Published in final edited form as:

Essays Biochem. 2008 ; 45: 211–222. doi:10.1042/BSE0450211.

Standards and Ontologies in Computational Systems Biology

Herbert M. Sauro^{*} and **Frank Bergmann**

Department of Bioengineering, William H. Foege Building, Box 355061 Seattle, USA, 98195-5061

Abstract

With the growing importance of computational models in systems biology there has been much interest in recent years to develop standard model interchange languages that permit biologists to easily exchange models between different software tools. In this chapter two chief model exchange standards, SBML and CellML are described. In addition, other related features including visual layout initiatives, ontologies and best practices for model annotation are discussed. Software tools such as developer libraries and basic editing tools are also introduced together with a discussion on the future of modeling languages and visualization tools in systems biology.

Keywords

SBML; CellML; Ontology; MIRIAM; SBGN; TEDDY; MIASE; Standards; libSBML; SBMLEditor; Biomodels.net; SBO

1 Introduction

What do screw threads, Transistor–Transistor Logic (TTL) and the Systems Biology Markup Language (SBML) have in common? Standardization of course. Standardization is one of those overlooked but essential features of modern life without which our world would largely come to a grinding halt. The main benefit of standardization is increased productivity. Thus the introduction in the 19th century of standard screw threads by Whitworth in the UK and Sellers in the US made the manufacture of the humble bolt and screw cheaper and more reliable. Likewise the introduction of TTL circuits in 1962 by Texas Instruments led to the standardization of the electrical characteristics for digital circuits and allowed the development of reliable and cheap interchangeable logic parts for the growing electronics industry. History has shown us repeatedly, that on the whole, standardization is a good thing.

In systems biology there are already proposed standards for accomplishing certain things. In particular model exchange has been an active area of discourse where at least two standards, SBML and CellML have been proposed as a means for users to exchange models between different software tools. Prior to the year 2000 each software tool used its own format to store models. This meant that it was very difficult to move a model from one software tool to another and worse still, if a software tool ceased development and was no longer supported (which is all too common in systems biology), it would become imperative to translate one's models to a different tool. Clearly this state of affairs was quite unproductive and out of these obvious shortcomings, a number of groups set out to gather community support to develop a standard that model developers would be happy to use. This meant that a model could be stored in a format that was independent of the software tool. There was an early effort in 1998 by the BTK (BioThermoKinetics) group to standardize on a practical format for exchanging models

^{*}Address for Correspondence: Herbert M Sauro, Department of Bioengineering, William H. Foege Building, Box 355061, Seattle, WA, USA, 98195-5061 Phone: (206) 685 2119 Fax: (206) 685 3300 hsauro@u.washington.edu, web page www.sys-bio.org.

between two widely used tools, Gepasi (17) and SCAMP (23). Around the same time, bioengineers at the University of Auckland began investigating the role that XML (8) could play in defining a standard for exchanging computational models in order to reduce errors that appeared frequently in published models. From the Auckland team emerged CellML (15). Members from the BTK group subsequently took their experience and contributed significantly to the other major model exchange standard, called SBML (9). SBML was developed in 2000 at Caltech, Pasadena as a result of funding received from the Japanese ERATO program. Both CellML and SBML are today viewed as the main de facto standards for exchanging cellular network models. There are however fundamental differences between the approaches that CellML and SBML take in the way models are represented which we will touch upon in this chapter.

2 Quantitative Approaches

2.0.1 Quantitative Models Based on Differential Equations

Many simulation models in systems biology are constructed using differential equations. These equations describe the continuous rate of change of molecular species in time. Other researchers use a stochastic based description either by explicitly modeling the particulate nature of the cellular milieu or by simply adding noise to differential equations.

When modeling systems based on differential equations, many researchers will express these models using the following equation:

$$\frac{dS}{dt} = N\nu(S, p) \quad (1)$$

where S is the vector of molecular species concentrations, N , the stoichiometry matrix; ν the rate vector and p a vector of parameters which can influence the evolution of the system. Many software tools will permit users to enter models as a list of reactions and then automatically generate the mathematical model (23;22;24).

3 De Facto Standards

Although there has yet been no ratification of a standard exchange format by an official body such as OASIS or ISO, both SBML and CellML are considered as de facto standards simply because they are so widely used. In this section we will consider each standard, although our focus will mainly be on SBML and related technologies.

3.1 CellML

CellML (15) represents cellular models using a mathematical description. In addition, CellML represents entities using a component based approach where relationships between components are represented by connections. The literal translation of the mathematics however goes much further, in fact the representation that CellML uses is very reminiscent of the way an engineer might wire up an analog computer to solve the equations (though without specifying the integrators). As a result CellML is very general and in principle could probably represent any system that has a mathematical description (and not just the kind indicated by equation (1)). CellML is also very precise in that every item in a model is defined explicitly. However, the generality and explicit nature of CellML also results in increased complexity especially for software developers.

Given the general component / relationship based approach of CellML, models defined using CellML introduce biological information by way of metadata. As CellML is a XML based

language there are natural ways for introducing metadata. Most prominent is to embed annotations using the Resource Description Framework (RDF, <http://www.w3.org/RDF/>). RDF allows the description of specific CellML elements. Within the individual RDF descriptions the metadata is further qualified relying on a standardized set of terms from the Dublin Core Metadata Initiative (DCMI, <http://dublincore.org/documents/dcmi-terms/>). These terms (i.e.: authors, dates, titles and so on) can then be easily mined by other applications by just looking over the model. Apart from the general approach that relies on RDF and the Dublin Core, CellML metadata can also be in the form of a BioPAX description. BioPAX (or Biological Pathway Exchange, <http://www.biopax.org>) is an ontology that defines biological pathway data, such as metabolic pathways or molecular interactions. BioPAX thus represents a perfect complement to CellMLs rigorous mathematical formalism.

The CellML team has amassed a very large suite (hundreds) of models (www.cellml.org/examples/repository/) which provides many real examples of CellML syntax. This is an extremely useful resource for the community.

3.2 SBML

Whereas CellML attempts to be highly comprehensive, SBML was designed to meet the immediate needs of the modeling community and is therefore more focused on a particular problem set. One result of this is that the standard is simpler compared to CellML although more recent revisions add new functionality so that the difference in complexity between CellML and SBML is becoming less significant. Like CellML, SBML is based on XML, however unlike CellML, it takes a different approach to representing cellular models. The way SBML represents models closely maps the way existing modeling packages represent models. Whereas CellML represents models as a mathematical wiring diagram, SBML represent models as a list of chemical transformations. Since every process in a biological cell can ultimately be broken down into one or more chemical transformations this was a natural representation to use. However SBML does not have generalized elements such as components and connections, SBML employs specific elements to represent spatial compartments, molecular species and chemical transformations. In addition to these, SBML also has provision for rules which can be used to represent constraints, derived values and general math which for one reason or another cannot be transformed into a chemical scheme.

SBML, like any standard, evolves with time (5). Major revisions of the standard are captured in levels, while minor modifications and clarifications are captured in versions. An example of a major change within the standard would be the use of MathML in level two of SBML, whereas level one encoded infix strings to denote reaction rates and rules. A minor change on the other hand would for example be the introduction of semantic annotations (See section 3.3.2) that can be added to SBML level two version two, whereas this was not possible in a supported fashion in earlier versions. As of the time of this writing, SBML Level three is still in development. With Level three the standard will develop in a extensible manner. This means there will be a set of core features that must be supported around which additional features, such as spatial modeling, can be included.

For a well annotated repository of SBML models see the Biomodels.net database <http://www.ebi.ac.uk/biomodels/>.

3.2.1 SBML Development Tools—The success of SBML over competing standards can be ascribed in part to **libSBML** — <http://sbml.org/software/libsbml/>. LibSBML is a software library provided by the SBML Team. The software library is based around a C/C++ core, with wrappers provided for many programming languages. Furthermore the library is available for Windows and POSIX operating systems, and thus can be used virtually anywhere. With an

abundance of documentation and available examples, software developers can readily use libSBML for their SBML support.

By using libSBML a developer no longer has to worry about the level and version of an SBML document the software has to read, as libSBML encapsulates this and is even able to convert SBML models into the desired form. Hence a software developer can focus on how to interpret computational models rather than concerning themselves with the mechanics of reading and writing SBML. At the time of this writing, libSBML has released version 3. New in this version are features to validate the model, such as unit consistency checking, or checks on whether the model is over determined. LibSBML now also provides support for MIRIAM compatible annotations (14).

SBMLEditor (21), developed by Nicolas Rodriguez at the European Bioinformatics Institute, represents a low level SBML editor. The curators of the BioModels.net database (13), a repository of well annotated, curated and simulatable models, use this tool to annotate and curate the models. A user of the editor can view the XML tree, and make changes to the model. The software can convert MathML to the infix notation and back, in order to facilitate editing of kinetic laws, initial assignments, rules and constraints. SBMLEditor can check for consistency and the validity of the model by applying the libSBML consistency validators. SBMLEditor also features the SBW menu (2): this allows a user to send the model for further analysis, simulation or visualization to any installed module of the Systems Biology Workbench (24). The upcoming version of the SBMLEditor will support the Systems Biology Ontology, as well as SBML Level two Version three.

3.2.2 Example—Here we give an example of an SBML model. The following SBML text encodes a very basic model:

Node₀ as well as Node₂ have been chosen as model boundaries, that is they have fixed concentrations. The model features two reactions, converting Node₀ into Node₁ and Node₁ into Node₂, both employ mass action kinetics. It should be mentioned that in SBML each species “lives” in a compartment.

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns = "http://www.sbml.org/sbml/level2" level = "2" version = "1">
  <model id = "LinearChain" name = "Simple Linear Pathway">
    <listOfCompartments>
      <compartment id = "compartment" size = "1"/>
    </listOfCompartments>
    <listOfSpecies>
      <species id = "Node0" boundaryCondition      = "true"
              initialConcentration                = "10"
              compartment                          = "compartment" />
      <species id = "Node1" boundaryCondition      = "false"
              initialConcentration                = "0"
              compartment                          = "compartment" />
      <species id = "Node2" boundaryCondition      = "true"
              initialConcentration                = "0"
              compartment                          = "compartment" />
    </listOfSpecies>
  </model>
</sbml>
```

```

<listOfParameters>
  <parameter id = "k" value = "0.1"/>
</listOfParameters>
<listOfReactions>
  <reaction id = "J0" reversible = "false">
    <listOfReactants>
      <speciesReference species = "Node0" stoichiometry = "1"/>
    </listOfReactants>
    <listOfProducts>
      <speciesReference species = "Node1" stoichiometry = "1"/>
    </listOfProducts>
    <kineticLaw>
      <math xmlns = "http://www.w3.org/1998/Math/MathML">
        <apply>
          <times/>
          <ci> k </ci>
          <ci> Node0 </ci>
        </apply>
      </math>
    </kineticLaw>
  </reaction>
  <reaction id = "J1" reversible = "false">
    <listOfReactants>
      <speciesReference species = "Node1" stoichiometry = "1"/>
    </listOfReactants>
    <listOfProducts>
      <speciesReference species = "Node2" stoichiometry = "1"/>
    </listOfProducts>
    <kineticLaw>
      <math xmlns = "http://www.w3.org/1998/Math/MathML">
        <apply>
          <times/>
          <ci> k </ci>
          <ci> Node1 </ci>
        </apply>
      </math>
    </kineticLaw>
  </reaction>
</listOfReactions>
</model>
</sbml>

```

Even this simple example shows how the structured XML format, which is optimally readable by a machine, becomes unwieldy for humans to read. The format becomes even more complicated to read once we start annotating the model, say by identifying "Node₁" as a rat epidermal growth factor:

```

[...]
<species metaid="aboutNode1"
  Id = "Node1"
  compartment = "compartment"

```

```

        boundaryCondition      = "false"
        initialConcentration = "0">
<annotation>
  <rdf:RDF xmlns:rdf      = "http://www.w3.org/1999/02/22rdfsyntaxns#"
          xmlns:bqbiol = "http://biomodels.net/biologyqualifiers/" >
    <rdf:Description rdf:about="#aboutNode1">
      <bqbiol:hasPart>
        <rdf:Bag>
          <rdf:li rdf:resource="http://www.uniprot.org/#P07522" />
        </rdf:Bag>
      </bqbiol:hasPart>
    </rdf:Description>
  </rdf:RDF>
</annotation>
</species>
[...]
```

The problem is not so much that the format is unintelligible - it is simply a problem of long running scopes which have to be remembered. Fortunately SBML does not have to be written by humans. Looking at the home page of the Systems Biology Markup Language (<http://sbml.org>) we find more than 120 software applications available supporting SBML and many modeling tools among them. A recent review compares some of these tools (1).

3.3 Other Related Standards

3.3.1 Graphical Layout—Graphical modeling applications (2) routinely enhance computational models by layout annotations. Recently the SBML community has decided on a common standard on how to embed the layout information within SBML. The layout extension (7) allows a model to store the size and dimension of all model elements, along with textual annotations and reactions. Originally the view was to embed the layout extension in a model annotation for Level 2 versions of SBML but with the upcoming Level 3 the layout extension will be added to the SBML as a first-class construct. LibSBML has been modified to provide access to all elements of the Layout Extension. Also several reference implementations exist (2;3).

Whereas the layout extension is concerned with representing simple elements, the Systems Biology Graphical Notation (SBN) (<http://sbgn.org>) aims to standardize the visual language of computational models unambiguously. While this standard is still in development and strictly speaking independent of the SBML effort, experience in other fields such as electrical engineering has demonstrated the essential need for standardizing the visual notation for representing models in diagrammatic form.

3.3.2 MIRIAM—Model Definition Languages such as SBML and CellML target the exchange of models. They aim to pass on the quantitative computational model from one software tool to another. However these description formats do not concern themselves with semantic annotations. Semantic annotations here would uniquely identify model constituents, information about the relationship between model elements or could be basic identifications of model author and the date of last modification. These annotations can be interpreted even by software without knowledge about the model definition languages. Both SBML and CellML have launched efforts to remedy this problem. Both communities agreed on the Minimum Information Requested In the Annotation of biochemical Models (MIRIAM,(14)). These annotations aim to further the confidence in quantitative biochemical models, making it easier and more precise to search for particular biochemical models, enabling researchers to identify

biological phenomena captured by a biochemical model and perhaps most importantly to facilitate model reuse and model composition.

In order to call a model MIRIAM compliant, the model has to be encoded in a standard format, such as SBML. Furthermore it needs to be tied to a reference description, describing the properties and results that can be obtained from the model. Parameters of the computational model have to be provided so that the model can be loaded into a simulation environment where the results can be reproduced. Other information that has to be provided is a name for the model, the creator of the model the date and time of the last modification as well as a statement about the terms of distribution.

3.3.3 SBO – Systems Biology Ontology—In order to assign meaning to model constituents an ontology specific to Systems Biology has been developed (19): The Systems Biology Ontology (SBO, <http://www.ebi.ac.uk/sbo/>). The controlled vocabulary consists of two relationships: is-part-of and is-a. Qualifying model participants, say as enzyme, macromolecule, metabolites or small species such as ions will make it easy to generate meaning from the model. It will make the generation of standard visual notations such as SBN possible. Moreover it presents a solution on how to interpret the model computationally, as the SBO allows tagging a model as continuous, discrete or logical model. One could even go a step further, making kinetic interaction in a model obsolete, by referencing that the rate law is one specified by an ontology identifier (e.g.: tagging a reaction as following Henri-Michaelis Menten enzyme kinetics and specifying the parameters). Since SBML Level two Version three all SBML elements feature an optional sboTerm attribute, which makes tagging elements with the corresponding SBO term straightforward. The SBO is community driven and new terms or modifications to the existing ontology can be requested by the community.

4 Future Prospects and Conclusion

The most recent developments in CellML and particularly the SBML communities revolve around the creation of ontologies and refining the exchange semantics. Apart from classifying model constituents with an appropriate ontology, one of the current areas of interest is describing the dynamical behavior of a model. The "Terminology for the Description of Dynamics" (TEDDY, <http://www.ebi.ac.uk/compneur-srv/teddy/>) provides a rich ontology to describe and quantify the behavior a computational model is able to exhibit (e.g.: the characteristics of a model could describe bifurcation behavior where the functionality of a model could be described as featuring oscillations or switch behavior). However knowing that a model exhibits interesting behavior is not enough: more information is needed in order to recreate that behavior. The "Minimum Information About a Simulation Experiment" (MIASE, <http://www.ebi.ac.uk/compneur-srv/miase/>) project focuses in this problem. MI-ASE will help to describe the simulation algorithms and the simulation tool used along with all needed parameter settings. In order to do so it will use the Kinetic Algorithm Ontology (KiSAO) that relates simulation algorithms and methods to each other. As these ontologies are just being formulated, it will be interesting to see how they progress and are taken up by the community.

Although most recent developments in standardization have focused on the use of XML to represent models, there is a long tradition in the field to describing models using human readable text based formats. Indeed the very first simulator BIOSSIM, (6), allowed a user to describe a model using a list of reaction schemes. Variants of this have been employed by a number of simulators since, including, SCAMP (23), Jarnac (22), E-Cell (25) and more recently PySCeS (20). Being able to represent models in a human readable format offers many advantages, including conciseness, portability and ease of manipulation via a simple text editor.

There has also, in recent years, been a movement (18;16;4) to develop models based on the idea that cellular pathways, particularly signalling pathway, operate on a different molecular scale. This view focuses on the idea that the number of actual states in a pathway expands exponentially with the number of atomic and covalent modification states. This view represents a significant departure from the traditional picture of a biochemical pathway. In many cases the number of states can be extremely large which means that the average number of molecules in each state can be correspondingly extremely low. The number of connections is likewise very large. This approach necessitates a different method of modeling and in fact there is currently no theoretical framework that can adequately describe the dynamics of such an assembly. In view of this dramatic change in how we perceive signalling pathways, the traditional methods (23;11;12;10) that are used to define a 'signalling pathway' are unworkable and efforts have been made to develop rule based methods to describe such systems. The most well known of these is BioNetGen – Biological Network Generator, (4) – which allows a user to define the rules by which states and transformations are defined; computer software is then used to expand the set of rules into the full state model. Interchange standards such as SBML and CellML will need a significant revision to deal with rule based models.

In this chapter we have briefly summarized some of the developments and future prospects for model interchange in systems biology. Establishing standards is at the best of times very difficult, the process of acceptance is largely sociological and many factors contribute to the acceptance of a standard by a community. Although a small but significant minority of biologists are now publishing models in one of the two main interchange formats, many of the models we see published in peer-reviewed journals are published either in proprietary formats or are simple listings of equations in an appendix. As computational modeling becomes more important in biology, and as databases such as biomodels.net make models more accessible, this will undoubtedly change.

5 Recommended Resources

Three web sources which are of interest to readers of this chapter include:

<http://www.cellml.org> This is the main CellML site. It has a very rich set of models expressed in CellML including specifications for the standard and pointers to software toolkits.

<http://www.sbml.org> This is the main SBML site. The site has ample documentation, examples illustrating how SBML is and should be used. In addition it has a rich set of software tools, in particular libSBML, which allows developers to easily add SBML support to their tools.

<http://www.sys-bio.org> This is the main SBW (Systems Biology Workbench) site. The latest versions for SBW, developer documentation, example models, screen shots, user guides can be obtained from this site. A link to the main sourceforge site is given where all the source code for SBW is made available.

6 Summary

- Model exchange standards allow the free flow of computational models between different researchers.
- A variety of proposed standards exist, in particular, SBML and CellML and the most important.
- Standards such as CellML and SBML has enabled the development of model repositories such as [Biomodels.net](http://biomodels.net) and comparison sites such as found at www.sys-bio.org.

- The initial SBML and CellML standards has spawned a variety of other initiatives, including the development of graphical standards, model behavior standards, and new ontologies such as SBO.

Biography

Herbert M. Sauro

HERBERT SAURO was originally educated as a biochemist/ microbiologist but became interested in the use of simulation and theory to understand cellular networks after accidentally coming across a paper by David Garkinkel on the simulation of glycolysis. He wrote one of the first biochemical simulators for the PC (SCAMP) in the 1980s to assist work on extending metabolic control analysis (a theory closely related to biochemical systems theory) with David Fell. He also did postdoctoral work with Henrik Kacser in Edinburgh. However, with the lack of community interest in systems biology during the late 80s and early 90s, he left science to start a successful software company and offer consultancy work to finance firms in the UK. With the surge in interest in systems biology in the US in the late 90s, he returned to science by securing a position at Caltech to assist in the development of the Systems Biology Markup Language. He now works as an associate professor in the department of bioengineering at the University of Washington, Seattle where his interests focus on software, cellular control systems and synthetic biology. His e-mail address is hsauro@u.washington.edu.

Frank Bergmann

FRANK BERGMANN is currently a PhD student under the supervision of Herbert Sauro at the Keck Graduate Institute / University of Washington. He received his first degree in computer science from the Johann Wolfgang Goethe University Frankfurt, Germany. For his diploma he specialized in computer graphics and carried out his senior thesis on visualization of reaction-diffusion systems in biology. He is the lead developer for the Systems Biology Workbench and his PhD is concerned with the development of tools and applications of computer science to Systems Biology. His e-mail address is fbergman@u.washington.edu, and his web page is <http://public.kgi.edu/~fbergman>.

Acknowledgements

We would first like to acknowledge the generous support from the Japan Science and Technology Agency, DARPA (BAA01-26 Bio-Computation), the US Department of Energy GTL program and NIGMS program 1R01GM081070-01. We would also like to thank Anastasia Deckard for critically reading the manuscript.

References

1. Alves R, Antunes F, Salvador A. Tools for kinetic modeling of biochemical networks. *Nat Biotechnol* 2006 Jun;24(6):667–672. [PubMed: 16763599]
2. Bergmann FT, Vallabhajosyula RR, Sauro HM. Computational tools for modeling protein networks. *Current Proteomics* 2006 October;3(17):181–197.
3. Deckard A, Bergmann FT, Sauro HM. Supporting the SBML layout extension. *Bioinformatics* 2006;22(23):2966–2967. [PubMed: 17038346]
4. Faeder JR, Blinov ML, Goldstein B, Hlavacek WS. Rule-based modeling of biochemical networks. *Complexity* 2005;10:22–41.
5. Finney A, Hucka M. Systems biology markup language: Level 2 and beyond. *Biochemical Society Transactions* 2003;31(6):1472–1473. [PubMed: 14641091]
6. Garfinkel D. A machine-independent language for the simulation of complex chemical and biochemical systems. *Comput. Biomed. Res* 1968;2:31–44. [PubMed: 5743538]

7. Gauges R, Kummer U, Sahle S, Wegner K. A model diagram layout extension for SBML. *Bioinformatics* 2006;22(15):1879–1885. [PubMed: 16709586]
8. Harold, ER.; Means, ES. XML in a Nutshell. OREILLY; 2001.
9. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr J-H, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Le Novère N, Loew LM, Lucio D, Mendes P, Mjolsness ED, Nakayama Y, Nelson MR, Nielsen PF, Sakurada T, Schaff JC, Shapiro BE, Shimizu TS, Spence HD, Stelling J, Takahashi K, Tomita M, Wagner J, Wang J. The Systems Biology Markup Language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics* 2003;19:524–531. [PubMed: 12611808]
10. Kitano H, Funahashi A, Matsuoka Y, Oda K. Using process diagrams for the graphical representation of biological networks. *Nat Biotechnol* 2005 Aug;23(8):961–966. [PubMed: 16082367]
11. Kohn KW. Molecular interaction map of the mammalian cell cycle control and dna repair systems. *Mol Biol Cell* 1999 Aug;10(8):2703–2734. [PubMed: 10436023]
12. Kohn KW, Aladjem MI, Kim S, Weinstein JN, Pommier Y. Depicting combinatorial complexity with the molecular interaction map notation. *Mol Syst Biol* 2006;251–51
13. Le Novère N. Model storage, exchange and integration. *BMC Neurosci* 2006 Oct;7:S11. [PubMed: 17118155]
14. Le Novère, Nicolas; Finney, Andrew; Hucka, Michael; Bhalla, UpinderS; Campagne, Fabien; Collado-Vides, Julio; Crampin, EdmundJ; Halstead, Matt; Klipp, Edda; Mendes, Pedro; Nielsen, Poul; Sauro, Herbert; Shapiro, Bruce; Snoep, JackyL; Spence, HughD; Wanner, BarryL. Minimum information requested in the annotation of biochemical models (MIRIAM). *Nature biotechnology* 2005;23(12):1509–1515.
15. Lloyd CM, Halstead MD, Nielsen PF. CellML: its future, present and past. *Prog Biophys Mol Biol* 2004;85:433–450. [PubMed: 15142756]
16. Lok L, Brent R. Automatic generation of cellular reaction networks with molecuizer 1.0. *Nat Biotechnol* 2005 Jan;23(1):131–136. [PubMed: 15637632]
17. Mendes P. GEPASI: A software package for modelling the dynamics, steady states and control of biochemical and other systems. *Comput. Applic. Biosci* 1993;9:563–571.
18. Morton-Firth CJ, Bray D. Predicting temporal fluctuations in an intracellular signalling pathway. *J Theor Biol* 1998 May;192(1):117–128. [PubMed: 9628844]
19. Le Novère N, Courtot M, Laibe C. Adding semantics in kinetics models of biochemical pathways. Proceedings of the 2nd International Symposium on experimental standard conditions of enzyme characterizations. 2007 <http://www.beilstein-institut.de/?id=196&L=3>
20. Olivier BG, Rohwer JM, Hofmeyr JH. Modelling cellular systems with pyses. *Bioinformatics* 2005;21:560–561. [PubMed: 15454409]
21. Rodriguez, Nicolas; Donizelli, Marco; Le Novre, Nicolas. Sbmleditor: effective creation of models in the systems biology markup language (sbml). *BMC Bioinformatics* 2007;8(1):79. [PubMed: 17341299]
22. Sauro, HM. In: Hofmeyr, J-HS.; Rohwer, JM.; Snoep, JL., editors. Jarnac: A system for interactive metabolic analysis; Animating the Cellular Map: Proceedings of the 9th International Meeting on BioThermoKinetics; Stellenbosch University Press; 2000.
23. Sauro HM, Fell DA. Scamp: A metabolic simulator and control analysis program. *Mathl. Comput. Modelling* 1991;15:15–28.
24. Sauro HM, Hucka M, Finney A, Wellock C, Bolouri H, Doyle J, Kitano H. Next generation simulation tools: The systems biology workbench and biospace integration. *OMICS* 2003;7(4):355–372. [PubMed: 14683609]
25. Tomita M, Hashimoto K, Takahashi K, Shimizu TS, Matsuzaki Y, Miyoshi F, Saito K, Tanida S, Yugi K, Venter JC, Hutchison CA. E-cell: software environment for whole-cell simulation. *Bioinformatics* 1999;15:72–84. [PubMed: 10068694]