

SBGN ANNUAL COMPETITION

PROJECT-TEAM CONTRAINTES

[HTTP://CONTRAINTES.INRIA.FR/](http://contraintes.inria.fr/)

FRENCH NATIONAL INSTITUTE FOR
RESEARCH IN COMPUTER SCIENCE AND
CONTROL

INRIA, PARIS-ROCQUENCOURT, FRANCE

SBGN EDITOR DEVELOPER: DRAGANA JOVANSKA



Category: Best SBGN software support - Completeness, Exactitude,
Validation

Software: BIOCHAM 3.1

15/08/2010

Contents

1	Introduction	2
2	BIOCHAM - BioChemical Abstract Machine	3
2.1	Installation	3
2.2	Constituents of a Biocham Model	3
2.3	Biocham objects	4
2.4	Biocham reaction and transport rules	5
2.5	Model reductions	6
3	SBGN Specification support	7
3.1	Completeness	7
3.2	Exactitude	10
3.3	Validation	12
4	Discussion	14
5	Conclusion	16
6	Appendix	18

1 Introduction

The Biochemical Abstract Machine (Biocham) is a software environment for modeling and analyzing biochemical systems [4]. Biocham is mainly composed of:

- a rule-based language for modeling biochemical systems (compatible with SBML and SBGN)
- different simulators (Boolean, differential, stochastic),
- a temporal logic based language to formalize the temporal properties of a biological system and validate models with respect to such specifications,
- unique features for developing/correcting/completing/reducing/coupling models, including the inference of kinetic parameters in high dimension from temporal logic constraints.

Biocham models can be imported and exported in SBML format, and are thus compatible with a large number of model repositories and other modeling software.

Since version 3.0, Biocham contains an SBGN compliant graphical editor which allows the users to *draw* their biochemical network. For this purpose, we have chosen to implement the SBGN specification and to support its standardization.

In the following sections, after a brief presentation of Biocham, we analyze the compliance of the Biocham graphical editor to the SBGN specification, and we underline in which aspects Biocham enforces some further consistency checks.

2 BIOCHAM - BioChemical Abstract Machine

2.1 Installation

Biocham is a free software protected by the GNU General Public License GPL version 2. This is an Open Source license that allows free usage of this software.

To download and install Biocham, go to its web page: <http://contraintes.inria.fr/Biocham/>. The current release is 3.1, which has a Reaction Graph Editor that follows the SBGN PD specification. There you can find instructions for the installation as well as the reference manual. The installation of Biocham is described in the INSTALL file of the distribution. Biocham is launched with the new graphical user interface that includes the SBGN graphical editor by using the command `biocham_gui` or by clicking the Biocham icon.

The distribution contains a directory of examples of models and queries. The examples can be browsed from the graphical user interface using the button “open model” and new models can be created using either the rule editor or the SBGN graphical editor using the button “new model”.

For any questions, please be free to contact us: `biocham@inria.fr`.

2.2 Constituents of a Biocham Model

A Biocham model is composed of a list of

- reaction rules (with or without kinetic expressions)

plus optionnally:

- a specification of the initial state,
- definitions of numerical parameters, macros, invariants, events,
- declarations of molecules and locations
- a specification of the system’s behavior by qualitative and/or quantitative temporal logic formulae

The rule-based language of Biocham is compatible with the Systems Biology Markup Language SBML and the SBGN PD notation for the graphical editor.

The temporal logic language is an original feature of Biocham for specifying temporal properties of the system [2], as known from biological experiments,

<i>object</i> =	<i>molecule</i> <i>molecule</i> :: <i>name</i>	located molecule
<i>molecule</i> =	<i>name</i>	
	<i>molecule</i> - <i>molecule</i>	molecular complex
	<i>molecule</i> ~{ <i>name</i> ,..., <i>name</i> }	modified molecule
	<i>gene</i>	
	(<i>molecule</i>)	
<i>gene</i> =	# <i>name</i>	

Table 1: Syntax of Biocham objects.

checking them automatically, or using them as constraints for searching parameter values and revising the model [6].

In the following sections, we only focus on the aspects relevant to SBGN, i.e. the syntax of objects and reactions.

2.3 Biocham objects

Biocham manipulates formal objects which represent chemical or biochemical compounds, ranging from small molecules to macromolecules, protein complexes and genes. Biocham objects can be used also to represent control variables and processes. The syntax of Biocham objects is defined by the grammar given in Table 1.

Molecules can be located in a symbolic location using the `::` operator, otherwise the molecule is supposed to be in the default location. The syntax of a molecule has four forms.

1. The first form is the simplest and the most flexible one. It concerns the case in which a molecule is simply given a (case sensitive) name.
2. The second form serves to denote multimolecular complexes with the linking operator `-`. This binary operator is assumed to be associative and commutative, hence the order of the elements in a complex does not matter. In the cases where one would like to distinguish between different orders of association, one should give names to the different complexes.
3. The third form serves to write modified forms of molecules, by attaching the set of modified sites of the protein with the operator `~`, like the set of phosphorylated sites for example. Several sets can be attached, the union is considered. The order of the elements is irrelevant.

<i>reaction</i> =	<i>kinetics for basic_reaction</i> <i>basic_reaction</i> <i>name : basic_reaction</i> <i>name : kinetics for basic_reaction</i>
<i>basic_reaction</i> =	<i>solution => solution.</i> <i>solution =[object]=> solution.</i> <i>solution =[solution => solution]=> solution.</i> <i>solution <=> solution.</i> <i>solution <=[object]=> solution.</i>
<i>solution</i> =	<i>_</i> <i>object</i> <i>integer*object</i> <i>solution + solution</i> <i>(solution)</i>

Table 2: Syntax of Biocham rules.

4. The fourth syntactical form is used to denote genes or gene promoters, with a name beginning with #. These objects are assumed to be unique, which has a consequence on the way reactions involving such objects are interpreted by Biocham (see below).

2.4 Biocham reaction and transport rules

Biocham reaction rules are used primarily to represent biochemical reactions and transport. They can be used also to represent state transitions involving control variables, or to represent the main effects of complete subsystems such as protein synthesis by DNA transcription without necessarily introducing RNAs in the model. The syntax of Biocham rules is defined by the grammar given in Table 2.

A solution is a multiset of molecular objects written as a linear expression. The character `_` denotes the empty solution.

A reaction rule involving a catalyst molecule (i.e. a molecule appearing in both the left and right-hand sides of the rule) can be written with the catalyst molecule between square brackets in the arrow.

In the Boolean semantics of Biocham models where one reasons on the presence and absence of molecules over time, the order and multiplicity (stoichiometry) of molecules in a solution and the reaction kinetics are ignored. In such a Boolean abstraction, a reaction transforms one solution matching the left-hand side of the rule, in another solution in which the objects of the right-hand side have been added. The molecules in the left-hand side of the rule which do not appear in the right-hand side may be non-deterministically present or consumed in the resulting solution. This convention reflects the capability of Biocham to reason about all possible behaviors of the system with unknown kinetic parameters [2, 3].

In the differential semantics (resp. stochastic semantics) of Biocham, the kinetic expressions in the reaction rules are used to associate to a Biocham model an ordinary differential equation (ODE), resp. a continuous-time Markov Chain (CTMC). If a rule is provided without kinetic expression, a mass action law kinetic with reaction rate 1 is assumed.

2.5 Model reductions

Better than developing one model is to relate it to other models by reduction or refinement relations.

Biocham contains four graph operation commands that can be used to formally define model reductions in a very flexible manner (ignoring the kinetics). These four operations are for merging and deleting molecular species and reaction rules. One sequence of such operations defines a model reduction.

Furthermore, the existence of such a reduction relation between two given models can be searched automatically (as a subgraph epimorphism problem) with good efficiency, as has been shown in [1] between pairs of models in the `biomodels.net` repository.

Accordingly, the Biocham Reaction Graph editor in the next release is going to integrate some features to define model reductions graphically and search model reductions between existing models.

3 SBGN Specification support

The Biocham Reactions Graph Editor tries to support the SBGN Process Diagram language. In the following sections, we will describe how and how much it supports it. There will be given visual proofs about the completeness, exactitude and the validation of the specification support, but in addition we recommend that you also check out Biocham and try its Reactions Graph Editor. There are some video tutorials available that can help you be more familiar with using the Biocham GUI.

3.1 Completeness

Biocham Reactions Graph Editor implements the SBGN Process Diagram language. We have chosen it because it corresponds the most to the way how the biochemical network can be described in Biocham.

Biocham Entity Pool Nodes

Using the SBGN PD Specification, we can express graphically the Biocham Molecule with a glyph "*Macromolecule*", which can be modified, so holds a glyph "*State variable*", showing the general state "Modification", but specifying the modified sites inside the "*Macromolecule*" glyph; can hold a glyph "*Unit of information*" showing its initial concentration; can be copied, so it's represented with the glyph of "*Labeled Clone Marker*", and if it's a multimer, it's drawn with the glyph "*Multimer*" and holds another glyph "*Unit of information*" to show the cardinality. This is illustrated in the following pictures. To express graphically a gene, we are using the "*Nucleic Acid Feature*" glyph, and we can apply the same settings as in the previous case with a glyph "*Macromolecule*". It means, it can be a multimer, it can be modified, and it can be a copy.

These are 2 kinds of general biocham molecules. It exist also a molecular complex in Biocham, that corresponds to the definition of a "*Complex*" glyph in SBGN PD language, thus this glyph is used. A Complex can contain macromolecules, nucleic acid features and complexes also. It can also be modified, a multimer, and a copy, too. The forth kind of object in Biocham is an empty solution, marked

with an underscore '_' ,and with the meaning of the SBGN PD glyph "Source and Sink". These are the 4 principal objects used in Biochem when modelling a biochemical network. We can also specify a location of a molecule, thus we use the glyph "Compartment" for drawing molecular locations if different then default.



Figure 1: Simple Macromolecule



Figure 2: Modified Macromolecule

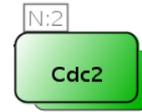


Figure 3: Multimer Macromolecule

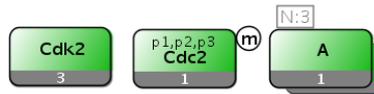


Figure 4: Copies of macromolecules in different states



Figure 5: Simple NucleicAcidFeature



Figure 6: Modified NucleicAcidFeature

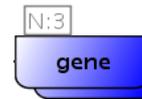


Figure 7: Multimer NucleicAcidFeature

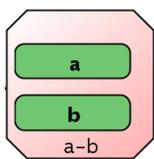


Figure 8: Simple Complex

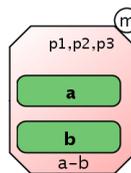


Figure 9: Modified Complex

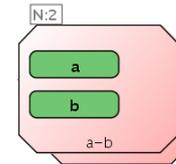


Figure 10: Multimer Complex



Figure 11: Source or Sink

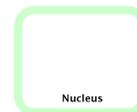


Figure 12: Compartment

Biocham Process Nodes

From the side of reactions, there are 7 principal reactions that can be described in Biocham language. Complexation, Decomplexation, Synthesis, Degradation, Phosphorylation, Dephosphorylation and Transport. To suit this formalism into SBGN language, we are using the glyph "Association" to describe the process of Complexation; the glyph "Dissociation" to describe the process of Decomposition; and the glyph "State Transition" to express all the other Biocham reactions. As in Biocham language we can express that a reaction is being affected by a catalyst or an inhibitor, we are using the glyph "Modulation" to graphically display this process affection, whether it's a positive or negative affection. These glyphs are illustrated in the following pictures.



Figure 13: Synthesis



Figure 14: Degradation



Figure 15: Phosphorylation

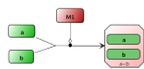


Figure 16: Association

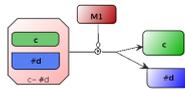


Figure 17: Dissociation

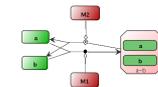


Figure 18: Reversible Association with 2 Modulators



Figure 19: Multimerization



Figure 20: Demultimerization



Figure 21: Transport

3.2 Exactitude

In the section of *Completeness* we have described how Biocham uses the SBGN PD language's glyphs. In this section we are going to show that not only it uses the glyphs accurately, but also their syntax and semantics, as described in the SBGN PD Specification in the *Chapter 3: Process Description Language Grammar*.

The Syntax

1. Node connectivity

Macromolecules, Nucleic Acid Features, Complexes, Multimers, Sink and Sources are the implemented SBGN Entity Pool Nodes in the Biocham Reaction Graph Editor. Sink and Source glyphs are represented with the same glyph, meaning the empty solution. Consumption arcs start from all of them, Production arcs end in all of them, and Modulation arcs start from all of them except from Sink and Sources, as an empty solution cannot be a Modulator in a Biocham reaction. Process, Association and Dissociation are the implemented SBGN Process Nodes in the Biocham Reaction Graph Editor. Production arcs start at the Process Node, Association(1) and Dissociation, Consumption arcs end in Process Node, Association and Dissociation(1), and the Modulation arcs end in Process Node, Association and Dissociation Node.

2. Containment definition

Complex and Compartment are the implemented SBGN Containers in Biocham. Macromolecule, Nucleic Acid Feature, Complex, and Multimer, can be drawn in Complex and Compartment. Sink/Source, Compartment, Process, Association, Dissociation, Consumption, Production, and Modulation can be drawn in a Compartment, but not in a Complex.

Semantic Rules

1. EPNs rules

All state variables in an EPN are unique and not duplicated within that node. If a state variable is used in one EPN, then it must be used in all equivalent EPNs, but in Biocham if the EPN is not modified, we don't draw to it an empty modification state. EPN's in Biocham are not orphaned. They can be drawn orphaned if copied on the Editor, and they can be removed from the Editor on user's interaction.

2. Process Node Rules-Flux Arcs, Association, Dissociation, Modulation, Reversible Processes

In Biocham we have the concept of "sideness" of a reaction, where the reactants are on the LHS and the products on the RHS. So all process nodes have LHS and RHS, and all LHS and RHS of a process are unique, otherwise there is a Biocham exception. The process in Biocham always constitute a balanced biochemical reaction. If some of the arcs have stoichiometry bigger than 1, then it is displayed, otherwise it is not displayed. Association is drawn as an irreversible process. Because in Biocham language can be written reversible Association, in the reaction Graph Editor it is drawn as 2 processes, Association and Dissociation in respect with the SBGN notation. The same is true for a Dissociation. In Biocham we can express the change of the basal rate at which a reaction happens, but we don't have a syntax with which we distinguish the increase or decrease of the process rate, so Modulators are being used. And when multiple modulators on a process node are being used, then their effect is additive. In Biocham we can write and draw reversible State Transitions. Modulation arc can not be reversible in Biocham.

3. Clonning

Macromolecule, Nucleic Acid Feature, Complex and Multimer can be cloned on the reaction graph to increase the graph's understandability and aesthetics, and in this case they are drawn as Labeled Clone Markers. There can be copies of reactions on the Graph Editor, too, respecting the SBGN notation. They are called dummy rules in Biocham.

4. Compartment spanning

One EPN usually belongs to only one compartment, but in case of a Transport reaction, it moves to another compartment, but as in Biocham language it's represented as a new molecule, in the Reaction Graph also it is drawn as a different EPN. The assignement to a compartment is defined in the molecule's name.

5. Submaps

They cannot yet be expressed in Biocham.

3.3 Validation

The validation of the syntax and the semantics of the drawn Reaction Graph is taking place almost 2 times during the process of modelling. First time it's during the drawing phase, because the drawing is controlled by the Editor. The users cannot draw whatever they want. They can draw only the things that are possible in the Editor, and what is SBGN and Biocham correct is made possible. The second phase of validation is during the synchronization of the Reaction Graph with Biocham. When a rule has been drawn and sent to Biocham for validation, if this rule is not semantically correct, a warning is being generated and the reaction is being removed from the Editor. In the following pictures, some validation cases are shown.



Figure 22: Wrong kinetics for the reaction

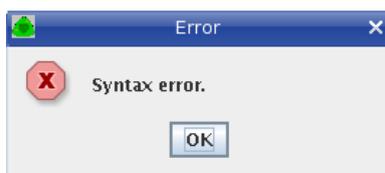


Figure 23: Wrong name for a reaction participant



Figure 24: Wrong name for a reaction participant

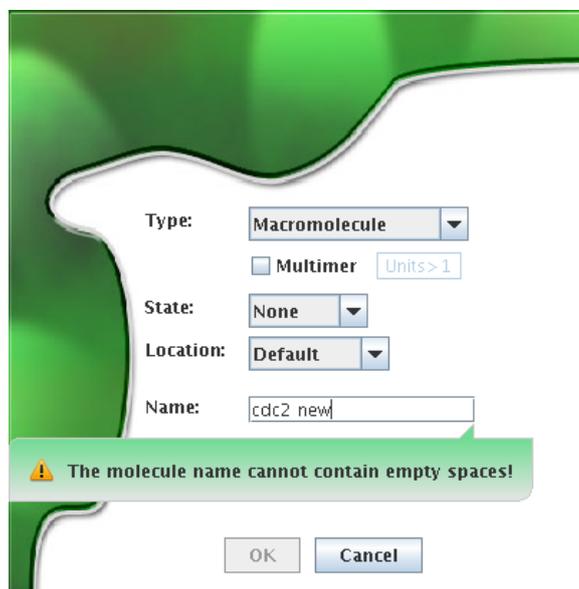


Figure 25: The Editor takes care for correct molecule's name



Figure 26: There is gene not on the RHS of the reaction

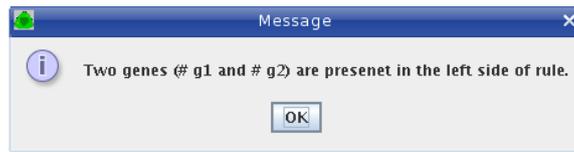


Figure 27: There are 2 genes present only on the LHS



Figure 28: Two genes not present on the RHS

4 Discussion

1. Nested Complexes

In the SBGN PD Specification, a Complex can contain a Complex. It can go to infinitive, where a Complex can contain multiple Complexes that contain Complexes, etc. Not just that is very difficult to present it graphically, we find that it's not so correct. First, it can result to one very big object on the reaction graph, and second, it's complicated. So, we have another solution for nested complexes. We don't draw its containing molecules, we just display the name of the complex, and its state, if it's modified or a multimer. Also, when we have a big network, the complexes, when they represent the containing molecules, are taking a lot of space. For the aesthetics reasons, we have an option for the user if he prefers to compact his network, and here all the complexes take the compact form. This is illustrated in the following pictures.

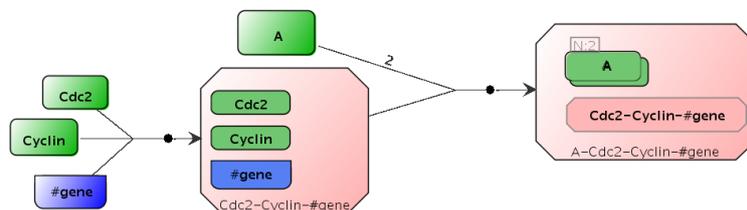


Figure 29: Nested Complex

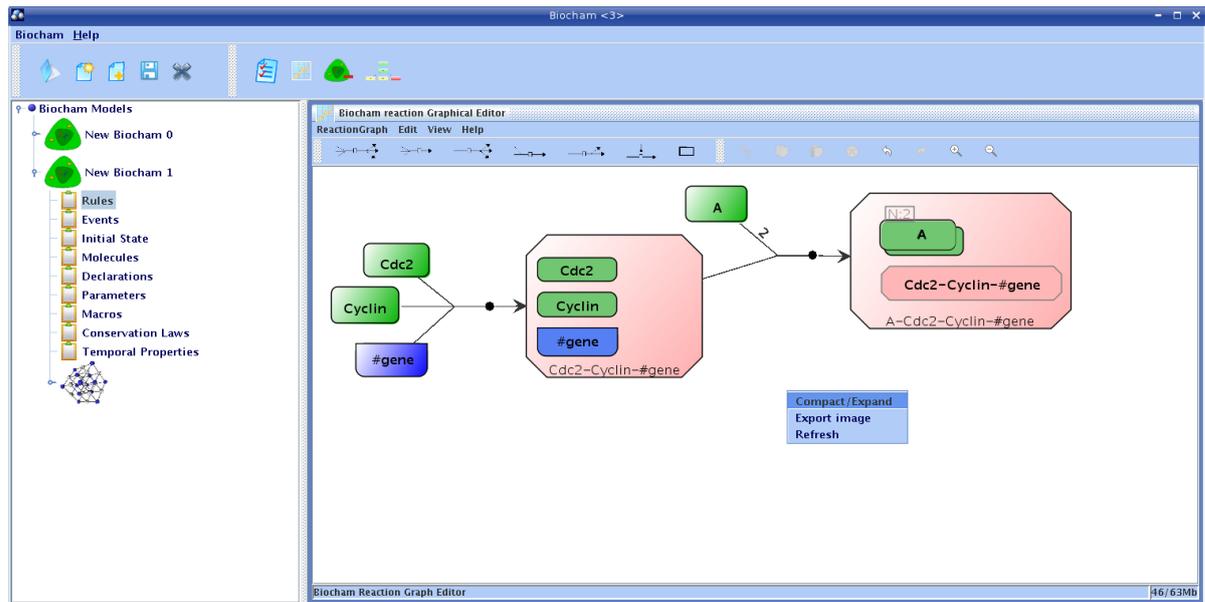


Figure 30: Option to show the Compacted Form of Reaction Graph

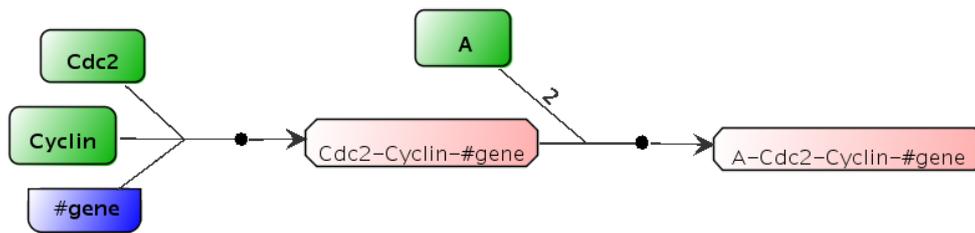


Figure 31: Compact form of Complex

2. Model Reduction Notation

In Biocham today, we have functions for model reductions that are going to be implemented in the graphical editor also. We have some ideas how to do it graphically, but it's not bad idea to have some graphical notation about it also.

3. Layout

We find that the rules for layout defined by the SBGN PD Specifications are sometimes contradictory, in the meaning for example, if we respect the LHS-RHS rule, we get more crossings of edges, etc. In Biocham, we have automatic

layout, that tries to respect the SBGN rules, but we don't have yet layouts when a model is being loaded for the whole reaction graph. It's to be done for the next release.

5 Conclusion

By integrating an SBGN PD compliant editor in a modeling environment such as Biocham we get more than the sum of the tools.

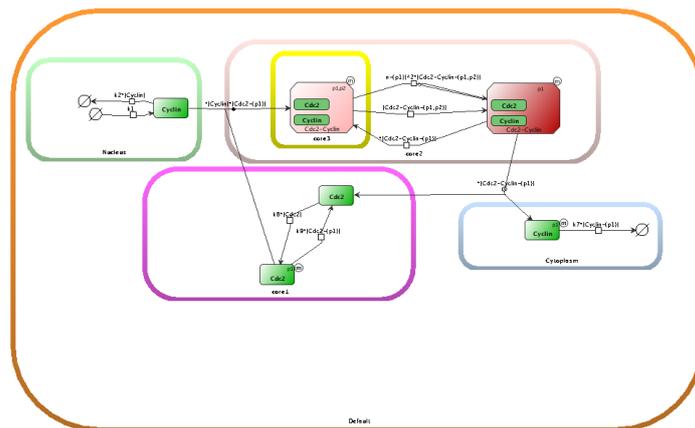
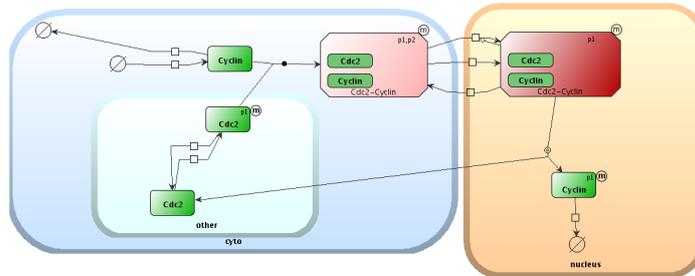
On the one hand, it is easier for the biologist to create and maintain his models simply by drawing SBGN interaction networks and get transparent access to the whole Biocham features for model analyses.

On the other hand, the Biocham analyses features enforce further consistency checks on SBGN diagrams that take into account the semantics of the interactions.

References

- [1] Steven Gay, Sylvain Soliman, and François Fages. A graphical method for reducing and relating models in systems biology. *Bioinformatics*, October 2010.
- [2] François Fages and Sylvain Soliman. Formal cell biology in BIOCHAM. In M. Bernardo, P. Degano, and G. Zavattaro, editors, *8th Int. School on Formal Methods for the Design of Computer, Communication and Software Systems: Computational Systems Biology SFM'08*, volume 5016 of *Lecture Notes in Computer Science*, pages 54–80, Bertinoro, Italy, February 2008. Springer-Verlag.
- [3] Nathalie Chabrier and François Fages. Symbolic model checking of biochemical networks. In Corrado Priami, editor, *CMSB'03: Proceedings of the first workshop on Computational Methods in Systems Biology*, volume 2602 of *Lecture Notes in Computer Science*, pages 149–162, Rovereto, Italy, March 2003. Springer-Verlag.
- [4] Laurence Calzone, Nathalie Chabrier-Rivier, François Fages, and Sylvain Soliman. Machine learning biochemical networks from temporal logic properties. In Gordon Plotkin, editor, *Transactions on Computational Systems Biology VI*, volume 4220 of *Lecture Notes in Bioinformatics*, pages 68–94. Springer-Verlag, November 2006. CMSB'05 Special Issue.
- [5] Aurélien Rizk, Grégory Batt, François Fages, and Sylvain Soliman. On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology. In Monika Heiner and Adeline Uhrmacher, editors, *CMSB'08: Proceedings of the fourth international conference on Computational Methods in Systems Biology*, volume 5307 of *Lecture Notes in Computer Science*, pages 251–268. Springer-Verlag, October 2008.
- [6] Aurélien Rizk, Grégory Batt, François Fages, and Sylvain Soliman. A general computational method for robustness analysis with applications to synthetic gene networks. *Bioinformatics*, 12(25):il69–il78, June 2009.

6 Appendix



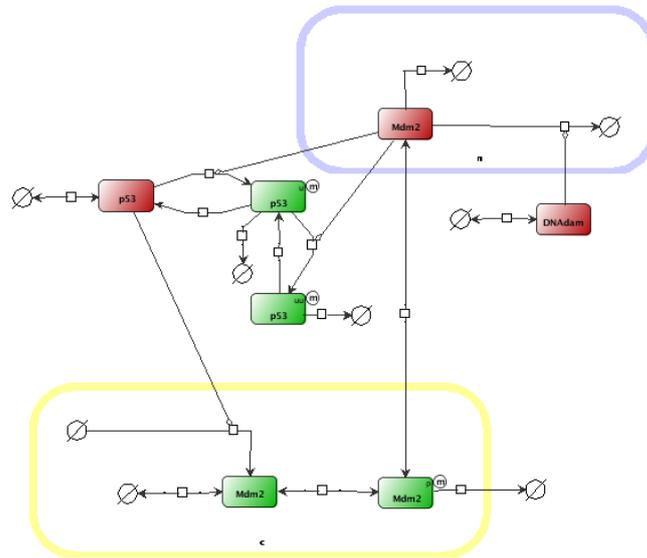


Figure 32: p53Mdm2 Model

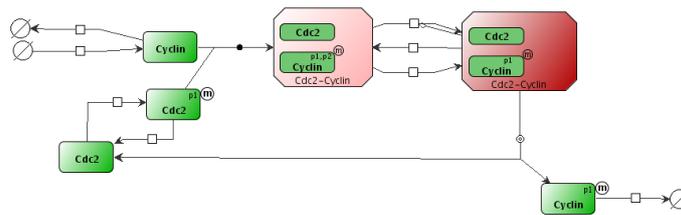


Figure 33: tyson91 Model Model