Building Executable Biological Pathway Models Automatically from BioPAX

Timo Willemsen, Anton Feenstra, and Paul Groth

Department of Computer Science, VU University Amsterdam, The Netherlands

timo.willemsen@gmail.com,{k.a.feenstra,p.t.groth}@vu.nl

Abstract. The amount of biological data exposed in semantic formats is steadily increasing. In particular, pathway information (a model of how molecules interact within a cell) from databases such as KEGG and WikiPathways are available in a standard RDF-based format BioPAX. However, these models are *descriptive* and not *executable* in nature. Being able to simulate or execute a pathway is one key mechanism for understanding the operation of a cell. The creation of executable models can take a significant amount of time and only relatively few such models currently exist. In this paper, we leverage the availability of semantically represented pathways, to bootstrap the creation of executable pathway models. We present an approach to automate the creation of executable models in the form of Petri-Nets from BioPAX represented pathways. This approach is encapsulated in an online tool, BioPax2PNML.

Keywords: biological pathways, biological networks, BioPax, executable models, Petri nets

1 Introduction

A biological pathway, simply said, is a sequence of interactions among molecules of a cell. There are many different types of pathways; gene regulation pathways, signaling pathways and protein interaction pathways are among the most commonly used ones. [1]

Originally, pathways were hand-drawn and presented in papers. Pathways are now made available in online databases in computer parsable formats (e.g. BioPAX). For example, the WikiPathways has over 1700 available pathways[1]. While these pathway descriptions are highly useful, they contain mostly static information about interacting molecules and do not describe how *pathways actually work* or give insight into the dynamics of these interactions [2].

To address this lack of information, work has been undertaken to create computational models of these pathways [3]. Two types of models can be distinguished: executable and mathematical [4]. The mathematical models give insight into quantities and how they change over time, and are frequently created by systems biologists. Executable models are valuable to biologists because they have

[1] See http://WikiPathways.org/index.php/WikiPathways:Statistics for statistics on WikiPathways

a large variety of uses [4,5]. They can be used to summarize available knowledge of interactions and mechanisms in a system, and to investigate how components cooperate to produce global system behaviour. Creating an executable model is still a tedious manual process, mostly because they contain parameters that need to be collected manually. On the other hand, mathematical models typically require detailed knowledge of (kinetic and rate) parameters, which are often not available and can be very hard to obtain from experiments. From our experience, for executable models, the process of model construction and parameter calibration usually takes several months [3,6,7], even for a modestly sized network. This is currently one of the major bottlenecks in computational life sciences research [8].

This paper begins to address this bottleneck by leveraging the availability of semantic representations of pathways and converting them to an executable model. Concretely, the contributions of this paper are: *i)* to present a method to automate validation of pathway data; *ii)* a mapping of the BioPAX format to an executable model (Petri nets, represented in the Petri Net Markup Language; PNML); and, *iii)* a method to automatically create these executable models. We have developed a webservice that encapsulates the described method and can be accessed at `www.few.vu.nl/~twn370/BioPax2PNML/`. Additionally, all code is available online at: `https://github.com/TimoWillemsen/Biopax2PNML`.

The rest of this paper is organized as follows. We begin in Section 2 with background information on biological pathways and common formats for both descriptive (BioPAX) and executable (PNML) representations of them. We then describe our approach for mapping between these two formats (Section 3). To ensure that a BioPAX pathway has the appropriate information to be converted to PNML, we present a validation approach in Section 4. This is followed, in Section 5, by a description of the implementation of our method. Finally, we conclude with some thoughts on future work in Section 6.

2 Biological Pathways

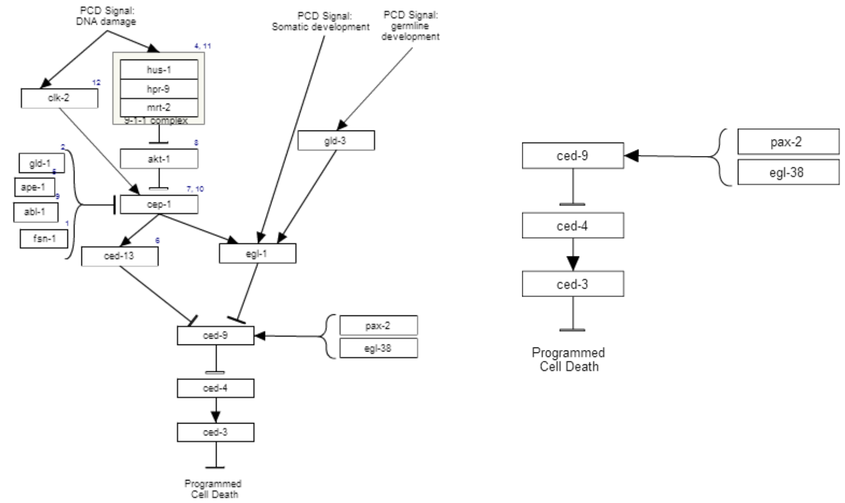
There are different types of biological pathways, corresponding to different levels of abstraction. For example, a pathway may describe interactions between different cells, or between genes, or between proteins, or it may describe biochemical reactions (or combinations thereof). Many databases exist that collect this information in a variety of forms, and some are very specialized on particular types of data. It is beyond the scope of this work to provide a comprehensive overiew. Some of the most well-known are WikiPathways [9], focused on signal transduction; the KEGG Pathway database [10,11], with a focus on metabolic pathways; and Reactome [12] which has a broader scope.

The examples provided in this paper will focus on signal transduction pathways, as these tend to be well-studied and therefore well-defined. Such pathways typically include protein-protein interactions, protein-gene interactions and biochemical reactions.

We have based our research on the pathways provided by the WikiPathways database [9]. This is a community-driven service where biological pathways are extensively manually curated. The context of the pathways included in WikiPathways can vary considerably, depending on their intended use. For example, simply representing known interactions in a shareable way is considered useful, but such pathways likely will not include details that are crucial for computational analysis, even as simple as explicit notation of interactions among proteins and genes. As a result of this, only certain pathways are suitable for computational analysis.

One such example is the *C. elegans* Programmed Cell Death pathway from the WikiPathways Database, as shown in Fig. 1.

Fig. 1. *C. elegans* Programmed Cell Death Pathway from the WikiPathways Database ID:WP367. The left panel shows the complete pathway, the right panel shows the subset of 5-genes used.



For the purpose of this paper, we have taken a subset of this pathway, as shown in Fig. 1. This pathway consists of 5 genes. When ced-3 is activated, it will trigger the cell's programmed death.

We now discuss the computational representation of pathways used by WikiPathways. After which, we briefly describe the use of Petri-nets to as a language for executable models of pathways.

2.1 BioPax

In 2010 Demir et al [13] created the Web Ontology Language (OWL) based standard for modeling pathways: BioPax. A key aspect of this standard is that it allows for referring to external databases for information (e.g. linking to UniProt protein descriptions.) This standard has been used in many different biological databases; all the three mentioned above, Reactome, KEGG and WikiPathways expose BioPax through an RDF interface [9,11,12].

BioPax can be used to model different types of pathway components. An example of how genes are modelled in BioPax, is shown below; the ced-3 and ced-4 genes of the *C. elegans* Programmed Cell Death pathway, as shown in Fig. 1.

Two genes, ced-3 and ced-4, from the C. elegans Programmed Cell Death Pathway from the WikiPathways Database `ID:WP367`

```
<bp:Protein rdf:about="eef1e">
 <bp:displayName>ced-3</bp:displayName>
 <bp:entityReference rdf:resource="id3" />
</bp:Protein>
<bp:Protein rdf:about="c0b3e">
 <bp:displayName>ced-4</bp:displayName>
 <bp:entityReference rdf:resource="id4" />
</bp:Protein>
```

An example of interactions in a pathway modelled in BioPax is shown below; we see a reaction '`id40`' that connects a right-hand-side element (`eef1e`; ced-3) with a left-hand-side (`c0b3e`; ced-4) element.

Gene interaction of the C. elegans Programmed Cell Death Pathway from the WikiPathways Database `ID:WP367`

```
<bp:BiochemicalReaction rdf:about="id40">
 <bp:right rdf:resource="eef1e" />
 <bp:left rdf:resource="c0b3e" />
</bp:BiochemicalReaction>
```

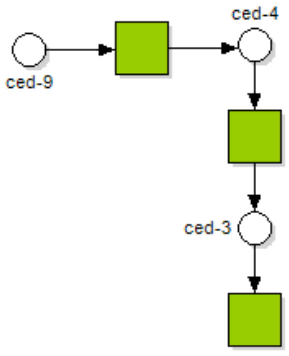
2.2 Petri nets

Petri nets are a formalism geared towards modelling and analysis of concurrent systems. A Place-Transition (PT) Petri net is a quadruple (P, T, A, m), where P is a set of places and T a set of transitions. A describes arcs which connect places with transitions or vice versa. Each place holds zero or more tokens, which represent flow of control through this place. The number of tokens in each place all together are called a marking m of the network.

Fig. 2 shows a graphical representation of such a Petri Net, again for our small example part of the *C. elegans* Programmed Cell Death pathway. Squares are transitions, representing interactions, and circles are places, representing genes. Arcs are represented by arrows, and the marking is empty. Firing of a transition depends on the availability of resources (tokens) in the input places, and represents the execution of a reaction: consuming substrates and creating products.[14,15]

For computational purposes we have chosen to represent Petri nets in the Petri Net Markup Language (PNML) format. This is a straightforward XML

Fig. 2. An example Petri net of a small part of the *C. elegans* Programmed Cell Death Pathway (WikiPathways:WP367)



standard that a number of systems support.[16] Fig. 2.2 shows the Petri net of Fig. 2 in an XML representation. Petri nets are recognized as a powerful tool to model biological pathways [14,15], as the formalism readily allows to capture both the complexity and the highly concurrent nature of biological systems, while optimally leveraging the large amounts of qualitative data available.[15,3,6]

Fig. 3. *PNML representation of the C. elegans Programmed Cell Death pathway (WikiPathways:WP367) Petri net as shown in Fig. 2.*

```
<transition id="t11">
</transition>
<place id="eef1e">
    <name>
        <text>ced-3</text>
    </name>
</place>
<place id="c0b3e">
    <name>
        <text>ced-4</text>
    </name>
</place>
<arc id="a2" source="c0b3e" target="t11" />
<arc id="a3" source="t11" target="eef1e" />
```

3 BioPax to PNML mapping

To transform static BioPax data into an executable Petri net, we have developed a mapping between the two formats. BioPax is an RDF format, while PNML is an XML format. It should be taken into account that the semantic linking is lost when a BioPax pathway is converted to PNML Petri-net. For example, genes or proteins have different identifiers in different databases. BioPax gives a way to link multiple identifiers to a gene or protein, but PNML does not support this feature.

3.1 Genes or Proteins

Each gene or protein is modelled as a place in the Petri net. Because the creation of the Pathways in WikiPathways has been done manually, often they are not consistent and may, for example, contain multiple instances of one gene or protein. The mapping does not take into consideration the fact that duplicate genes or proteins may represent the same entity and are modelled twice simply for readability, or rather that they are modelled twice because they represent a different entity of the same gene/protein (for example in a different location, or in a different state). However we address this issue with the validation rules introduced in Section 4.

The first stage in mapping is shown in Algorithm 1, which transforms BioPax proteins/genes to PNML.

Algorithm 1 Genes/Proteins BioPax to PNML

```
$P = \emptyset$
for all <bp:Protein> p in BioPax do
    if $p \notin P$ and p is other entity then
        add p to P
    end if
end for
```

3.2 Interactions

Interactions are also mapped to PNML. Each <bp:BiochemicalInteraction> is mapped to a transition. Then for each <bp:Left> an arc is added pointing into the transition and out from the corresponding place; for each <bp:Right> an arc is created pointing out of the transition and into the corresponding place. Algorithm 2 shows the straightforward way to do this.

Once both algorithms 1 and 2 are executed a Petri net is created. Formally, the Petri net can be described as $PN = P, T, A, \emptyset$ where P are the places, T the transitions, A the arcs and markings $m = \emptyset$ since there are no tokens in the system yet. In terms of modelling the biological system, the places represent biological entities, like genes, proteins or complexes, the transitions represent biochemical reactions and interactions, and the arcs represent the associations between these two. Tokens represent the availability of the resources of the corresponding place in the Petri net.

Algorithm 2 Gene/protein interaction BioPax to PNML

```
$T = \emptyset$
$A = \emptyset$
for all <bp:BiochemicalInteraction> t in BioPax do
    Add t to T
    for all <bp:Left> left in BioPax do
        left.in = t
        left.out = left.resource
        Add left to A
    end for
    for all <bp:Right> right in BioPax do
        right.in = right.resource
        right.out = t
        Add right to A
    end for
end for
```

If we then execute both Algorithm 1 and Algorithm 2 on Fig. 1, a petri net is generated. Part of the output is shown in Fig. 2.2

4 BioPax Validation

The mapping described in Section 3 is based on several assumptions about the contents of the input BioPax file. The basic assumptions are that genes, proteins and complexes (bound combinations of proteins, possibly including a gene) are entities, and that these entities can change state or identity only through biochemical interactions.

However, because of the manual nature of pathway construction, these assumptions may not hold for a given pathway instance in the database. To make sure the data is presented as it should be, we have developed a set of validation rules and a validator available online.

We have developed two types of validation rules; semantic and syntactic. The syntactic validation consists of basic RDF-validation. This is necessarily because from our preliminary survey, a large fraction of pathways are not modelled correctly for translation.

More interesting is the semantic validation. These rules ensure that the information contained in the model is consistent and complete enough to create an executable Petri net. Table 1 shows these validation rules.

These rules ensure that the provided BioPax file contains everything needed. We have categorized the validation rules by severity:

- **Category error** rules are minimal requirements for mapping.
- **Category warning** rules that mean the mapping can proceed but may lead to an unconnected or incomplete Petri net.

This framework is set up in a modular fashion, so that extension is easy.

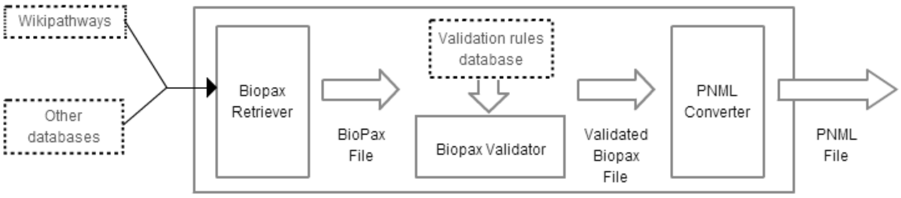
Table 1. Semantic Validation Rules

Id	Severity	Rule
1	Error	Each BioChemicalReaction should have a Left child element.
2	Error	Each BioChemicalReaction should have a Right child element.
3	Error	Each Pathway should have one or more PathwayComponents of type BiochemicalReactions.
4	Warning	Each BiochemicalReaction Left child is the actor of the interaction.
5	Warning	Each BiochemicalReaction Right child is the actant of the interaction.
6	Warning	Each unique entity of a protein/gene is modelled as a different Protein.
7	Warning	Each Protein should have a corresponding RelationshipXref.
8	Warning	Whenever a BiochemicalReaction has multiple Left or Right tags, it means that it has effect on multiple genes/proteins.
8	Warning	Protein complexes are modelled as a Complex tag.

5 Implementation

We have implemented the methods described above as a webservice. The service consists of 4 components: a validation rule database, a validator, a BioPax to PNML converter and a pathway retriever, as is shown schematically in Fig. 4.

Fig. 4. BioPax to PNML webservice architecture



5.1 Pathway retriever

The webservice provides an interface to query different datasources. At the time of writing only an interface to WikiPathways is provided, using the available webservices [17]. However, support for other generic BioPax could be a future extension.

The retriever queries WikiPathways and downloads the pathway in the BioPax format, so validation and conversion can be done.

5.2 Validation rule database

The validation rule database is a set of SPARQL queries. Each query returns a set of RDF triples that violate the rule (this set may be empty). This way feedback can be given about where the rule violation takes place in the BioPax File.

The way the database is set up allows easy addition of rules. This modularity makes it possible to improve on the current validation rules, but also allows validation rule sets for different types of pathways (for example signalling pathways vs. gene regulatory networks). Fig. 5 shows as an example the implementation of `rule 1` of Table 1.

Fig. 5. `SPARQL` *implementation of rule 1 of Table 1*

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX bp: <http://www.biopax.org/release/biopax-level3.owl#>

SELECT ?reaction
WHERE {
  ?reaction rdf:type bp:BiochemicalReaction.
    OPTIONAL {
        ?reaction bp:left ?left.
    }
    FILTER (!BOUND(?left))
}
```

This query returns every `bp:BiochemicalReaction` that does not have a `bp:left` child element associated to it.

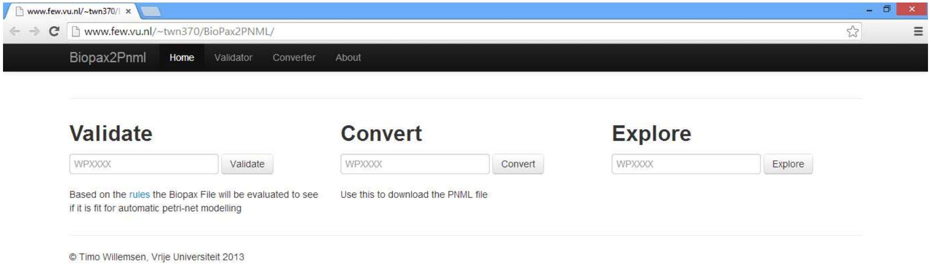
5.3 BioPax Validator

The biopax validator is software that can analyze BioPax files according to the validation rules provided by the rule database. It is essentially a graphical user interface around the SPARQL queries. It annotates the place where errors or warnings have occurred and provides an easy to use interface to solve them.

5.4 BioPax to PNML Converter

Once a BioPax file has been validated, the BioPax to PNML converter can be used to generate an executable Petri net. This converter works according to the mapping described in Section 3. This is implemented as an online tool, named BioPax2PNML, and can be accessed on www.few.vu.nl/∼twn370/BioPax2PNML/.

Fig. 6. *Screen shot of the user interface of the BioPax2PNML tool at* www.few.vu.nl/∼twn370/BioPax2PNML/.



Based on the rules the Biopax File will be evaluated to see if it is fit for automatic petri-net modelling

Use this to download the PNML file

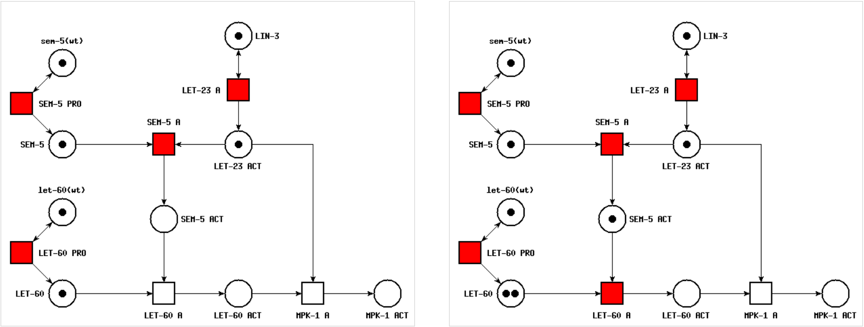
© Timo Willemsen, Vrije Universiteit 2013

5.5 Executing the PNML file

Although the proof of concept of the current work stops with the generation of a valid Petri net model in the form of a PNML file, it is nevertheless instructive to consider what subsequent steps should be. Execution of a Petri net can be performed under different execution semantics, however the most relevant for biological systems is commonly thought to be the so-called 'bounded asynchronous' execution [18,3,15]. Under this semantics, as many transitions as possible are executed simultaneously in each execution step. This represents the inherent concurrency of biological systems, where molecules typically act independently, certainly if they reside in different locations. This is also known as the 'token game', because execution of transitions has the effect of shifting tokens around the Petri net. Fig. 7 shows an example network and the change in state due to execution of enabled transitions.

Execution leads to a trajectory of markings, that represent the progression of states of the system in response to the intial marking, which corresponds to

Fig. 7. *Example of execution of a slightly non-trivial example network, taken from [15]. Enabled transitions (with input requirements satisfied; marked in red) will execute each step, execution of enabled transitions in the left panel will lead to the state shown in the right panel.*



a particular state or condition of the biological system. Typically, token levels are collected from a few places of interest and compared to experimental data of the corresponding biological molecule, or used to predict the behaviour of that particular molecule under the conditions modeled. Examples of these for signalling pathways can be found in [15,6], and for gene regulatory networks in [7].

6 Conclusion

Automatic Petri net creation of biological pathways is still a tedious process. The manual labor involved makes it so that even a modestly sized model can take several months to develop. In this paper we have provided a method to bootstrap this process. By using a mapping between the commonly used BioPax format and the PNML format, we have developed a way to automate the construction of Petri net models. Because biological information online may be inconsistent or incomplete, we have developed a set of validation rules to make sure that the data is suitable for automatic conversion.

To facilitate this, and as a proof of concept, an online tool BioPax2PNML that executes this and provide an easy interface for Petri net modelers to bootstrap the process of model creation.

The approach outlined here is an initial start to making fully developed executable models. In particular, deriving the weights on edges of the Petri nets is a challenging task. In terms of future work, we believe that by leveraging the links to other databases (e.g. Uniprot) we may be able to find additional information to infer such edge weights. Moreover, we may be able to connect additional parts of the resulting Petri-nets based on background knowledge about interactions contained in other databases or even use knowledge of chemistry provided by other data sources to create more precise models. A key foundation for work going forward is that Linked Data and Semantic Web standards facilitate the merging and acquisition of this information.

7 References

1. Ganesh A Viswanathan, Jeremy Seto, Sonali Patil, German Nudelman, Stuart C Sealfon. Getting Started in Biological Pathway Construction and Analysis *PLoS Comput Biol* **4**(2): e16, 2008
2. Pinney JW, Westhead DR, McConkey GA. Using Petri Net tools to study properties and dynamics of biological systems. *J Am Med Inform Assoc.* **12**(2):181-99, 2005.
3. Nicola Bonzanni, Elzbieta Krepska, K. Anton Feenstra, Wan Fokkink, Thilo Kielmann, Henri Bal, and Jaap Heringa. Executing multicellular differentiation: Quantitative predictive modelling of *C. elegans* vulval development. *Bioinformatics* **25**, 2049–2056, 2009.
4. Jasmin Fisher and Tom Henzinger. Executable cell biology. *Nature Biotechnology* **25**(11):1239–1249, November 2007.
5. Aviv Regev and Ehud Shapiro. Cellular abstractions: Cells as computation. *Nature* **419**:343, September 2002.
6. Bonzanni, N., Zhang, N., Oliver, S.G. and Fisher, J. The role of proteasome-mediated proteolysis in modulating activity of potentially harmful transcription factor activity in Saccharomyces cerevisiae. *Bioinformatics* **27**: i282–i287, 2011.
7. Nicola Bonzanni, Abhishek Garg, K. Anton Feenstra, Sarah Kinston, Diego Miranda-Saavedra, Judith Schutte, Jaap Heringa, Ioannis Xenarios, Berthold Göttgens. Hard-wired heterogeneity in blood stem cells revealed using a dynamic regulatory network model *Bioinformatics* in press (2013).
8. Susanna-Assunta Sansone, Philippe Rocca-Serra, Dawn Field, Eamonn Maguire, *et al.* Toward interoperable bioscience data *Nat Genet* **44**(2): 121–126, January 2012.
9. Thomas Kelder, Martijn P. van Iersel, Kristina Hanspers, Martina Kutmon, Bruce R. Conklin, Chris T. Evelo, Alexander R. Pico. WikiPathways: building research communities on biological pathways *Nucleic Acids Res* **40**(Database issue): D1301–D1307, January 2012.
10. Kanehisa M, Goto S. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res* **7**:27–30, 2000.
11. Kanehisa M, Goto S, Hattori M, Aoki-Kinoshita KF, Itoh M, Kawashima S, Katayama T, Araki M, Hirakawa M. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res* **7**(Database issue):D354–D357, 2006.
12. Robin A. Haw, David Croft, Christina K. Yung, Nelson Ndegwa, Peter D'Eustachio, Henning Hermjakob, Lincoln D. Stein. The Reactome BioMart In *Database*, Oxford, October 2011.
13. Emek Demir, Michael P. Cary, Suzanne Paley, Ken Fukuda, *et al.* BioPAX – A community standard for pathway data sharing *Nat Biotechnol* **28**: 935–942, September 2010.
14. Elzbieta Krepska, Nicola Bonzanni, K. Anton Feenstra, Wan Fokkink, Thilo Kielmann, Henri Bal, and Jaap Heringa. Design issues for qualitative modelling of biological cells with Petri nets. In *Proc. FMSB'08*, **5054** *LNCS*, 48–62. Springer, June 2008.

15. Nicola Bonzanni, K. Anton Feenstra, Wan Fokkink and Elzbieta Krepska. What can Formal Methods bring to Systems Biology? In: *Proc. FM'09*, **5850** *LNCS*, 16–22. Springer, 2009.
16. Masao Nagasaki, Ayumu Saito, Atsushi Doi, Hiroshi Matsuno, and Satoru Miyano. *Using Cell Illustrator and Pathway Databases.* Springer, April 2009.
17. Kelder T, Pico AR, Hanspers K, van Iersel MP, Evelo C, *et al.* Mining Biological Pathways Using WikiPathways Web Services. *PLoS ONE* **4**(7): e6447, 2009
18. Jasmin Fisher, Tom Henzinger, Maria Mateescu, and Nir Piterman. Bounded asynchrony: A biologically-inspired notion of concurrency. In *Proc. FMSB'08*, Cambridge, **5054** *LNCS* 17–32. Springer, June 2008.