., IEEE (2002) 1263–1268
18. Llorens, M., Oliver, J.: Structural and dynamic changes in concurrent systems: Reconfigurable Petri nets. IEEE Trans. on Computers **53**(9) (2004) 1147–1158
19. Morandin Jr, O., Kato, E.: Virtual Petri nets as a modular modeling method for planning and control tasks of FMS. International Journal of Computer Integrated Manufacturing **18**(2-3) (2005) 100–106
20. Larrañaga, P., Kujipers, C., Murga, R., Inza, I., Dizdarevic, S.: Genetic algorithms for the travelling salesman problem: A review of representations and operators. Artificial Intelligence Review **13**(2) (1999) 129–170
21. Pereira, F., Machado, P., Costa, E., Cardoso, A.: Graph based crossover – a case study with the busy beaver problem. In: Proc. of the Genetic and Evolutionary Computation Conference (GECCO '99). Volume 2. (1999) 1149–1155
22. Chaouiya, C.: Petri net modelling of biological networks. Briefings in Bioinformatics **8**(4) (2007) 210–219
23. Voss, K., Heiner, M., Koch, I.: Steady state analysis of metabolic pathways using Petri nets. In Silico Biology **3**(3) (2003) 367–387
24. Li, C., Ge, Q., Nakata, M., Matsuno, H., Miyano, S.: Modelling and simulation of signal transduction in an apoptosis pathway by using timed Petri nets. Journal of Biosciences **32**(1) (2006) 113–127
25. Goss, P., Peccoud, J.: Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets. PNAS **95**(12) (1998) 6750–6755
26. Gillespie, D.T.: Stochastic simulation of chemical kinetics. Annual Review of Physical Chemistry **58** (2007) 35–55
27. Cho, D., Cho, K., Zhang, B.: Identification of biochemical networks by S-tree based genetic programming. Bioinformatics **22**(13) (2006) 1631–1640
28. Nobile, M.S., Besozzi, D., Cazzaniga, P., Pescini, D., Mauri, G.: Reverse engineering of kinetic reaction networks by means of cartesian genetic programming and particle swarm optimization. In: Proc. IEEE Congress on Evolutionary Computation (CEC2013). (2013, in press)
29. Ghosh, S., Matsuoka, Y., Asai, Y., Hsin, K.Y., Kitano, H.: Software for systems biology: from tools to integrated platforms. Nature Reviews Genetics **12**(12) (2011) 821–832
30. Nobile, M.S., Besozzi, D., Cazzaniga, P., Mauri, G., Pescini, D.: A GPU-based multi-swarm PSO method for parameter estimation in stochastic biological systems exploiting discrete-time target series. In: Proc. EvoBIO 2012. Volume 7246 of LNCS. Springer (2012) 74–85
31. Moles, C.G., Mendes, P., Banga, J.R.: Parameter estimation in biochemical pathways: a comparison of global optimization methods. Genome Research **13**(11) (2003) 2467–2474
32. Szederkenyi, G., Banga, J.R., Alonso, A.A.: Inference of complex biological networks: distinguishability issues and optimization-based solutions. BMC Systems Biology **5**(1) (2011) 177

Identification of Key Regulators in Glycogen Utilization in E. coli Based on the Simulations from a Hybrid Functional Petri Net Model

Zhongyuan Tian[1], Adrien Fauré[1], Hirotada Mori[2], and Hiroshi Matsuno[1,⋆]

[1]Graduate School of Science and Engineering, Yamaguchi University,
1677-1 Yoshida, Yamaguchi-shi, Yamaguchi 753-8512, Japan
{r002wa,afaure}@yamaguchi-u.ac.jp
matsuno@sci.yamaguchi-u.ac.jp
[2]Graduate School of Biological Sciences, Nara Institute of Science and Technology,
8916-5 Takayama, Ikoma, Nara 630-0101, Japan
hmori@gtc.naist.jp

Abstract. Glycogen and glucose are two sugar sources available during the lag phase of *E. coli*, but the mechanism that regulates their utilization is still unclear. Attempting to unveil the relationship between glucose and glycogen, we propose an integrated hybrid functional Petri net (HFPN) model including glycolysis, PTS, glycogen metabolic pathway, and their internal regulatory systems. By comparing known biological results to this model, basic regulatory mechanism for utilizing glucose and glycogen were identified as a feedback circuit in which HPr and EIIA^{Glc} play key roles. Based on this regulatory HFPN model, we discuss the process of glycogen utilization in *E. coli* in the context of a systematic understanding of carbohydrate metabolism.

Keywords: metabolic pathway, glycogen, hybrid functional Petri net, PTS, HPr and EIIA^{Glc} proteins, gene regulation

1 Introduction

The carbohydrate pathway occupies a central position in a cell's metabolism. In our previous paper [1], we proved that glycogen plays an important role in the lag phase of *E. coli*. But how the cell regulates the utilization of these carbon sources, intracellular glycogen and extracellular glucose, was yet to be clarified. In a cell, glycogen works as a sugar store or a sugar supply depending on different nutrition conditions, under the regulation of enzymes expressed by *glg* gene clusters (*glgBXCAP*) [2]. Uptake of extracellular glucose is conducted via the phosphotransferase system (PTS) in *E. coli*, whose enzymes are expressed from two operons, *ptsHIcrr* and *ptsG* [3]. Although several shared regulators of PTS and glycogen metabolism, such as ppGpp, Cra, CsrA and cAMP/CRP, have

---

⋆ Corresponding author.

been studied [2, 4–10], a basic regulation system for the utilization of glucose and glycogen has not been studied yet.

Computer modeling is a general and effective method for the integration of biological systems. The purpose of this paper is to construct an integrated model for the systematic understanding of the carbohydrate pathway system of *E. coli*. In this work we first transposed into the hybrid functional Petri net (HFPN) [11] two published models controlling different aspects of the central carbohydrate pathway: glycolysis and pentose phosphate (PP) pathway [12], and PTS [13]. These models have then been assembled together with a newly developed general mass action model of the glycogen metabolic pathway into a single comprehensive HFPN model.

By applying metabolic regulatory mechanisms in our combined HFPN model, a basic control system regulating the utilization glucose and glycogen was identified, in which HPr::GlgP complex [14–16], EIIA^{Glc}&cAMP system [8, 17], EI dimerization [18, 19], FDP&Cra mutual feedback [6], HPr subcellular location [2, 16, 20] etc. are working as regulators. In this paper, with the support of simulation results from the HFPN model, we clarify functions of HPr and EIIA^{Glc} as key regulators of glucose and glycogen utilization.

2   Molecular mechanism for regulating glucose and glycogen utilization

2.1   Regulators

Fig. 1 shows possible regulators that control glucose and glycogen utilization, in which these components are classified into five levels, labeled 0-4, according to regulation "source" and "target". These regulators constitute a circuit that gives a whole view of the regulation of glucose and glycogen utilization as shown in Fig. 2 .

Level-1 P2P (Regulation from protein to protein). HPr regulates consumption of both glucose and glycogen by its phosphorylation state and concentration [8]. HPr is a member enzyme of PTS, which is involved in glucose uptake in *E. coli*. (From here on, PHPr denotes the phosphorylated form of HPr, HPr denotes the unphosphorylated form, and (P)HPr denotes either phosphorylated or unphosphorylated Hpr). The phosphorylation state of the (P)HPr::GlgP complex controls glycogen decomposition. The glycogen decomposing activity of HPr::GlgP is about five times higher than that of PHPr::GlgP [14]. In PTS, HPr transfers phosphate groups from (P)EI to (P)EIIA^{Glc}. Thus (P)HPr regulates the speed of carbohydrate decompositions from both glycogen and glucose.

Level-2 F2P (Regulation from molecule flux speed to protein). EI dimerization is thought to be the limiting step in the process of PTS transfer phosphate from PEP to G6P via PTS, which is regulated by PEP [19]. Whether EI acts as a dimer or a monomer to transfer phosphate from PEP to HPr is still under discussion [19, 21, 22]. Different phosphorylation states of HPr result from the balance of phosphate influx into PTS from PEP and outflux to G6P, in which phosphate influx is under the regulation of EI dimerization.
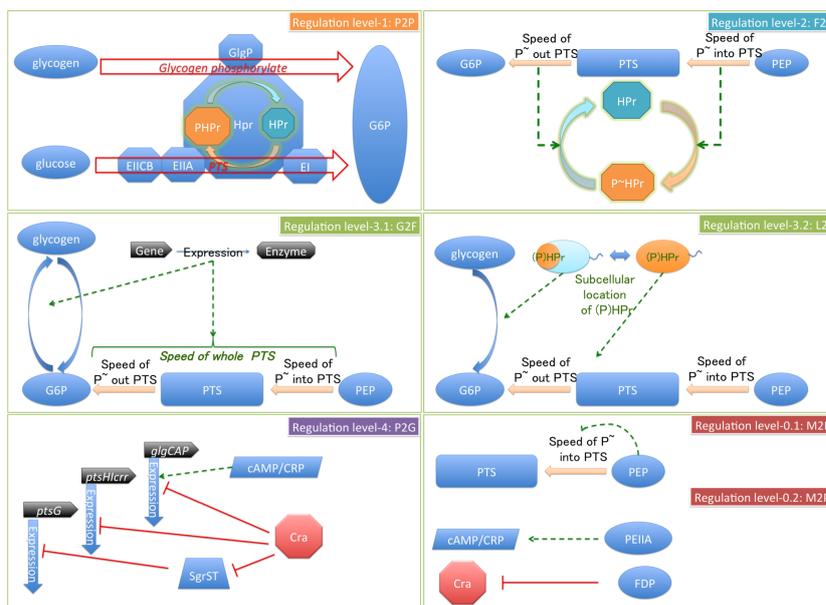
Fig. 1. Regulation mechanisms controlling glucose and glycogen utilization. **Level-1**: HPr controls glucose phosphorylation (as a PTS enzyme) and glycogen decomposition (in complex with GlgP). **Level-2**: Phosphorylation of HPr is controlled by the balance between phosphate groups influx into and outflux from the PTS. **Level-3.1**: Gene expression regulates PTS the speed of transportation of phosphate and the process of glycogen metabolism. **Level-3.2**: In HPr function, either for PTS or for glycogen degradation, depends on its subcellular location. **Level-4**: PTS and *glg* genes expression are controlled by cAMP/CRP and Cra. **Level-0.1**: PEP concentration controls the amount of phosphate entering the PTS. **Level-0.2**: Cra expression is controlled by FDP amount, and cAMP/CRP is enhanced by PEIIA^{Glc}.
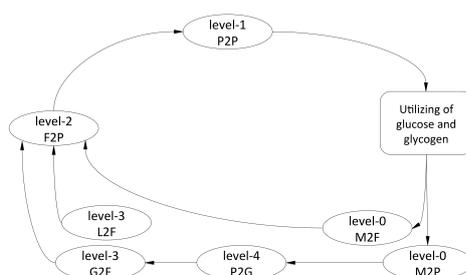
Fig. 2. Regulatory circuit system for glucose and glycogen utilization. **M2F** is a regulation from metabolite to molecule flux speed; **M2P** is a regulation from metabolite to protein; **P2G** is regulation from protein to gene expression; **G2F** is a regulation from gene expression to molecule flux speed; **L2F** is a regulation from molecule subcellular location to molecule flux speed; **F2P** is a regulation from molecule flux speed to protein; **P2P** is a regulation from protein to protein.

Level-3.1 G2F (Regulation from gene expression level to protein). PTS enzymes for glucose uptake in *E. coli* include EI, HPr, EIIA^{Glc} and EIICB^{Glc}, the former three enzymes are expressed from *ptsHIcrr* gene cluster and EIICB^{Glc} is from *ptsG*. After an exponential increasing, when an enzyme concentration increases above a certain threshold, its catalyzed reaction speed will remain in a high level [23]. Here we assumed that, when PTS enzymes are expressed above a certain threshold, the whole PTS reaction speed would be extremely accelerated.

Level-3.2 L2F (Regulation from molecule subcellular location to molecule flux speed): When (P)HPr is located at the cell's poles, it mainly functions for glycogen phosphorylation. And when (P)HPr is scattered in cytosol, it serves for the function of PTS, which is responsible for glucose uptake. The deduction of subcellular location of (P)HPr controlling system will be explained in **Subsection 2.2**.

Level-4 P2G (Regulation from protein to gene expression). Cra is a global regulator of the genes for carbon metabolism in *E. coli* [6], which directly regulates *glgC* and *glgA* and *ptsHIcrr* operon, or indirectly influences *ptsG* transcription via SgrST pathway [6, 24]. The function of upregulation of *glgC* and *glgA* by cAMP/CRP complex is confirmed by experiments of [17]. Comprehensively say, when Cra levels decreases, it releases the inhibition of *glgC* and *glgA*; as a consequence cAMP/CRP activates extremely strong expression of *glgC* and *glgA*.

Level-0.1 M2F (Regulation from metabolite to molecule flux speed). High enough PEP levels activate the phosphate influx into PTS by stimulating EI dimerization [18, 19]. This reaction EI+EI⇒EIEI has been thought to be the limiting speed of PTS.

Level-0.2 M2P (Regulation from metabolite to protein): When fructose-1,6-bisphosphate (FDP) reaches a high level, Cra expression is repressed, which releases its inhibition of *glgC* and *glgA* [6]. High concentration PEIIA^{Glc} leads to the accumulation of cAMP [8].

2.2   HPr role in glycogenolysis or PTS depends on its subcellular localization

Lopian et al. (2010) described the spatial and temporal organization of PTS enzymes in *E. coli*, especially HPr and EI [20]. According to their study, HPr and EI mainly stay in the poles of a cell independently, and if HPr is released to the cytosol, it should be phosphorylated by PEI in the presence of glucose. Genobase also shows a photo of HPr scattering in the cytosol [25].

In the glycogen metabolism, interestingly, glycogenesis enzymes (GlgC, GlgA) and glycogen granules locate at the poles, while GlgP is scattered in the cytosol [2]. GlgP is considered always bound in a complex with HPr, since the concentration of HPr is much higher than that of GlgP in *E. coli* [14, 15].

Based on these studies, we hypothesize that HPr controls the priority in glucose and glycogen utilization in *E. coli*: (1) If there is no glucose, HPr cannot get phosphate from EI, keeping its location at the poles. Hence, this pole-located

HPr mainly serves for glycogen decomposition, whose speed is regulated by phosphorylation state of (P)HPr:GlgP as described in **Subsection 2.1**. (2) If there is a little glucose supply, at the very beginning of lag phase, glucose uptake takes place at poles areas for a very short time until all the phosphates are removed from these PTS enzymes including HPr (See ***Early lag phase (1)*** in **Subsection 4.1**). Note that the pole-located HPr also has the ability of exchanging phosphate with other PTS enzymes. (3) If glucose is abundant, HPr gets phosphate from PEI, causing its release to the cytosol. Cytosol-scattered HPr works as a PTS protein, but not for glycogenolysis, hence, transporting phosphate from EI to EIIA^{Glc}.

3 Construction of a dynamic simulation model of central metabolic pathway with HFPN

Central metabolic pathway in *E. coli* is constituted by the glycolysis, the PP pathway, and the tricarboxylic acid cycle (TCA cycle). Most glycolysis models are based on ordinary differential equation (ODE) [12, 26, 27]. Chassagnole et al. (2002) constructed an integrated ODE model of glycolysis and PP pathways [12], which is often used as a base model in many studies [26–29]. By assembling TCA cycle with the model of [12], Kadir et al. (2010) set up an ODE model together with six pieces of logical controlling rules [27], and Usuda et al. (2010) included gene regulation in [26]. Kinetic parameters of these ODE model has been stored in many databases, such as BRENDA [30], SABIO-RK [31], and BioModels [32], and a number of works focused on parameter optimization [33, 34]. PTS are usually represented by one or a few equations in these ODE models. Rohwer at el (2000) set a mass balance theory model of PTS, by using experimentally tested mass action constant for each elementary biochemical reaction within PTS enzymes [13], and some studies are based on it [9, 35].

The simulation of our HFPN models are conducted on Cell Illustrator 4.0 [36]. Before realizing a whole model, we have first set up two independent HFPN models based on these published, ODE models of glycolysis and PP pathway [12, 32] and mass balance theory models of PTS [13, 35]. Subsequently, these two HFPN models are combined into one.

This HFPN model was further extended by incorporating glycogen metabolism pathway and basic regulatory mechanisms, and finally we got an extended HFPN model of carbohydrate metabolism, as shown in Fig. 3. We employed general mass action method to construct this integrated HFPN model, in which mass action constants were manually fitted so as to meet biological data of glycogen and other metabolites concentrations from our former study [1]. From this URL[0], a complete HFPN model, lists of places, transitions, and arcs can be obtained.

---

[0] `http://ds0n.cc.yamaguchi-u.ac.jp/~mzemi/etchp/ecoli_doc/MatsunoLab.htm`

Fig. 3. HFPN model of extent center metabolism pathway in *E. coli*. This model includes 3 major parts: glycolysis and pentose phosphate pathway part is in **bright yellow area**, PTS part is in **bright green area** and glycogen metabolism pathway part is **bright blue area**. **Green filed transitions** are degradation or dilution processes of their connected metabolites or enzymes. **Black border components** are of the 3 major parts. **Red border components** are of regulatory mechanisms, in which transition *ptsG* and *ptsHIcrr* represent PTS genes expression process; **G_cAMP** is the process of PEIIA^{Glc} activating cAMP production; **Dimer** is the process of EI dimerization; ***k*** is the parameter controlling the whole PTS reaction speed; **Location** is of the molecular subcellular localization regulation mechanism; **(P)HPr::GlgP** represent the process of the binding of (P)HPr::GlgP catalyzing glycogen decomposition. This is the snapshot of main part of our HFPN model, other components can be found in Suppl. documents in the URL[0]

The integrated HFPN model produced the correct behavior of metabolite concentrations of G6P, PEP, FDP etc. in a batch culture as well as the concentrations of glycogen and extracellular glucose in Fig. 4, which can be confirmed by comparing with their experimental data in Supplementary data of [1]. Further, PTS enzymes level are also illustrated in this figure, and the noisy behaviors in EIEI and PEI come from the rapid alternation of phosphate between these molecules. Although more evaluations are required, we can say that our simulation results express similar behavior to the experimental data.

4   Confirmation of the role of HPr and EIIA^{Glc} as key regulators by simulation

4.1   Biological analyses based on the simulation results

With running simulations on the constructed HFPN model, we are able to systematically understand the process of carbohydrate metabolism in a batch cul-
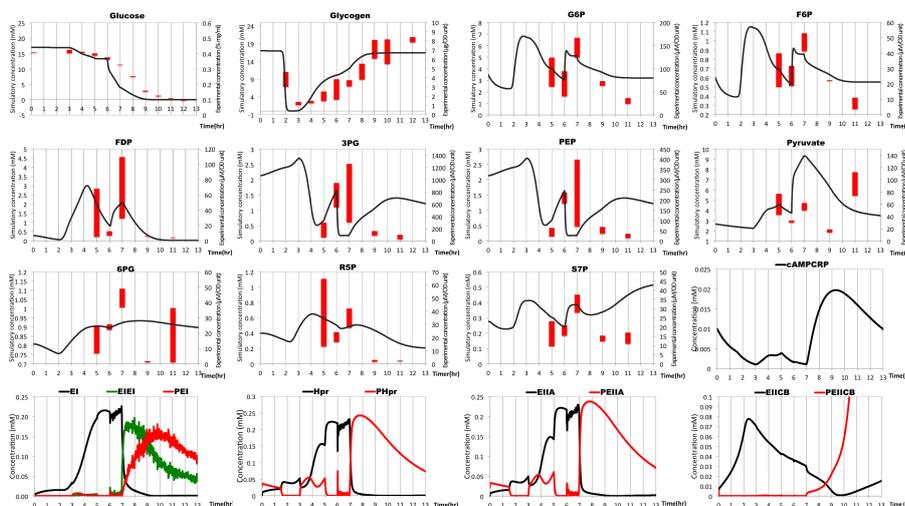
Fig. 4. Calculation results of HFPN model of extended central metabolism pathway in *E. coli*. Solid-curve is simulation result of this work. **Red-bar** denotes experimental data of our previous work [1].

ture in *E. coli* along its lifetime, which consists of 5 phases, early lag phase, late lag phase, early log phase, late log phase, and stationary phase. Simulated concentrations of glucose, glycogen, FDP, HPr (EIIA^{Glc}), PHPr (PEIIA^{Glc}), and cAMP are shown in the left panel of Fig. 5.

Early lag phase (1). At the beginning of this phase, *E. coli* begins its growth just after being put into a fresh medium. At this point, (P)HPr is mainly present at the poles and causes a little glucose uptake locally. Glycogen is not utilized well in this phase, because it is surrounded by PHPr. Indeed the higher affinity of PHPr than HPr isolates GlgP from glycogen, resulting in a very slow speed decomposition rate of glycogen.

Early lag phase (2). Although this phase begins with PHPr, this protein slowly loses its phosphate. Because glycolytic pathway is not working in this phase, so phosphate cannot be provided through PTS. As HPr dephosphorylation completes, glycogen catalysis by HPr::GlgP begins, and *E. coli* uses glycogen as its main carbon source. Along with the quick consumption of glycogen, HPr is moved to the cytosol by the function of PEI [20]. Meanwhile, glycogen supplied phosphate flows into the central metabolic pathway, causing PEP accumulation. Distribution of (P)HPr in the cytosol will be finished at almost the same time.

Late lag phase. This is a period of slow glucose uptake, which is caused by a relevant lower level of PEP, due to a low speed EI dimerization [18]. This means that metabolites produced from glycogen support the transportation of phosphate for glucose uptake. During this period, (P)HPr has been distributed in the cytosol, whose major role is to work for PTS not for glycogen, and this
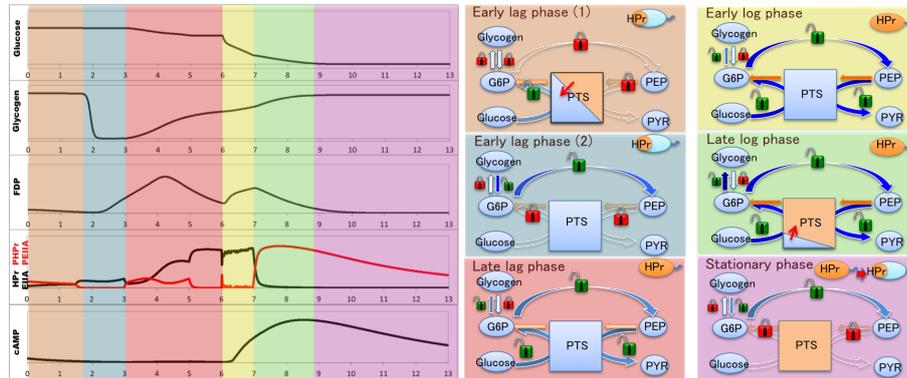
Fig. 5. Systematic understanding of the phases of extended central metabolism in an *E. coli* along its whole lifetime based on this works HFPN model. In the figure, **left panel** illustrates experiment and simulation behaviors of some major metabolites and enzymes, **right panel** shows sugar and phosphate flux processes of *E. coli* utilizing glucose and glycogen, and same colored area of left panel and right panel are of the same phase. In right panel: **Blue colored PTS** represents unphosphorylated PTS, **orange colored PTS** is phosphorylated PTS. **Blue filled arrows** indicate carbon flux routes, in which deeper blue color represents more flowing amount; **orange color filled arrows** indicate phosphate flux routes, in which deeper orange color represents more flowing amount. **Closed red locks** means inactivated pathway; **open green locks** means activated pathway. **The whole orange colored *E. coli* marked with "HPr"** indicates HPr is scattered in cytosol; **only head area orange colored *E. coli* marked with "HPr"** indicates HPr is at poles.

also causes the start of glycogen accumulation. Meanwhile in this phase more PTS enzymes are expressed, preparing for the impending log phase.

Early log phase. Uptake of glucose is very fast in this phase due to the highly expressed PTS proteins and the active transportation of phosphate by these PTS proteins. Glucose is the main sugar source in this phase.

Late log phase. In this phase, under the combined regulation of PEIIA^{Glc} (via cAMP/CRP), and FDP (via Cra), *glgC* and *glgA* are expressed at extremely high levels [2, 6, 8], causing efficient glycogen accumulation. Due to the lower speed of phosphate output from the PTS comparing with its input speed from PEP, high level of PHPr are working for glucose uptake. (P)HPr is mainly expressed in the cytosol, so it can hardly contribute to glycogen decomposition.

Stationary phase. When cells come to a stationary phase, glycogen is in its slow speed catalyzing state. Since (P)HPr is maintained in phosphorylated state, it concentrates towards the poles, where glycogen is located. In the post stationary phase, there is no glucose supplied outside, glycogen is used as a carbon source for cells to survive. Glycogen low speed catalyzation is regulated by surrounding PHPr in poles. Next, if the *E. coli* is put into another culture, a new lag phase begins.

Table 1. Multi-valued formulation of the regulation in utilizing glycogen and glucose in *E. coli*. α represents α (*glgC* & *glgA* activation), β represents β (*glgC* & *glgA* activation), γ represents γ (glycogen composition). The meaning of multi-values are 0 (no/off), 1 (on/slow) , and 2 (fast).

α *glgC* & *glgA* activation	β *glgC* & *glgA* activation	γ glycogen composition
0	0	0
1	0	1
0	1	1
1	1	2

4.2 Logical expressions of regulator states throughout the phases

Multi-valued logic rule. *glgC* and *glgA* are the genes that forms an operon with *glgP* [2, 17, 37]. According to experimental result, *glgC* and *glgA* are regulated by cAMP [2, 17] and FDP [6], respectively. Hence we can consider that the transcription of this two genes is regulated by the combination of FDP amount and cAMP level, which are distinguished α (*glgC* & *glgA* activation) and β (*glgC* & *glgA* activation), respectively. Actually, from the biological literature [2, 6, 17], it is known that the composition speed of glycogen varies depending on the expression pattern of α and β. If either α or β is expressed, glycogen is composed in slow speed, but if both α and β are expressed, glycogen is composed in high speed. This function can be expressed by multi-valued logic as presented in Table 1.

Phase transitions based on the regulatory factors. According to aforementioned analysis, the importance of HPr and EIIA^{Glc} on glycogen regulation is pointed out from a biological point of view. To make it more precise, we will express this regulatory system from an engineering point of view, presenting logical representation of this system as shown in Table 2. Glycogen process is controlled by the regulators FDP, EIIA^{Glc}, and HPr in the left column of this table. Among them, FDP and EIIA^{Glc} are involved in glycogen synthesis, and HPr in its decomposition. In the following, we will show, phase by phase, how composition and decomposition take place on the controls with these regulators in this table.

Early lag phase. Because of "very slow" uptake speed of glucose, FDP amount is in "low" level, resulting in "off" expression of *glgC* & *glgA* genes (α). EIIA^{Glc} and HPr display the same behavior, changing these phosphorylation states, "yes → no". In addition, *glgC* & *glgA* activation (β) is influenced by this state transition as "on → off" in Table 2. Glycogen composition, however, is not influenced by these regulations, because the uptake speed of glucose is too slow to produce glycogen. On the other hand, glycogen decomposition takes place in this phase, with changing its speed "slow → fast" according to the phosphorylation

Table 2. Behaviors of key regulators (HPr and EIIA^{Glc}) adjusting glucose and glycogen utilization in an *E. coli*. In this table, the five proliferation phases (e.g. Late lag phase) are corresponding with their processes of experiment and simulation data in Fig. 5 .

Regulator	Lag phase		Log phase		Stationary
	Early	Late	Early	Late	phase
speed of glucose uptake	very slow	slow	very fast	fast	no
FDP amount	low	high	low→high	high→low	no
α(**glgC&glgA activation**)	off	on	off→on	on→off	off
EIIA^{Glc} phosphorylation	yes→no	no	no	yes	yes
(**regulated cAMP level**)	high→low	low	low	high	high
β(**glgC&glgA activation**)	on→off	off	off	on	on
glycogen γ(**composition**)	no	slow	slow	fast→slow	no
(**decomposition**)	slow→fast	no	no	no	slow
HPr (**phosphorylation**)	yes→no	no	no	yes	yes
(**localization**)	pole	cytosol	cytosol	cytosol	pole

state transition of HPr "yes → no". Hence, glycogen is the major sugar source in this phase.

Late lag phase. Since *E. coli* have not consumed much energy yet in this phase, FDP accumulates in "high" levels despite the "slow" glucose uptake speed. Hence *glgC* & *glgA* (α) is "on". On the contrary, *glgC* & *glgA* (β) is "off", resulted from "no" phosphorylation state of EIIA^{Glc} via "low" cAMP level. According to the rule (if α=1 and β=0 then γ=1) in Table 1, glycogen is composed (γ) in "slow" speed. On the other hand, glycogen decomposition does not take place in this phase, because HPr is not located at the poles but distributed in the cytosol, which does not satisfy the requirement for glycogen decomposition.

Early log phase. Due to "very fast" speed of glucose uptake, FDP is accumulated in *E. coli*, despite its high metabolic activity, changing its amount as "low → high". Accordingly, the state of *glgC* & *glgA* (α) activation is changed as "off → on". In this stage, HPr is not phosphorylated, then the expression of *glgC* & *glgA*(β) is "off"; consequently the composition speed of glycogen (γ) is "slow", though it temporally drops to "no" level. On the other hand, "no" decomposition of glucose takes place in this phase from the same reason as late lag phase above.

Late lag phase. Because much glucose was consumed in the previous phase, its uptake speed is going to be slow down. Accordingly, for the phosphate flow in PTS, the input speed of phosphate from PEP becomes faster than the output speed to G6P, causing EIIA^{Glc} phosphorylation "yes" and cAMP level "high". As a result, *glgC* & *glgA* activation (β) turns "on". In addition, because, in the early half of this phase, FDP is in a high level, *glgC* & *glgA* activation (α) is also turned "on". Hence, both α and β regulations are working. In this case, according to Table 1, glycogen composition (γ) should be marked at "fast"

speed. Accompanying with decreasing glucose amount, FDP concentration drops later in this phase, that is "high → low", resuling in *glgC* & *glgA* activation (α) as "on → off". As a result, in the later part of this phase, the speed of glycogen composition (γ) changes as "fast → slow". On the other hand, in this phase, HPr is still in cytosol working for PTS, not for glycogenolysis. In all, since "fast" composition and "no" decomposition are conducted, glycogen accumulates quickly in this period.

Stationary phase. In this period, because extracellular glucose has been totally consumed off, the speed of glycogen is marked as "no" despite the "on" state of *glgC* & *glgA* activation (β). Hence there is "no" glycogen composition (γ). Because of the inactive PTS and the high amount glycogen, (P)HPr is concentrated at the "poles", decomposing glycogen (γ) in a "slow speed" for long survival of cells.

5   Conclusion

Some works focus on modeling glycolysis, pentose phosphate pathway, TCA cycle etc. [12, 26, 27], and some focus on the calculation of PTS performance with a protein mass balance theory method [13, 35]. And also some of them set up ODE models by combining PTS into their glycolysis pathways [26, 27]. But none of them take the glycogen metabolic pathway into account. In this work we firstly integrated general mass action based glycogen metabolism model and mass balance theory based PTS model into a computational model with HFPN.

By applying this model, basic regulators for *E. coli* to utilize extracellular glucose and intracellular glycogen were identified. That is, (P)HPr not only works as a member of PTS enzymes but also functions to realize different catalyzing speeds of glycogen by its phosphorylation state combined with GlgP. Actually, phosphorylation state of (P)HPr is controlled by the phosphate flux speed influx and outflux of PTS, and this flux speed is controlled by gene expression, sub-cellular localization, and metabolite concentration (glucose, PEP, FDP). HPr and $EIIA^{Glc}$ are considered to be key roles among these regulators during the utilization of glycogen and glucose by *E. coli*.

Based on the model with regulatory systems in this work, we provided a systematic view of glucose and glycogen utilization by *E. coli*. This confirms our previous conclusion that glycogen plays an important role as a primary carbon source in lag phase [1].

In our model, the behavior after log phase does not correspond well to experimental data. The reasons might be inconsistencies in the referenced ODE and PTS that were modeled so as to function in a short time course (50 s) or steady stat context, and the difficulty in controlling its flux speed dynamically in an hour time scale. One of our future tasks is to address this limitation.

Acknowledgement

References

1. Yamamotoya, T., Dose, H., Tian, Z., Faur, A., Toya, Y., Honma, M., Igarashi, K., Nakahigashi, K., Soga, T., Mori, H., Matsuno, H.: Glycogen is the primary source of glucose during the lag phase of E. coli proliferation. Biochim. Biophys. Acta. (BBA) - Proteins and Proteomics. **1824** (2012) 1442–1448
2. Wilson, W.A., Roach, P.J., Montero, M., Baroja-Fernndez, E., Muoz, F.J., Eydallin, G., Viale, A.M., Pozueta-Romero, J.: Regulation of glycogen metabolism in yeast and bacteria. FEMS Microbiol. Rev. (2010)
3. Kotrba, P., Inui, M., Yukawa, H.: Bacterial phosphotransferase system (PTS) in carbohydrate uptake and control of carbon metabolism. J. Biosci. Bioeng. **92** (2001) 502–517
4. Edwards, A.N., Patterson-Fortin, L.M., Vakulskas, C.A., Mercante, J.W., Potrykus, K., Vinella, D., Camacho, M.I., Fields, J.A., Thompson, S.A., Georgellis, D., Cashel, M., Babitzke, P., Romeo, T.: Circuitry linking the Csr and stringent response global regulatory systems. Mol. Microbiol. **80** (2011) 1561–1580
5. Baker, C.S., Morozov, I., Suzuki, K., Romeo, T., Babitzke, P.: CsrA regulates glycogen biosynthesis by preventing translation of glgC in Escherichia coli. Mol. Microbiol. **44** (2002) 1599–1610
6. Shimada, T., Yamamoto, K., Ishihama, A.: Novel Members of the Cra Regulon Involved in Carbon Metabolism in Escherichia coli. J. Bacteriol. **193** (2011) 649–659
7. Vinuselvi, P., Kim, M.K., Lee, S.K., Ghim, C.-M.: Rewiring carbon catabolite repression for microbial cell factory. BMB Rep. **45** (2012) 59–70
8. 8Deutscher, J., Francke, C., Postma, P.W.: How Phosphotransferase System-Related Protein Phosphorylation Regulates Carbohydrate Metabolism in Bacteria. Microbiol. Mol. Biol. Rev. **70** (2006) 939–1031
9. Francke, C., Postma, P.W., Westerhoff, H.V., Blom, J.G., Peletier, M.A.: Why the Phosphotransferase System of Escherichia coli Escapes Diffusion Limitation. Biophys. J. **85** (2003) 612–622
10. Choi, Y.-L., Kawamukai, M., Utsumi, R., Sakai, H., Komano, T.: Molecular cloning and sequencing of the glycogen phosphorylase gene from Escherichia coli. FEBS Lett. **243** (1989) 193–198
11. Nagasaki, M., Yamaguchi, R., Yoshida, R., Imoto, S., Doi, A., Tamada, Y., Matsuno, H., Miyano, S., Higuchi, T.: Genomic data assimilation for estimating hybrid functional Petri net from time-course gene expression data. Genome Inform. Ser. **17** (2006) 46
12. 12Chassagnole, C., Noisommit-Rizzi, N., Schmid, J.W., Mauch, K., Reuss, M.: Dynamic modeling of the central carbon metabolism of Escherichia coli. Biotechnol. Bioeng. **79** (2002) 53–73
13. Rohwer, J.M., Meadow, N.D., Roseman, S., Westerhoff, H.V., Postma, P.W.: Understanding Glucose Transport by the Bacterial Phosphoenolpyruvate:Glycose Phosphotransferase System on the Basis of Kinetic Measurements in Vitro. J. Biol. Chem. **275** (2000) 34909–34921

14. Seok, Y.-J., Sondej, M., Badawi, P., Lewis, M.S., Briggs, M.C., Jaffe, H., Peterkofsky, A.: High Affinity Binding and Allosteric Regulation ofEscherichia coli Glycogen Phosphorylase by the Histidine Phosphocarrier Protein, HPr. J. Biol. Chem. **272** (1997) 26511–26521

15. Koo, B.M., Seok, Y.J.: Regulation of glycogen concentration by the histidine-containing phosphocarrier protein HPr in Escherichia coli. J. Microbiol. **39** (2001) 24–30

16. Seok, Y.J., Koo, B.M., Sondej, M., Peterkofsky, A.: Regulation of E. coli glycogen phosphorylase activity by HPr. J. Mol. Microbiol. Biotechnol. **3** (2001) 385–394

17. Romeo, T., Preiss, J.: Genetic regulation of glycogen biosynthesis in Escherichia coli: in vitro effects of cyclic AMP and guanosine 5-diphosphate 3-diphosphate and analysis of in vivo transcripts. J. Bacteriol. **171** (1989) 2773–2782

18. Gabor, E., Ghler, A.-K., Kosfeld, A., Staab, A., Kremling, A., Jahreis, K.: The phosphoenolpyruvate-dependent glucose-phosphotransferase system from Escherichia coli K-12 as the center of a network regulating carbohydrate flux in the cell. Eur. J. Cell Biol. **90** (2011) 711–720

19. Patel, H.V., Vyas, K.A., Savtchenko, R., Roseman, S.: The Monomer/Dimer Transition of Enzyme I of the Escherichia coli Phosphotransferase System. J. Biol. Chem. **281** (2006) 17570–17578

20. Lopian, L., Elisha, Y., Nussbaum-Shochat, A., Amster-Choder, O.: Spatial and temporal organization of the E. coli PTS components. The EMBO Journal. **29** (2010) 3630–3645

21. Lengeler, J.W., Jahreis, K.: Bacterial PEP-dependent carbohydrate: phosphotransferase systems couple sensing and global control mechanisms. Contrib. Microbiol. **16** (2009) 65–87

22. Karelina, T.A., Ma, H., Goryanin, I., Demin, O.V.: EI of the Phosphotransferase System of Escherichia coli: Mathematical Modeling Approach to Analysis of Its Kinetic Properties. J. Biophys. **2011** (2011) 1–17

23. Vieira, A.P. de A., Da Silva, M.A.P., Langone, M.A.P.: Biodiesel production via esterification reactions catalyzed by lipase. Latin American applied research. **36** (2006) 283–288

24. Traxler, M.F., Summers, S.M., Nguyen, H.-T., Zacharia, V.M., Hightower, G.A., Smith, J.T., Conway, T.: The global, ppGpp-mediated stringent response to amino acid starvation in Escherichia coli. Mol. Microbiol. **68** (2008) 1128–1148

25. GenoBase: Results of protein localization references, `http://ecoli.naist.jp/GB8-dev/GFP/gfp_result.jsp?fword=JW3643`

26. Usuda, Y., Nishio, Y., Iwatani, S., Van Dien, S.J., Imaizumi, A., Shimbo, K., Kageyama, N., Iwahata, D., Miyano, H., Matsui, K.: Dynamic modeling of Escherichia coli metabolic and regulatory systems for amino-acid production. J. Biotechnol. **147** (2010) 17–30

27. Kadir, T., Mannan, A., Kierzek, A., McFadden, J., Shimizu, K.: Modeling and simulation of the main metabolism in Escherichia coli and its several single-gene knockout mutants with experimental verification. Microb. Cell Fact. **9** (2010) 88

28. Baldazzi, V., Ropers, D., Markowicz, Y., Kahn, D., Geiselmann, J., De Jong, H.: The carbon assimilation network in Escherichia coli is densely connected and largely sign-determined by directions of metabolic fluxes. PLoS Comput. Biol. **6.6** (2010) e1000812

29. Tohsato, Y., Ikuta, K., Shionoya, A., Mazaki, Y., Ito, M.: Parameter optimization and sensitivity analysis for large kinetic models using a real-coded genetic algorithm. Gene. **518** (2013) 84–90

30. Schomburg, I., Chang, A., Ebeling, C., Gremse, M., Heldt, C., Huhn, G., Schomburg, D.: BRENDA, the enzyme database: updates and major new developments. Nucl. Acids Res. **32** (2004) D431–D433
31. Wittig, U., Kania, R., Golebiewski, M., Rey, M., Shi, L., Jong, L., Algaa, E., Weidemann, A., Sauer-Danzwith, H., Mir, S., Krebs, O., Bittkowski, M., Wetsch, E., Rojas, I., Muller, W.: SABIO-RK–database for biochemical reaction kinetics. Nucleic Acids Res. **40** (2011) D790–D796
32. Li, C., Donizelli, M., Rodriguez, N., Dharuri, H., Endler, L., Chelliah, V., Li, L., He, E., Henry, A., Stefan, M.I., Snoep, J.L., Hucka, M., Novre, N.L., Laibe, C.: BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. BMC Syst. Biol. **4** (2010) 92
33. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional (1989)
34. Ono, I and Kobayashi, S: A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover, Proceedings of the Seventh International Conference on Genetic Algorithms. (1997) 246-253.
35. Rodríguez, J.V., Kaandorp, J.A., Dobrzyński, M., Blom, J.G.: Spatial stochastic modelling of the phosphoenolpyruvate-dependent phosphotransferase (PTS) pathway in Escherichia coli. Bioinformatics. **22** (2006) 1895–1901
36. Nagasaki, M., Saito, A., Jeong, E., Li, C., Kojima, K., Ikeda, E., Miyano, S.: Cell Illustrator 4.0: A Computational Platform for Systems Biology. In Silico Biology. **10** (2010) 5–26
37. Montero, M., Almagro, G., Eydallin, G., Viale, A.M., Muoz, F.J., Bahaji, A., Li, J., Rahimpour, M., Baroja-Fernndez, E., Pozueta-Romero, J.: Escherichia coli glycogen genes are organized in a single glgBXCAP transcriptional unit possessing an alternative suboperonic promoter within glgC that directs glgAP expression. Biochem. J. **433** (2011) 107–117

Author Index