Evolving a lingua franca and associated software infrastructure for computational systems biology: the Systems Biology Markup Language (SBML) project

M. Hucka, A. Finney, B.J. Bornstein, S.M. Keating, B.E. Shapiro, J. Matthews, B.L. Kovitz, M.J. Schilstra, A. Funahashi, J.C. Doyle and H. Kitano

Abstract: Biologists are increasingly recognising that computational modelling is crucial for making sense of the vast quantities of complex experimental data that are now being collected. The systems biology field needs agreed-upon information standards if models are to be shared, evaluated and developed cooperatively. Over the last four years, our team has been developing the Systems Biology Markup Language (SBML) in collaboration with an international community of modellers and software developers. SBML has become a de facto standard format for representing formal, quantitative and qualitative models at the level of biochemical reactions and regulatory networks. In this article, we summarise the current and upcoming versions of SBML and our efforts at developing software infrastructure for supporting and broadening its use. We also provide a brief overview of the many SBML-compatible software tools available today.

1 Introduction

It may seem as though computational modelling in biology is a development born of the digital age. In fact, the application of computational techniques to the simulation and analysis of biological systems has a history dating to the earliest analogue and even mechanical computers [1–4]. The recent resurgence of interest in quantitative modelling [5–19] can be attributed at least partly to the greater power afforded by modern information technology [20], but even more to the explosion of data brought about by modern molecular techniques. It is now clear to many researchers that future progress in understanding biological function rests inescapably in the development and application of computational methods [21–30].

Practical computational modelling requires the use of software tools. To be useful as formal embodiments of our understanding of biological systems [31], computational models must be put into a form that can be communicated effectively between the software tools used to work with

them. This format must help overcome a number of problems facing a systems biologist:

• Users often need to work with complementary resources from multiple software tools in the course of a project because different tools have different strengths and capabilities. For example, one tool may have a good model editing interface, another tool may provide unparalleled facilities for analysis, yet another may implement an advanced simulation capability but lack a good graphical interface, etc. Working with multiple tools today generally requires having to re-encode the model in each tool, a time-consuming and error-prone process.
• Models published in peer-reviewed journals are sometimes accompanied by instructions for obtaining the model definitions in electronic form. However, because each author may use a different modelling environment (and model representation language), such definitions are often not straightforward to examine, test and reuse. If a researcher wishes to use a published model, the researcher typically must manually transcribe the model into a different format.
• When simulators are no longer supported, models developed in those systems can become stranded and unusable. This has already happened on a number of occasions, with a resulting loss of usable models to the community. Continued innovation and development of new tools will only aggravate this problem unless the issue of standard formats is addressed.

We developed the Systems Biology Markup Language (SBML) in an effort to address these problems. SBML is a format for representing computational models in a way that can be used by different software systems to communicate and exchange those models [32, 33]. By supporting SBML as an input and output format, different software tools can all operate on an identical representation of a model, removing opportunities for errors in translation and assuring a common starting point for analyses and simulations.

The SBML project is not an attempt to define a universal language for representing quantitative models; the fluid and

M. Hucka, B.J. Bornstein, B.E. Shapiro, B.L. Kovitz and J.C. Doyle are with Control and Dynamical Systems, Mail Code 107-81, California Institute of Technology, Pasadena, CA 91125, USA, email: mhucka@caltech.edu

A. Finney, S.M. Keating, J. Matthews and M.J. Schilstra are with the STRS, University of Hertfordshire, Hatfield, AL10 9AB, UK

B.J. Bornstein and B.E. Shapiro are with the Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA 91109, USA

A. Funahashi and H. Kitano are at the Kitano Symbiotic Systems Project, JST ERATO-SORST, 6-31-15 Jingumae, M-31 Suite 6A, Shibuyaku, Tokyo 150-0001, Japan

H. Kitano is also at the Systems Biology Institute, 6-31-15 Jingumae, M-31 Suite 6A, Shibuyaku, Tokyo 150-0001, Japan and the Sony Computer Science Laboratories, Inc., 3-14-15 Higashi-Gotanda, Shinagawa, Tokyo 141-0022, Japan

rapidly evolving views of biological function, and the vigorous rate at which new computational techniques and individual tools are being developed today, are incompatible with a one-size-fits-all idea of a universal language. A more realistic alternative is to acknowledge the diversity of approaches and methods being explored by different software tool developers, and seek a common intermediate format — a lingua franca — enabling communication of the most essential aspects of the models.

As a practical consequence of how SBML is being developed, it reflects how theoreticians and software developers conceptualise and structure their computational models of biochemical reaction networks. Thus, in a very concrete fashion, SBML represents a consensus view of how these computational models are understood today, and the form the language is taking in its continued evolution offers a glimpse of where the field is headed in the near future.

2 The genesis and general form of SBML

SBML is a machine-readable model definition language based upon XML, the eXtensible Markup Language [34, 35], which in turn is a simple and portable text-based substrate that has been gaining widespread acceptance in computational biology and bioinformatics [36, 37]. The need for a language like SBML was manifest during the first *Workshop on Software Platforms for Systems Biology*, held at the California Institute of Technology in early 2000. The two or three dozen attendees at the time represented less than a dozen software projects: BioSpice [38], Cellerator [39–43], DBsolve [44–47], E-Cell [48–60], Gepasi [61–67], Jarnac [68–70], StochSim [71–77], and Virtual Cell [78–93]. Yet even within this small community, it proved impossible to share models without having to re-encode them anew in each software tool. This needless impediment to collaboration directly inspired the SBML project. Since then, SBML has continued to evolve and grow in popularity, to the point that at the time of this writing, it is used by over fifty software packages worldwide (discussed further below) and has influenced the development of standards in related areas [94]. Moreover, it is the standard model definition language used by several consortia, notably the US-based BioSPICE project [17] and the International E. coli Alliance [9].

2.1 SBML Levels

SBML is being developed in levels, where each higher level adds richness to the model definitions that can be represented by the language. By delimiting sets of features at incremental stages, the SBML development process provides software authors with stable standards and the community can gain experience with the language definitions before new features are introduced. Two levels have been defined so far, named (appropriately enough) Level 1 and Level 2. Level 1 is simpler than Level 2, but also has less representational power. These separate levels of SBML are intended to coexist; SBML Level 2 does not render SBML Level 1 obsolete. Software tools that do not need or cannot support higher levels can go on using lower SBML levels; tools that *can* read higher levels are assured of also being able to interpret models defined in the lower levels. The open-source software infrastructure we are developing around SBML (see Section 3.2) allows developers to support both SBML Levels 1 and 2 in their software with a minimum amount of effort.

2.2 Brief summary of the structure and features of SBML Level 2

SBML can encode models consisting of biochemical entities (species) linked by reactions to form biochemical networks. An important principle in SBML is that models are decomposed into explicitly-labelled constituent elements, the set of which resembles a verbose rendition of chemical reaction equations; the representation deliberately does *not* cast the model directly into a set of differential equations or other specific interpretations of the model. This decomposition makes it easier for a software tool to interpret the model and translate the SBML form into whatever internal form the tool actually uses.

Level 2 is the highest level of SBML currently defined; it represents an incremental evolution of the language [32] resulting from the practical experiences of many users and developers working with Level 1 since its introduction in the year 2001 [33, 95]. A definition of a model in SBML Level 2 consists of lists of one or more of the following components: *compartment:* a container of finite volume for well-stirred substances where reactions take place; *species:* a pool of a chemical substance located in a specific compartment (a species represents the concentration or amount of a chemical substance and not a single molecule); *reaction:* a statement describing some transformation, transport or binding process that can change one or more species (each reaction is characterised by the stoichiometry of its products and reactants and optionally by a rate equation); *parameter:* a quantity that has a symbolic name; *unit definition:* a name for a unit used in the expression of quantities in a model; *rule:* a mathematical expression that is added to the model equations constructed from the set of reactions (rules can be used to set parameter values, establish constraints between quantities, etc.); *function:* a named mathematical function that can be used in place of repeated expressions in rate equations and other formulas; and *event:* a set of mathematical formulas evaluated at specified moments in the time evolution of the system. This simple formalism allows a wide range of biological phenomena to be modelled, including cell signalling, metabolism, gene regulation, and more. Significant flexibility and power comes from the ability to define arbitrary formulae for the rates of change of variables as well as the ability to express other constraints mathematically.

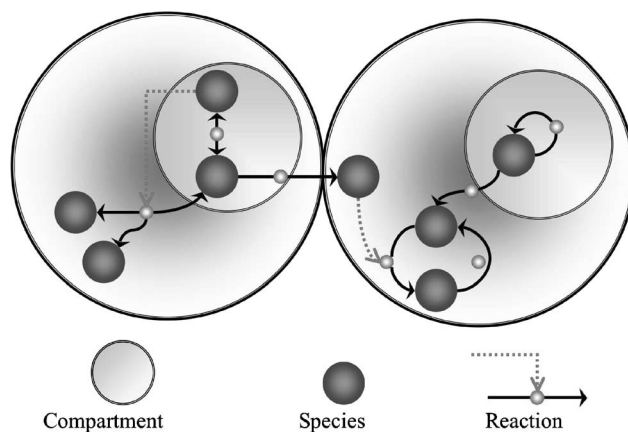Figure 1 provides a conceptual diagram of an example model featuring four compartments, multiple chemical

Fig. 1 *Schematic illustration of the concepts in a simple SBML model*

species, and multiple reactions. The chemical species are drawn here as single circles; each is meant to symbolise the concentration of a different chemical and not only a single molecule (though one could represent low-molecular count models in SBML just as easily). The diagram illustrates the notion of reactions that transport chemical species between compartments as well as species that act as modifiers in reactions.

Some additional noteworthy features of SBML Level 2 include: the use of MathML [96], an XML-based mathematical formula language; support for a systematic way of including metadata ('data about data', such as information about authorship or links to online databases); and support for delay functions. The latter are useful for representing biological processes having a delayed response, but where the details of the processes and the actual delay mechanism are not relevant to the operation of the model.

2.3 Relationships to other efforts

Many XML-based formats have been proposed for representing data and models in biology, including the architecture for genomic annotation, visualization and exchange (AGAVE) [97], BIOpolymer Markup Language (BIOML) [98], Bioinformatics Sequence Markup Language (BSML) [99], Chemical Markup Language (CML) [100], Microarray Gene Expression Markup Language (MAGE-ML) [101], Multiple Sequence Alignments Markup Language (MSAML) [102], Proteomics Experiment Markup Language (PEML) [103], Protein Markup Language (ProML) [104, 105], Protein Extensible Markup Language (PROXIML) [106], the Proteomics Standards Initiative's Molecular Interaction (PSI MI) format [94], and Ribonucleic Acid Markup Language (RiboML) [107]. However, we know of only two XML-based formats that are suitable for representing compartmental reaction network models with sufficient mathematical depth that the descriptions can be used as direct input to simulation software. The two are SBML and CellML [108–113].

CellML is built around an approach of composing systems of equations by linking together the variables in those equations; this is augmented by features for declaring biochemical reactions explicitly, as well as encapsulating arbitrary components into modules. Its focus is on a component-based architecture to facilitate reuse of models and parts of models, and the mathematical description of models. By contrast, SBML provides constructs that are more similar to the internal data objects used in many contemporary simulation/analysis software packages specialised for biochemical networks.

These differences notwithstanding, the SBML and CellML efforts share much in common and represent somewhat different approaches to solving the same general problems. They were initially developed independently, but the primary developers of both languages are actively engaged in exchanges of ideas and are seeking ways of making the languages more interoperable. SBML Level 2 borrows a number of approaches from CellML, making the formats that much easier to translate between each other.

3 Software for systems biology: a survey of SBML-compatible software

In this Section, we discuss software tools that support the SBML format and are available today. We discuss software developed by others as well as tools developed by our own group.

3.1 Tools for computational modelling in systems biology

Figure 2 is a list of specialised software tools providing facilities for computational modelling in biology. The list is limited to self-contained modelling tools that support SBML and have been publicly released (either freely or as commercial products) at the time of this writing; not listed are libraries (e.g. libSBML, discussed in the next Section), conversion utilities (e.g. KEGG2SBML, also discussed in the next Section), software packages that support SBML but are not generally available (e.g. Modesto [114, 115], pathSCOUT™ [116, 117]), and packages that do not yet support SBML but whose authors expressed intentions to support it in the future (e.g. Karyote™ [118, 119], Kinsolver [120, 121], MOMA [122, 123], ProMoT/DIVA [124–126]). For each package in Fig. 2, we also indicate its characteristics along many dimensions.

The column labelled 'Type' in Fig. 2 indicates whether the software in question is a stand-alone application (one that can be installed and run locally on a computer), or whether it is web-based, offering a service located on the Internet which users access remotely using a web browser. For those tools that can be used in either mode, this column indicates the way it is most commonly used.

The column labelled 'User Interface' indicates the dominant type of interface provided by the tool. The meanings of the categories under 'User Interface' are as follows:

• *Diagrammatic:* enables users to express models visually by placing or drawing elements, structures and relationships on a digital canvas. Often this takes the form of a graph resembling the block-and-arrow diagrams commonly presented by biologists as depictions of metabolic or signalling pathways. Additional quantitative information about the model is usually obtained from the user using a small number of fill-in-the-blank forms.
• *Spreadsheet:* provides a multicolumn grid interface reminiscent of spreadsheet programs offered in contemporary office productivity software suites. Information about reactions, species and compartments typically are entered in separate spreadsheet areas, each having separate columns for different characteristics of the elements being entered.
• *Forms-based:* prompts the user for information about a model using fill-in-the-blank forms and dialog boxes and pull-down menus. (Some tools take the information so gathered and display the resulting model using a diagram or a spreadsheet view but do not allow the user to *edit* the model directly using the diagram or spreadsheet; these are also listed as providing a forms interface rather than the other kinds of interfaces.)
• *Text-based:* enables users to define models using a formalised textual language and notation meant to be read and written by a human (i.e. not SBML). Some tools provide an editing interface for writing models in this language; others only read files created by external programs. Some of these languages mix constructs for defining models with directives for controlling simulations or other actions on the model.

All tools with these user interfaces translate the model from the input format (diagrammatic, spreadsheet, forms, text) into another format. For tools that directly incorporate simulation and analysis capabilities, this other format may be the program's internal data structures. For other tools, this format may be a custom format provided in addition to SBML. However, all the tools listed provide the ability to convert the model definitions into SBML and/or read

Syst. Biol., Vol. 1, No. 1, June 2004

43

System	Ref.	Type		User Interface				Mathematical Framework						DB		Avail.		SBML	
		Standalone	Web-based	Diagrammatic	Spreadsheet	Forms-based	Text-based	Logical	ODE	DAE	PDE	Hybrid	Stochastic	Integral	External	Free	Commercial	Level 1	Level 2
BALSA	[177]	■		■												■		■	
BASIS	[13, 178]		■			■			■				■	■		■			■
BioCharon	[179-181]	■					■			■		■				■			■
Bio Sketch Pad	[182]	■		■												■			■
BioSpreadsheet	[183]	■			■											■			■
BioUML	[184, 185]	■		■					■						■	■			■
BSTLab	[186]	■					■		■							■		■	
CADLIVE	[187, 188]	■		■												■			■
CellDesigner	[158-160]	■		■												■			■
Cellerator	[39-43]	■					■		■							■		■	
Cellware	[189, 190]	■		■					■				■			■		■	
Cytoscape	[191, 192]	■		☆												■		■	
DBsolve	[44-47]	■				■			■						■	■			■
Dizzy	[193]	■					■			■			■			■		■	
E-Cell 3	[48-60]	■		■		■	■			■		■	■		■	■		■	■
ecellJ	[194]	■				■										■		■	
ESS	[195, 196]	■					■						■			■			■
Gepasi	[61-67]	■				■			■							■		■	
Jarnac	[68-70]	■					■			■		■	■			■		■	■
JDesigner	[70, 197]	■		■												■		■	■
JigCell	[198-203]	■			■					■		■				■			■
JSIM	[204, 205]	■					■				■					■		■	
JWS	[206-209]		■	■		■				■				■		■		■	
MathSBML	[152, 153]	■					■			■		■				■		■	■
Monod	[210, 211]		■			■								■		■			■
NetBuilder	[212, 213]	■		■				■						■		■		■	
PathArt	[214]		■	☆										■			■	■	
PathwayBuilder	[215]	■		■										■		■			■
SCIpath	[216, 217]	■		■												■		■	■
SigPath	[218]		■			■								■		■		■	■
SigTran	[219]	■				■				■			■			■		■	
Simpathica	[220-223]	■				■		■	■										■
SimWiz	[224]	■		☆												■			■
StochSim	[71-77]	■				■							■			■		■	■
STOCKS	[225, 226]	■				■							■			■			■
Trelis	[227]	■				■							■			■		■	
Virtual Cell	[78-93]		■	■		■	■		■		■			■		■		■	■
VLX Suite	[228, 229]	■		■		■	■		■			■		■	■		■	■	■
WinSCAMP	[230-232]	■					■			■		■				■		■	

Fig. 2 *A sampling of software tools available today for modelling in systems biology. A square in a column indicates that a given tool possesses the feature noted in that column. An asterisk in the Diagrammatic User Interface column indicates that the given tool possesses this capability primarily for displaying information to the user, not for user input. See text for further explanations*

models in SBML. Finally, some of the tools are output-only model visualisation systems; these are indicated by an asterisk rather than a filled square in the 'User Interface' columns.

For those tools providing simulation and other analysis capabilities, the column labelled 'Mathematical Framework' in Fig. 2 indicates the model representation framework(s) available. The meanings of the categories under 'Mathematical Framework' are as follows:

- *Logical:* converts the model description into a Boolean or extended logical representation [127, 128]. Certain classes of models, such as abstract models of genomic regulatory networks, are more conveniently cast into this form than into, for example, differential equations.
- *Ordinary differential equations (ODEs):* converts the model description into a system of ordinary differential equations. This commonly involves one differential equation for each chemical species in the model. ODE representations are a popular framework, but biological models often include algebraic constraints that require the use of DAEs representations (see next).
- *Differential-algebraic equations (DAEs):* converts the model into a system of ordinary differential equations with algebraic constraints. An example is when a model imposes constraints on species concentrations. The DAE framework subsumes the ODE framework, but can support more constructs that modellers often want to express and thus is a better match for modellers' needs. Full DAE solvers are more difficult to implement than ODE solvers [129]; for this reason, many simulation tools support limited forms of DAEs.
- *Partial differential equations (PDEs):* converts the model into a system of partial differential equations. These arise when there is more than one independent variable in the system, as when modelling spatial diffusion (leading to a model having not only time but also spatial variables as independent variables). PDE solvers are significantly more difficult to implement and use properly than either ODE or DAE solvers, which helps explain why so few software tools today use a PDE framework. Note that SBML does not currently have a means to represent PDE-level models or diffusion terms.
- *Hybrid:* converts the model to a (continuous) differential equation framework and also supports time-dependent discontinuous events; these discontinuities can cause abrupt changes in the system of equations and the behaviour of the system. Hybrid modelling frameworks are necessary for properly handling such things as cell cycle models.
- *Stochastic:* casts the model as a system consisting of a set of discrete quantities (molecules of chemical species) and associated probabilities for interactions (the reactions). Most (but not all) such software systems use the stochastic simulation algorithm (SSA) devised by Gillespie [130] or the Gibson-Bruck variant of SSA [131], or (more rarely) the Tau-Leap algorithm [132]. Unlike the various differential equation frameworks described above, a stochastic framework does not approximate the model as a continuous deterministic system, but treats it as the underlying biochemical reality really is random discrete processes. However, this increased accuracy comes at a high cost: each individual chemical entity is modelled as a stochastic process and as a result, simulations are extremely demanding of computational resources.

There are many more modelling frameworks in use [128] than these six, but these cover the capabilities of the SBML-compatible software tools available today. Some of the tools support multiple mathematical frameworks; this is indicated by circles in multiple columns for those tools.

The 'DB' columns in Fig. 2 indicate whether a given software tool provides database functionality. A mark in the 'Integral' databases column means the tool possesses an integral database system; i.e. the database stores models and model components (e.g. species) in a form more organised than simply a collection of files, and the systems sometimes offer a means to share this database between multiple users. A mark in the 'External' DB column indicates the tool provides access to third-party external repositories of data, models or other information. Note that a tool may feature both an integral database and provide external database access functionality.

The column labelled 'Availability' indicates the availability of the tool: either 'Free', meaning the tool is available at no cost for personal use (but possibly at cost for other users), or 'Commercial', meaning the tool is a commercial product that must be purchased. Many of the free tools are also open-source.

Finally, the 'SBML' column indicates the SBML level supported by the tool: either Level 1 only, or Level 2 only, or both. (Although tools that can read SBML Level 2 would certainly be able to interpret Level 1 models as well, not all tools were developed using parsers that provide support for both levels. This is likely to change in the future as libSBML, discussed below, and other parser tools gain wider use.)

The tools listed in the Figure possess other features and capabilities than are shown here, further distinguishing them along other dimensions. Most are also under active development and are gaining new features on a regular basis. Nevertheless, the Figure shows something of the wide range of feature combinations offered in today's software packages for computational modelling in systems biology.

3.2 SBML software infrastructure

To make it easier for software developers and users to work with SBML, and more generally to promote the language's use as a common exchange format, our group as well as other groups have released and continue to develop a number of open-source SBML software tools. They fall under the following categories:

1. software libraries for reading and writing files and data streams containing SBML content;
2. interface packages supporting the use of SBML in general-purpose mathematical environments (specifically, Mathematica and MATLAB);
3. conversion tools for translating models in other formats into SBML format; and
4. online web-based facilities for performing a variety of tasks on SBML files.

We briefly describe this software below.

All software developed by our group is freely and openly available under the terms of the GNU lesser general public licence (LGPL) [133]. The use of the LGPL ensures that the software is available at no cost indefinitely, while simultaneously allowing developers the freedom to release closed-source modules (e.g. commercial packages) that work with the software.

3.2.1 Software libraries for programming with SBML: While it is a simple matter for a software package to use a general-purpose XML parsing library to read and write SBML-formatted files and data streams, it is considerably more convenient for software developers to

have at their disposal a higher-level interface tailored specifically to the kinds of data object structures present in SBML. A specialised interface also offers the opportunity to provide powerful capabilities for such things as automated model consistency checking.

The most recent and advanced library for SBML is *libSBML*. It provides an application programming interface (API) for reading, writing and manipulating data in SBML format. By using libSBML, programmers can save themselves the work of implementing their own parsing, manipulation and validation software. Developers can embed the library in their applications, and by calling on the library's functions, the application can gain support for SBML Levels 1 and 2. LibSBML is written in ISO C and C++ and currently provides language bindings for C, C++, Java, Python and MATLAB, with support for Perl and other languages planned for the future. We distribute the package in both source-code form and as precompiled dynamic libraries for the Microsoft Windows, Linux and Apple MacOS X operating systems; they are available from the 'sbml' project on SourceForge.net [134], the world's largest open-source software repository and project hosting service, as well as from the SBML project web site [135].

LibSBML provides both syntactic validation and consistency checking of SBML. Syntactic validation involves verifying that the SBML input is well-formed, and for example that data values are of the correct types. This is achieved by leveraging the power of an XML parser library that can check SBML input against an XML Schema [136–138] for SBML Level 1 and Level 2. The XML parser is of the 'Simple API for XML' (SAX) variety [139]. Consistency checking involves verifying the contents of an SBML model for self-consistency and adherence to the SBML specifications. The tests are implemented as rules within libSBML; the library reports back validation failures (including the location in the input and nature of the failure) to the calling application via the libSBML API.

At the API level, the library provides the same interface to data structures independently of whether the model originated in SBML Level 1 or 2. Thus it is possible to use libSBML to enable an application to support both Level 1 and 2 by accessing a single API. The library currently also offers the ability to translate SBML Level 1 models to SBML Level 2.

Prior to the development of libSBML, we had developed parser libraries for SBML Level 1 in Java as well as Microsoft Windows DLL form. These libraries were distributed as part of the systems biology workbench [70, 140–144] and are still available separately [145], but they have been superseded by libSBML and we discourage their use.

Other groups besides ours have also developed parser libraries for SBML. Two such libraries that are freely available as separate components are a Java library for SBML Level 2 created for JigCell [146], and a library for Common Lisp created as part of the Simpathica project [147].

3.2.2 Interfaces to popular general-purpose mathematical environments:
Special-purpose software tools for computational systems biology offer distinct advantages and capabilities, but many users nevertheless prefer to work with general-purpose mathematical environments. To support the use of SBML in two popular commercial systems, Mathematica [148, 149] and MATLAB [150, 151], we have developed separate software packages for these environments.

MathSBML [152, 153] is an open-source package that allows investigators to explore SBML models using the full range of features available in Mathematica, an exhaustive mathematical environment widely used by biological modellers and available in many academic and commercial environments. MathSBML provides full interoperability with this environment as well as a candidate reference implementation for nearly all features in SBML Levels 1 and 2, including the solution of differential-algebraic equations and discontinuous events. MathSBML has a complete API for model manipulation, and includes simple function points for model-based hybrid simulation, model exploration, plotting, file importing, and model exporting. The main feature is the ability to convert SBML models into a Mathematica data structure that is compatible with standard Mathematica functions; it also contains an event driver that allows modellers to run simulations from event-driven models, and plot the results, across time spans containing multiple events. Using this API, users can modify a pre-existing model or build a new model from scratch, iteratively running simulations and modifying the model before writing a new SBML file to disk. Additional features include the ability to display models in a tabular, human-readable form (either within a Mathematica notebook or via an HTML file), and export to other formats such as FORTRAN and XPP.

The *SBMLToolbox* is a MATLAB toolbox that provides facilities for reading, writing and manipulating SBML models within the MATLAB environment. It enables users to import an SBML model and automatically construct a MATLAB object structure that reflects the components of the model, enabling it to be analysed by a user's own MATLAB routines as well as publicly available MATLAB toolboxes. The SBMLToolbox provides additional functionality for working with SBML models within MATLAB, including validation of the resulting data structures, saving and loading imported models to/from data files, a simple forms-based model editor, and facilities for converting elements of a model into MATLAB's symbolic form. SBMLToolbox uses libSBML as its underlying parser library, and is currently available for the Linux, Apple Macintosh, and Microsoft Windows operating systems.

3.2.3 Conversion tools:
Data of use to computational modellers including, whole models, already exist in a number of forms besides SBML. In order to help modellers make use of these resources, we and others have been developing stand-alone conversion software capable of transforming models in a number of non-SBML formats into SBML. Available today are converters for the Kyoto Encyclopaedia of Genes and Genomes (KEGG); [154, 155]), CellML and BioCyc [156, 157], with others in development.

Identification of gene-regulatory logic and biochemical networks is a major challenge of systems biology. Several attempts are underway to create large-scale, comprehensive databases of gene-regulatory and biochemical networks. Making the contents of these databases available in SBML format will be useful for the following reasons: (1) it will enable researchers to apply many SBML-aware software tools to the networks in those databases, and (2) the feedback from developing the translation tools will provide valuable information for the continued evolution of SBML. As a first attempt at writing database translation tools, we have decided to convert the KEGG database. KEGG contains more than 13 000 metabolic pathways and 960 regulatory pathways for more than 150 organisms. We have implemented a converter called *KEGG2SBML* that converts KEGG metabolic pathway database files into SBML Level 1 and Level 2 files. Further, KEGG2SBML can parse diagram

46

Syst. Biol., Vol. 1, No. 1, June 2004

layout information from KEGG and add it to SBML as annotations; the result can be used in CellDesigner, a process network diagram editor we have also developed [158–160]. Using KEGG2SBML, we have successfully converted 10 869 KEGG metabolic pathways (out of 13 333) into SBML Level 1 and Level 2, a conversion rate of 82%.

As discussed in Section 2.3, CellML is another XML-based format for representing computational models in systems biology. We have developed conversion software, consisting of four XSLT scripts [161, 162], which can convert a CellML 1.1 model into an SBML Level 2 model. The XSLT scripts are applied consecutively. The first script performs a rough translation of the CellML document into SBML. This rough model contains a single compartment. (CellML has no equivalent of SBML's explicit compartments.) CellML unit definitions are also expanded into their constituent SI units because SBML's more limited unit system only allows unit definitions to be directly composed from SI units. The first XSLT script creates SBML annotations containing information on the names of CellML 'delta' variables and connections which are used to further simplify the model in later stages. Otherwise, the first XSLT transformation maps CellML elements to SBML elements as one would expect: CellML reactions to SBML reactions, CellML variables occurring as species in reactions to SBML species, and all other CellML variables to SBML parameter elements. The second XSLT script merges units of the same type, simplifies the names of parameter and species names by following CellML connections, computes species substance units and species spatial size units, and eliminates CellML delta variables and the equations in which they occur. The third XLST script removes duplicate species and parameters created by the name simplification in the previous style sheet, performs further name simplification, and sorts the SBML assignment rules into a valid order while eliminating any cyclic dependencies among the assignment rules. These cyclic dependencies are eliminated by rewriting selected assignment rules as algebraic rules. The fourth XLST script checks the validity of the SBML model and removes any annotations created by the conversion process.

We have applied the CellML-to-SBML conversion system to the models in the CellML model repository [163]. Of the 261 models in that repository, the converter was able to convert 243 models (93%) into valid SBML Level 2 models. Six more models were added to the list of successful conversions after manually changing new base units into equivalent units that are legitimate in SBML. The remaining 12 models either contained PDEs (two models) or unusual combinations of delta variables (ten models), which cannot be mapped to SBML at this time.

Finally, besides KEGG2SBML and the CellML-to-SBML converter, there is also a converter for the Biocyc pathway/genome database [156, 157] developed by another group. This program, implemented by Zucker [164] in Common Lisp, can generate SBML Level 2 models from the Biocyc database.

3.2.4 *Online web-based facilities:* We have made a number of simple online facilities available on the SBML project web site [135]. Using any web browser, it is possible for users to upload an SBML model and then perform the following operations on it: check the syntax of the uploaded model, view the model in diagrammatic form, and convert a Level 1 model to Level 2. We plan on extending these facilities before the end of 2004 to enable the conversion of models from SBML Level 2 to Level 1,

CellML to SBML, SBML to Fortran, SBML to XPP and SBML to HTML. The majority of the current and planned facilities are currently or will be provided by server-side instances of the other software tools developed as part of the SBML project.

3.3 SBML model repository

We have implemented a simple model repository on our project web site [144]. It contains sample models in SBML format; nearly all of them are based on models published in the literature. Among these sample models are: cell cycle models from John Tyson's group [165, 166], a model of IP3-sensitive Calcium channel [167], models of MAPK cascades [168, 169], and others. We anticipate placing the SBML models translated from the CellML repository (see Section 3.2.3) in the SBML model repository in the second half of 2004. At the same time, we will make all the models available in a variety of presentation formats, including diagrams.

4 Planning for SBML Level 3

As noted in Section 2, SBML from the beginning has been largely driven by the practical needs of researchers interested in exchanging quantitative computational models between different software tools, databases and other resources. The language reflects this, and in some respects exhibits the results of pragmatic choices more than elegant, top-down design. SBML Level 2 benefited from two years of experience in the use of SBML Level 1 by many modellers and software developers, and distills more effectively the fundamental needs of the biological network simulation community. It represents, in a concrete way, the consensus of a large segment of this community about the intersection of features that should be possessed by a lingua franca for communicating models between software tools such as those in Fig. 2.

SBML's popularity has led to the formation of an active community of researchers and software developers who are now working together to push SBML in new directions. As a language that is an intersection rather than a union of features needed by all tools, SBML currently cannot support all the representational capabilities that all software systems offer to users. Some tools offer features that have no explicit equivalent in SBML Level 2, and those tools currently can only store those features as annotations in an SBML model. But in many cases those features could potentially be used by more than one tool, and thus it would be appropriate to have some representation for them in SBML. Using Level 2 as a starting point, the SBML community has been developing proposals and prototype implementations of many new capabilities that will become part of SBML Level 3. The main current areas of interest are:

• *Diagram layout:* enabling the inclusion of diagrammatic renditions of a model.
• *Model composition:* allowing models to be constructed from instances of submodels.
• *Multicomponent species:* allowing species to be composed from instances of species types, enabling such things as the representation of complexes of phosphorylated proteins and generalised reactions acting on them.
• *Arrays:* allowing models to contain indexed collections of objects of the same type.
• *Spatial features:* allowing the representation of the geometric features of compartments, the diffusion rates of species and the spatial distribution of model parameters and boundary conditions.

Syst. Biol., Vol. 1, No. 1, June 2004

47

- *Controlled vocabularies:* enabling the annotation of SBML models with terms from controlled vocabularies.
- *Constraints:* enabling the definition of constraints on model variables.

In the Subsections that follow, we discuss some of these capabilities in more detail, focusing on those that are likely to be developed by the end of 2004. We begin by outlining our expected approach for extending the language with these features.

4.1 Modularising the representation

It is unreasonable to expect a tool to support every feature planned for Level 3 in order to be called Level 3 compatible. One of the challenges for SBML Level 3 will be to design a modular representation. The idea is to provide each model with explicit information about which capabilities are necessary to interpret it correctly, so that tools encountering the model may reject it gracefully if they do not possess the necessary facilities. For reasons of efficiency and correctness, an explicit indication is preferable to requiring tools to read and interpret the entire model and inferring the capabilities needed.

We anticipate that Level 3 will take the form of a core, consisting of minimal extensions to Level 2, and a set of *Level 3 modules*, each encapsulating the definition of one of the major features listed above. One of the extensions making up the *Level 3 core* will be explicit feature indicators, such that each of the modules has a corresponding feature tag which will appear in a list at the beginning of the model definition. The presence of a feature tag will signal to software tools reading the model that the model uses that particular feature. The software tool may then make a decision about whether it can handle the model or whether it should alert the user to a problem.

4.2 Diagram Layout

As Fig. 2 shows, graphical interfaces are popular among existing software tools. Users find diagrammatic representations of biochemical reaction networks intuitive and appealing, and many tools provide a means to edit models directly through a visual, diagrammatic interface. Users naturally would like the diagrams to be preserved with their model definitions. However, there is no specified approach for recording diagram layouts in SBML Level 2 and thus, the tools must store the information as annotations. This means that different tools cannot read each other's diagram representations (unless they happen to implement support for exactly the same annotations, and currently, none do).

One of the most active areas of interest in the SBML community has been developing a common format for incorporating diagram layout information in SBML. The current leading proposal in this area supports the inclusion of multiple, different layouts in a model definition. These diagram layouts describe the positions of diagram objects, or *glyphs*, representing species, reactions and compartments. Information about colour, line styles and fonts is not represented by the proposed data structures. Objects in a model, for example species, can have any number of glyphs occurring in the same diagram, to enable the user to avoid line crossings by placing representations of the objects in multiple locations in the model. So that model diagrams can hide particular details of a given model, model objects do not have to have any corresponding objects in a given diagram. The proposed data structures are closely connected to existing SBML structures, enabling the proposed scheme to be used by graphical model editing tools.

4.3 Composition

Biologists often describe and analyse biological systems in a hierarchical fashion, both in structural terms (a higher organism might be composed from a set of organs, which are in turn composed of tissues, which are in turn composed of cells [11]) and in functional terms (certain network motifs repeat themselves throughout biochemical networks in a cell [170–174]). Curiously, it is the case that most published computational models of biological networks do not make use of hierarchical composition, perhaps because the approach is more commonly taught in engineering disciplines, or because most software tools have not had explicit support for developing models in this fashion. However, this is clearly beginning to change, both because of the engineering influence on the field of systems biology and the development of software tools providing explicit support for model composition (e.g. E-Cell, NetBuilder, ProMoT/DIVA, VLX Suite).

Supporting model composition in SBML would have several benefits. First, it would allow models to be encoded in a form closer to the natural decomposition biologists make in terms of structure and function. Second, it would facilitate the development of large models, especially those developed by multiple individuals. Third (and related), as the field's ability to model the details of biological systems grows, there will come a time when libraries of vetted, standard model components become increasingly common, and reusing components in other models will be most naturally done using a model composition approach. For all of these reasons, a segment of the SBML community has been developing proposals for model composition extension for SBML Level 3.

The proposals put forward so far have the following goals:

- Enabling copies of the same submodel to be used (instantiated) within an enclosing model. For example, a model cell could reuse multiple instances of the same mitochondrial submodel.
- Enable modellers to incorporate several alternative submodels for a given model instance, in which each alternative submodel could contain a representation at a different level of detail and/or use a representation that is appropriate for a particular type of simulation algorithm. Modellers could select which submodel to use at simulation time.
- Allow the creation of libraries of model components. This requires mechanisms for referencing submodels located externally to a given model.

The model composition facility must treat many details carefully. For example, SBML model structures have many attributes linking together the various components of a model, such as the parts of reactions or species localised within compartments. The model composition facility must allow these links to be references between components in different submodels, and between enclosing models and submodels, to support such things as reactions involving species located between submodel instances.

Many other issues arise, but the details are outside the scope of this article. We refer interested readers to the SBML web site [135] for the text of the proposals for model composition that have been made so far, and for information about the status of these proposals with respect to inclusion in SBML Level 3.

4.4 Multicomponent species

SBML Levels 1 and 2 can represent models in which the chemical species are treated as simple, indivisible

biochemical entities having only one possible state. This approach becomes untenable when modelling systems in which the species have many possible internal states or the species are composed from subcomponents [175]. An example of this is a protein that can be phosphorylated at multiple locations: the possible phosphorylation combinations lead to a combinatorial explosion of states of the protein. Although currently this can be represented in SBML Levels 1 and 2 by treating each state or combination of subcomponents as a separately-named chemical species, the approach is awkward and limited. To address this problem, another area of SBML Level 3 development has been a representation scheme in which the subcomponents of chemical species are the smallest logical entities, rather than whole species. The research task is to define a representation scheme that is flexible enough to represent all the relevant biochemical phenomena while remaining computationally feasible for simulation and analysis.

One of the approaches being pursued for this problem involves having a model optionally contain a set of species types. A given species type simply represents all biochemical entities with the same biochemical structure (that is, it has identical structure for the purposes of the model). In SBML Level 2, species represent a pool of entities of the same type located in a specific compartment. In this proposal, the type of a species structure is made explicit so that, for example, the ATP in two compartments can be identified as having the same type. A reaction can be generalised to occur in any compartment by referring to reactants, products and modifiers by species type rather than by compartment specific species.

A species type can optionally contain a set of instances of other species types which define the composition of the containing species type. A model can be described using such a system of hierarchically contained components; however, under this proposal, the species type instances optionally can be connected. That is, a type can describe a graph where the nodes are species type instances and the arcs are bonds. A species type has a set of binding site structures, each of which is a potential end-point for a bond. A bond is simply a pair of references to binding sites on species type instances. The phosphorylation states of proteins could be represented as the binding of instances of a phosphate species type to binding sites on those proteins. Just as the bond structures are optional in species type structures, SBML would not specify the level of decomposition. A protein could be described as a single indivisible object, or as a sequence of amino acids, depending on what is appropriate for a given model and preferred by the modeller.

Although the structures described above capture a significant amount of information that cannot be made explicit in SBML Level 2, they do not provide any facilities for representing reactions generalised to apply to classes of species types. With just these structures an accurate model would still have to contain an enumeration of all the species type structures that could occur in the modelled system. This could easily get unwieldy. To solve this problem, under this proposal, reactions would be generalised to apply to classes of species types. The complete set of species and species type structures would be implied by the reactions rather than fully enumerated. In this context, the species structures contained in an SBML document would be taken to define the initial state and boundary conditions of the system. The species type structures would define a set of types that enable the definition of reactions and species. Reactions could then be considered to be analogous to the rules in a graph transformation system (e.g. PROGRES [176]) where species are pools of chemical entities where each entity is a graph.

The reactions would be applied to the biochemical entities in the modelled system that match the reactions' reactants and construct new entities as defined by the reactions' products. A generalised reaction would be a template for manipulating graphs of biochemical entities and contain variable structures which would enable a reactant to match with species from a range of species types. The non-variable components of reactants would match equivalent structures in the set of chemical entities and the variable parts match any binding site. The products of a reaction would be formed by assembling their non-variable components and copying the components which matched reactant variables to binding sites associated with the same variable in the product structures.

4.5 Arrays

The goal of the SBML arrays extension is to enable model components to be created from indexed collections of objects (arrays), where a collection contains nearly identical objects. The proposed format will facilitate the exchange of models between tools that support arrays such as Cellerator and the MEG extender for Gepasi [66]. In the current leading proposal for the array capability in SBML Level 3, an array can be a collection of objects of the same type. The types of objects that can be in these collections are compartments, species, reactions, parameters and events. As each object in a collection has a specific index, it possible to refer to individual objects within an array. By using object reference structures, similar to those described in Section 4.3, SBML objects can refer to other objects within arrays. By combining the structures proposed for arrays and model composition, it should be possible to construct such things as models of tissues of almost identical cells, the tissue being an array of instances of a single cell model.

5 Discussion

Computational modelling is becoming crucial for making sense of the vast quantities of complex experimental data that are now being collected. The systems biology community needs agreed-upon information standards if models are to be shared, evaluated and developed cooperatively. SBML is an XML-based format for representing computational models in a way that can be used by different software systems to communicate and exchange those models. It is supported today by over fifty software tools worldwide and a vibrant community of modellers and software authors.

In support of SBML and its community, we continue to develop and make available software infrastructure, including programming libraries, conversion utilities, interface packages for commonly-used software environments, and easy-to-access Internet-accessible online tools. All of our software development follows the open-source tradition to maximise the accessibility and utility of the products.

The success of SBML has led to requests from the community for new features and continued evolution of the language. We view our role as organisers and editors in the development and evolution of SBML; the process is open and crucially dependent on the involvement of others in the computational modelling field. We invite interested individuals and groups to join the SBML Forum, the informal community of SBML users and developers, to participate in the process and help us improve SBML and its capacity for acting as a common exchange format for computational modelling software in systems biology. Information on this and other aspects of the SBML project is available on the project web site [135].

6 Acknowledgments

7 References

1 Chance, B.: 'The kinetics of the enzyme-substrate compound of peroxidase', *J. Biol. Chem.*, 1943, **151**, pp. 553–577

2 Chance, B.: 'Analogue and digital representations of enzyme kinetics', *J. Biol. Chem.*, 1960, **235**, (8), pp. 2440–2443

3 Chance, B., Brainerd, J.G., Cajori, F.A., and Millikan, G.A.: 'The kinetics of the enzyme-substrate compound of peroxidase and their relation to the Michaelis theory', *Science*, 1940, **92**, (2394), p. 455

4 Garfinkel, D., Garfinkel, L., Pring, M., Green, S.B., and Chance, B.: 'Computer applications to biochemical kinetics', *Annu. Rev. Biochem.*, 1970, **39**, pp. 473–498

5 Abbott, A.: 'Alliance of US labs plans to build map of cell signalling pathways', *Nature*, 1999, **402**, pp. 219–220

6 Butler, D.: 'Are you ready for the revolution?', *Nature*, 2001, **409**, pp. 757–759

7 Chong, L., and Ray, L.B.: 'Whole-istic biology', *Science*, 2002, **295**, (5560), p. 1661

8 Doyle, J.: 'Computational biology – beyond the spherical cow', *Nature*, 2001, **411**, (6834), pp. 151–152

9 Holden, C.: 'Alliance launched to model E. coli', *Science*, 2002, **297**, (5586), pp. 1459–1460

10 Hood, L.: 'Systems biology: new opportunities arising from genomics, proteomics and beyond', *Experimental Hematology*, 1998, **26**, (8), p. 681

11 Hunter, P.J., and Borg, T.K.: 'Integration from proteins to organs: the Physiome project', *Nat. Rev. Mol. Cell Biol.*, 2003, **4**, pp. 237–243

12 Ideker, T., Galitski, T., and Hood, L.: 'A New Approach to Decoding Life: Systems Biology', *Ann. Rev. Genomics Human Genet.*, 2001, **2**, pp. 343–372

13 Kirkwood, T.B.L., Boys, R.J., Gillespie, C.S., Proctor, C.J., Shanley, D.P., and Wilkinson, D.J.: 'Towards an e-biology of ageing: integrating theory and data', *Nat. Rev. Mol. Cell Biol.*, 2003, **4**, (3), pp. 243–249

14 Kitano, H. (Ed.): 'Foundations of systems biology' (MIT Press, Cambridge, Massachusetts, 2001)

15 Kitano, H.: 'Computational systems biology', *Nature*, 2002, **420**, pp. 206–210

16 Kitano, H.: 'Systems biology: a brief overview', *Science*, 2002, **295**, pp. 1662–1664

17 Kumar, S., and Feidler, J.C.: 'BioSPICE: a computational infrastructure for integrative biology', *OMICS*, 2003, **7**, (3), p. 225

18 Pennisi, E.: 'Systems biology: tracing life's circuitry', *Science*, 2003, **302**, (5651), pp. 1646–1649

19 Service, R.F.: 'Exploring the systems of life', *Science*, 1999, **284**, (5411), pp. 80–83

20 Butler, D.: 'Computing 2010: from black holes to biology', *Nature*, 1999, **402**, (6761), pp. C67–C70

21 Alm, E., and Arkin, A.P.: 'Biological networks', *Curr. Opin. Struct. Biol.*, 2003, **13**, (2), pp. 193–202

22 Arkin, A.: 'Synthetic cell biology', *Curr. Opin. Biotechnol.*, 2001, **12**, pp. 638–644

23 Bialek, W., and Botstein, D.: 'Introductory science and mathematics education for 21st-century biologists', *Science*, 2004, **303**, pp. 788–790

24 Fraser, S.E., and Harland, R.M.: 'The molecular metamorphosis of experimental embryology', *Cell*, 2000, **100**, (1), pp. 41–55

25 Gilman, A.G., Simon, M.I., Bourne, H.R., Harris, B.A., Long, R., Ross, E.M., Stull, J.T., Taussig, R., Arkin, A.P., Cobb, M.H., Cyster, J.G., Devreotes, P.N., Ferrell, J.E., Fruman, D., Gold, M., Weiss, A., Berridge, M.J., Cantley, L.C., Catterall, W.A., Coughlin, S.R., Olson, E.N., Smith, T.F., Brugge, J.S., Botstein, D., Dixon, J.E., Hunter, T., Lefkowitz, R.J., Pawson, A.J., Sternberg, P.W., Varmus, H., Subramaniam, S., Sinkovits, R.S., Li, J., Mock, D., Ning, Y.H., Saunders, B., Sternweis, P.C., Hilgemann, D., Scheuermann, R.H., Decamp, D., Hsueh, R., Lin, K.M., Ni, Y., Seaman, W.E., Simpson, P.C., O'connell, T.D., Roach, T., Choi, S., Eversole-Cire, P., Fraser, I., Mumby, M.C., Zhao, Y.M., Brekken, D., Shu, H.J., Meyer, T., Chandy, G., Do Heo, W., Liou, J., O'rourke, N., Verghese, M., Mumby, S.M., Han, H.P., Brown, H.A., Forrester, J.S., Ivanova, P., Milne, S.B., Casey, P.J., Harden, T.K., Doyle, J., Gray, M.L., Michnick, S., Schmidt, M.A., Toner, M., Tsien, R.Y., Natarajan, M., Ranganathan, R., and Sambrano, G.R.: 'Overview of the Alliance for Cellular Signaling', *Nature*, 2002, **420**, (6916), pp. 703–706

26 Hartwell, L.H., Hopfield, J.J., Leibler, S., and Murray, A.W.: 'From molecular to modular cell biology', *Nature*, 1999, **402**, (6761 Suppl. S), pp. C47–C52

27 Noble, D.: 'The rise of computational biology', *Nat. Rev. Mol. Cell Biol.*, 2002, **3**, (6), pp. 460–463

28 Nurse, P.: 'A long twentieth century of the cell cycle and beyond', *Cell*, 2000, **100**, (1), pp. 71–78

29 Tyson, J.J., Chen, K., and Novak, B.: 'Network dynamics and cell physiology', *Nat. Rev. Mol. Cell Biol.*, 2001, **2**, pp. 908–916

30 Zerhouni, E.: 'The NIH roadmap', *Science*, 2003, **302**, pp. 63–64

31 Bower, J.M., and Bolouri, H.: 'Introduction: understanding living systems', in Bower, J.M., and Bolouri, H. (Eds.): 'Computational Modeling of Genetic and Biochemical Networks' (MIT Press, Cambridge, Mass, 2001), pp. xiii–xx

32 Finney, A.M., and Hucka, M.: 'Systems biology markup language: Level 2 and beyond', *Biochem. Soc. Trans.*, 2003, **31**, pp. 1472–1473

33 Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J.C., Kitano, H., Arkin, A.P., Bornstein, B.J., Bray, D., Cornish-Bowden, A., Cuellar, A.A., Dronov, S., Gilles, E.D., Ginkel, M., Gor, V., Goryanin, I.I., Hedley, W.J., Hodgman, T.C., Hofmeyr, J.-H., Hunter, P.J., Juty, N.S., Kasberger, J.L., Kremling, A., Kummer, U., Le Novére, N., Loew, L.M., Lucio, D., Mendes, P., Minch, E., Mjolsness, E.D., Nakayama, Y., Nelson, M.R., Nielsen, P.F., Sakurada, T., Schaff, J.C., Shapiro, B.E., Shimizu, T.S., Spence, H.D., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., and Wang, J.: 'The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models', *Bioinformatics*, 2003, **19**, (4), pp. 524–531

34 Bray, T., Paoli, J., Sperberg-McQueen, C.M., and Maler, E.: 'Extensible markup language (XML) 1.0 (second edition), W3C Recommendation 6-October-2000', http://www.w3.org/TR/1998/REC-xml-19980210

35 Bosak, J., and Bray, T.: 'XML and the second-generation web', *Sci. Am.*, 1999

36 Augen, J.: 'Information technology to the rescue!', *Nat. Biotechnol.*, 2001, **19**, (Supplement), pp. BE39–BE40

37 Achard, F., Vaysseix, G., and Barillot, E.: 'XML, bioinformatics and data integration', *Bioinformatics*, 2001, **17**, (2), pp. 115–125

38 Arkin, A.P.: 'Simulac and Deduce', 2001, http://gobi.lbl.gov/~aparkin/Stuff/Software.html

39 Shapiro, B., Levchenko, A., and Mjolsness, E.: 'Cellerator: a computational technique for automatic model generation of signal transduction pathways', in *Smart Systems 2000* (The Institute for Advanced Interdisciplinary Research, The Atlas Building, 16821 Buccaneer Lane, Suite 206, Houston, TX 77058, USA, 2000)

40 Shapiro, B.E., Levchenko, A., Meyerowitz, E.M., Wold, B.J., and Mjolsness, E.D.: 'Cellerator: extending a computer algebra system to include biochemical arrows for signal transduction simulations', *Bioinformatics*, 2003, **19**, (5), pp. 677–678

41 Shapiro, B.E., Levchenko, A., and Mjolsness, E.: 'Automatic model generation for signal transduction with applications to MAP-kinase pathways', in Kitano, H. (Ed.): 'Found. Syst. Biol.' (MIT Press, 2001), pp. 145–161

42 Shapiro, B.E., Mjolsness, E., and Levchenko, A.: 'Cellerator', 2004, http://www-aig.jpl.nasa.gov/public/mls/cellerator/

43 Shapiro, B.E., and Mjolsness, E.D.: 'Developmental simulations with cellerator'. Presented at 2nd Int. Conf. on Systems Biology (ICSB), 2001, Omnipress Inc.

44 Goryanin, I.: 'DBsolve: software for metabolic, enzymatic and receptor-ligand binding simulation', 2001, http://websites.ntl.com/~igor.goryanin/

45 Goryanin, I., Hodgman, T.C., and Selkov, E.: 'Mathematical simulation and analysis of cellular metabolism and regulation', *Bioinformatics*, 1999, **15**, (9), pp. 749–758

50

Syst. Biol., Vol. 1, No. 1, June 2004

46 Hodgman, C., Goryanin, I., and Juty, N.: 'Reconstructing whole-cell models', *Drug Disc. Tod.*, 2001, **6**, (15), pp. S109–S112

47 Juty, N., Spence, H.D., Hotz, H.R., Tang, H., Goryanin, I., and Hodgman, T.C.: 'Simultaneous modelling of metabolic, genetic and product-interaction networks', *Brief. Bioinformatics*, 2001, **2**, (3), pp. 223–232

48 Hashimoto, K., Seno, S., Dhar, P.K., and Tomita, M.: 'Integrative modeling of gene expression and metabolism with E-Cell simulation system', *Artif. Life Robot.*, 2002, **6**, pp. 99–107

49 Miyoshi, F., Nakayama, Y., and Tomita, M.: 'E-Cell simulation system and its application to the modeling of circadian rhythm', *Seikagaku*, 2003, **75**, (1), pp. 5–16

50 Normile, D.: 'Complex systems: building working cells 'in silico'', *Science*, 1999, **284**, (5411), pp. 80–81

51 Sugimoto, M., Takahashi, K., Kitayama, T., Ito, D., and Tomita, M.: 'Distributed computing with E-Cell system'. Presented at 1st Int. Workshop on Life Science Grid, Kanazawa, Japan, 2004

52 Takahashi, K., Ishikawa, N., Sadamoto, Y., Sasamoto, H., Ohta, S., Shiozawa, A., Miyoshi, F., Naito, Y., Nakayama, Y., and Tomita, M.: 'E-Cell 2: multi-platform E-Cell simulation system', *Bioinformatics*, 2003, **19**, (13), pp. 1727–1729

53 Tomita, M.: 'Whole-cell simulation: a grand challenge of the 21st century', *Trends Biotechnol.*, 2001, **19**, (6), pp. 205–210

54 Tomita, M., Hashimoto, K., Takahashi, K., Matsuzaki, Y., Matsushima, R., Saito, K., Yugi, K., Miyoshi, F., Nakano, H., Tanida, S., Saito, Y., Kawase, A., Watanabe, N., Shimizu, T.S., and Nakayama, Y.: 'The E-Cell project: Towards integrative simulation of cellular processes', *New Gen. Comput.*, 2000, **18**, (1), pp. 1–12

55 Tomita, M., Hashimoto, K., Takahashi, K., Shimizu, T., Matsuzaki, Y., Miyoshi, F., Saito, K., Tanida, S., Yugi, K., Venter, J.C., and Hutchison, C.: 'E-Cell: Software environment for whole cell simulation', *Bioinformatics*, 1999, **15**, (1), pp. 72–84

56 Tomita, M., Nakayama, Y., Naito, Y., Shimizu, T., Hashimoto, K., Takahashi, K., Matsuzaki, Y., Yugi, K., Miyoshi, F., Saito, Y., Kuroki, A., Ishida, T., Iwata, T., Yoneda, M., Kita, M., Yamada, Y., Wang, E., Seno, S., Okayama, M., Kinoshita, A., Fujita, Y., Matsuo, R., Yanagihara, T., Watari, D., Ishinabe, S., and Miyamoto, S.: 'E-Cell', 2001, http://www.e-cell.org/

57 Takahashi, K., Kaizu, K., Hu, B., and Tomita, M.: 'A multi-algorithm, multi-timescale method for cell simulation', *Bioinformatics*, 2004, **20**, (4), pp. 538–546

58 Takahashi, K., Yugi, K., Hashimoto, K., Yamada, Y., Pickett, C.J.F., and Tomita, M.: 'Computational challenges in cell simulation: a software engineering approach', *IEEE Intell. Syst.*, 2002, **17**, (5), pp. 64–71

59 Triendl, R.: 'Computerized role models', *Naturejobs*, 2002, **417**, (6892), p. 7

60 Yugi, K., and Tomita, M.: 'A general computational model of mitochondrial metabolism in a whole organelle scale', *Bioinform. Adv. Acc.*, 2004, **1**, (1), pp. 1–2

61 Mendes, P.: 'Gepasi—a software package for modeling the dynamics, steady-states and control of biochemical and other systems', *Comput. Appl. Biosci.*, 1993, **9**, (5), pp. 563–571

62 Mendes, P.: 'Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3', *Trends Biol. Sci.*, 1997, **22**, pp. 361–363

63 Mendes, P.: 'New research software to simulate biochemical processes', 2000, http://www.vbi.vt.edu/pr/press_releases/press_20001218_news_new-software.htm

64 Mendes, P.: 'Gepasi 3.21', 2001, http://www.gepasi.org

65 Mendes, P., and Kell, D.B.: 'Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation', *Bioinformatics*, 1998, **14**, (10), pp. 869–883

66 Mendes, P., and Kell, D.B.: 'MEG (model extender for Gepasi): a program for the modelling of complex, heterogeneous, cellular systems', *Bioinformatics*, 2001, **17**, (3), pp. 288–289

67 Mendes, P., Sha, W., and Ye, K.: 'Artificial gene networks for objective comparison of analysis algorithms', *Bioinformatics*, 2003, **19**, (Suppl. 2), pp. ii122–ii129

68 Sauro, H.M.: 'Jarnac: a system for interactive metabolic analysis', in Hofmeyr, J.-H.S., Rohwer, J.M., and Snoep, J.L. (Eds.): 'Animating the cellular map'. Proc. 9th Int. Meeting on biothermokinetics (Stellenbosch University Press, 2000)

69 Sauro, H.M.: 'Jarnac', 2000, http://www.cds.caltech.edu/~hsauro

70 Sauro, H.M., Hucka, M., Finney, A., Wellock, C., Bolouri, H., Doyle, J., and Kitano, H.: 'Next generation simulation tools: the Systems Biology Workbench and BioSPICE integration', *OMICS*, 2003, **7**, (4), pp. 355–372

71 Bray, D., Firth, C., Le Novére, N., and Shimizu, T.: 'StochSim', 2001, http://info.anat.cam.ac.uk/groups/comp-cell/StochSim.html

72 Le Novére, N., and Shimizu, T.S.: 'StochSim: modelling of stochastic biomolecular processes', *Bioinformatics*, 2001, **17**, (6), pp. 575–576

73 Levin, M.D., Shimizu, T.S., and Bray, D.: 'Binding and diffusion of CheR molecules within a cluster of membrane receptors', *Biophys. J.*, 2002, **82**, (4), pp. 1809–1817

74 Morton-Firth, C.J., and Bray, D.: 'Predicting temporal fluctuations in an intracellular signalling pathway', *J. Theor. Biol.*, 1998, **192**, pp. 117–128

75 Morton-Firth, C.J., Shimizu, T.S., and Bray, D.: 'A free-energy-based stochastic simulation of the Tar receptor complex', *J. Mol. Biol.*, 1999, **286**, (4), pp. 1059–1074

76 Shimizu, T.S., Aksenov, S.V., and Bray, D.: 'A spatially extended stochastic model of the bacterial chemotaxis signalling pathway', *J. Mol. Biol.*, 2003, **329**, (2), pp. 291–309

77 Shimizu, T.S., and Bray, D.: 'Computational cell biology—the stochastic approach', in Kitano, H. (Ed.): 'Foundations of systems biology' (MIT Press, 2001)

78 Choi, Y.S., Resasco, D., Schaff, J., and Slepchenko, B.: 'Electrodiffusion of ions inside living cells', *IMA J. Appl. Math.*, 1999, **62**, (3), pp. 207–226

79 Fink, C.C., Slepchenko, B., Moraru, I.I., Schaff, J., Watras, J., and Loew, L.M.: 'Morphological control of inositol-1,4,5-trisphosphate-dependent signals', *J. Cell Biol.*, 1999, **147**, (5), pp. 929–935

80 Fink, C.C., Slepchenko, B., Moraru, I.I., Watras, J., Schaff, J.C., and Loew, L.M.: 'An image-based model of calcium waves in differentiated neuroblastoma cells', *Biophys. J.*, 2000, **79**, (1), pp. 163–183

81 Loew, L.M., and Schaff, J.C.: 'The Virtual Cell: a software environment for computational cell biology', *Trends Biotechnol.*, 2001, **19**, (10), pp. 401–406

82 Mitchell, A.: 'Web Watch … and the Virtual Cell project', *Nat. Rev. Mol. Cell Biol.*, 2001, **2**, (4), p. 231

83 Moraru, I.I., Schaff, J.C., Slepchenko, B.M., and Loew, L.M.: 'The virtual cell—an integrated modeling environment for experimental and computational cell biology', *Chromaffin Cell: Trans. Biosyn., Storage, Release, Actions, and Informatics*, 2002, **971**, pp. 595–596

84 Schaff, J., Fink, C.C., Slepchenko, B., Carson, J.H., and Loew, L.M.: 'A general computational framework for modeling cellular structure and function', *Biophys. J.*, 1997, **73**, (3), pp. 1135–1146

85 Schaff, J., and Loew, L.M.: 'The Virtual Cell'. Biocomputing: Proc. 1999 Pacific Symp., 1999, pp. 228–239

86 Schaff, J., Slepchenko, B., and Loew, L.M.: 'Physiological modeling with the Virtual Cell framework', in Johnson, M., and Brand, L. (Eds.): Methods in Enzymology' (Academic Press, San Diego, 2000), pp. 1–23

87 Schaff, J., Slepchenko, B., Morgan, F., Wagner, J., Resasco, D., Shin, D., Choi, Y.S., Loew, L., Carson, J., Cowan, A., Moraru, I., Watras, J., Teraski, M., and Fink, C.: 'Virtual Cell', 2001, http://www.nrcam.uchc.edu

88 Schaff, J.C., Slepchenko, B.M., Choi, Y.S., Wagner, J., Resasco, D., and Loew, L.M.: 'Analysis of nonlinear dynamics on arbitrary geometries with the Virtual Cell', *Chaos*, 2001, **11**, (1), pp. 115–131

89 Slepchenko, B.M., Schaff, J.C., Carson, J.H., and Loew, L.M.: 'Computational cell biology: spatiotemporal simulation of cellular events', *Ann. Rev. Biophys. Biomol. Struct.*, 2002, **31**, pp. 423–441

90 Slepchenko, B.M., Schaff, J.C., and Choi, Y.S.: 'Numerical approach to fast reactions in reaction-diffusion systems: Application to buffered calcium waves in bistable models', *J. Comput. Phys.*, 2000, **162**, (1), pp. 186–218

91 Slepchenko, B.M., Schaff, J.C., Macara, I., and Loew, L.M.: 'Quantitative cell biology with the Virtual Cell', *Trends Cell Biol.*, 2003, **13**, (11), pp. 570–576

92 Smith, A.E., Slepchenko, B.M., Schaff, J.C., Loew, L.M., and Macara, I.G.: 'Systems analysis of Ran transport', *Science*, 2002, **295**, (5554), pp. 488–491

93 Shin, D.G., Liu, L., and Loew, L.M.: 'Virtual Cell: a general framework for simulating and visualizing cellular physiology'. Presented at 4th IFIP 2.6 Working Conf. on Visual Database Systems, L'Aquila, Italy, 1998

94 Hermjakob, H., Montecchi-Palazzi, L., Bader, G., Wojcik, R., Salwinski, L., Ceol, A., Moore, S., Orchard, S., Sarkans, U., Von mering, C., Roechert, B., Poux, S., Jung, E., Mersch, H., Kersey, P., Lappe, M., Li, Y.X., Zeng, R., Rana, D., Nikolski, M., Husi, H., Brun, C., Shanker, K., Grant, S.G.N., Sander, C., Bork, P., Zhu, W.M., Pandey, A., Brazma, A., Jacq, B., Vidal, M., Sherman, D., Legrain, P., Cesareni, G., Xenarios, L., Eisenberg, D., Steipe, B., Hogue, C., and Apweiler, R.: 'The Hupopsi's molecular interaction format—a community standard for the representation of protein interaction data', *Nature Biotechnol.*, 2004, **22**, (2), pp. 177–183

95 Hucka, M., Finney, A., Sauro, H.M., and Bolouri, H.: 'Systems Biology Markup Language (SBML) level 1: structures and facilities for basic model definitions', 2001, http://www.sbml.org/

96 Ausbrooks, R., Buswell, S., Dalmas, S., Devitt, S., Diaz, A., Hunter, R., Smith, B., Soiffer, N., Sutor, R., and Watt, S.: 'Mathematical Markup Language (MathML) version 2.0 (second edition) W3C recommendation 21 October 2003', 2001, http://www.w3.org/TR/MathML2/

97 Doubletwist Inc., 'AGAVE: architecture for genomic annotation, visualization and exchange', 2001, http://www.agavexml.org

98 Fenyö, D.: 'The BIOpolymer Markup Language', *Bioinformatics*, 1999, **15**, (4), pp. 339–340

99 LabBook Inc., 'BSML (bioinformatics sequence markup language) 2.2', 2002, http://www.labbook.com/products/xmlbsml.asp

100 Liao, Y.M., and Ghanadan, H.: 'The Chemical Markup Language', *Anal. Chem.*, 2002, **74**, (13), pp. 389A–390A

101 Spellman, P.T., Miller, M., Stewart, J., Troup, C., Sarkans, U., Chervitz, S., Bernhart, D., Sherlock, G., Ball, C., Lepage, M., Swiatek, M., Marks, W.L., Goncalves, J., Markel, S., Iordan, D., Shojatalab, M., Pizarro, A., White, J., Hubley, R., Deutsch, E., Senger, M., Aronow, B.J., Robinson, A., Bassett, D.C.J.S., Jr., and Brazma, A.: 'Design and implementation of Microarray Gene Expression Markup Language (MAGE-ML)', *Genome Biol.*, 2002, **3**, (9), pp. 0046.0041–0046.0049

102 Gordon, P.: 'Multiple Sequence Alignments Markup Language (MSAML) DTD specification', 2002, http://maggie.cbr.nrc.ca/~gordonp/xml/MSAML/docs/

103 Taylor, C.F., Paton, N.W., Garwood, K.L., Kirby, D.A., Stead, D.A., Yin, Z., Deutsch, E.W., Selway, L., Walker, J., Riba-Garcia, I., Mohammed, S., Deery, M.J., Howard, J.A., Dunkley, T., Aebersold,

Syst. Biol., Vol. 1, No. 1, June 2004

51

R., Kell, D.B., Lilley, K.S., Roepstorff, P., Yates, J.R., Bass, A., Brown, A.J.P., Cash, P., Gaskell, S.J., Hubbard, S.J., and Oliver, S.G.: 'A systematic approach to modeling, capturing, and disseminating proteomics experimental data', *Nature Biotechnol.*, 2003, **21**, pp. 247–254

104 Hanisch, D., Zimmer, R., and Lengauer, T.: 'ProML—the Protein Markup Language', 2001, http://cartan.gmd.de/promlweb/

105 Hanisch, D., Zimmer, R., and Lengauer, T.: 'ProML—the Protein Markup Language for specification of protein sequences, structures and families', *In Silico Biology*, 2002, **2**, (3), pp. 313–324

106 McArthur, D.C.: 'An extensible XML schema definition for automated exchange of protein data: PROXIML (PROtein eXtensIble markup language)', 2001, http://www.cse.ucsc.edu/~douglas/proximl/

107 Altman, R.B., Brown, J.W., Case, D., Gautheret, D., Gutell, R., Harvey, S.C., Macke, T., Major, F., Westbrook, J., Westhoff, E., and Zuker, M.: 'Ribonucleic Acid Markup Language', 2002, http://www.smi.stanford.edu/projects/helix/riboml/

108 Cuellar, A.A., Nelson, M., and Hedley, W.: 'The CellML metadata 1.0 specification working draft—16 January 2002', 2002, http://www.cellml.org/public/metadata/cellml_metadata_specification.html

109 Cuellar, A.A., Lloyd, C.M., Nielsen, P.F., Bullivant, D.P., Nickerson, D.P., and Hunter, P.J.: 'An overview of CellML 1.1, a biological model description language', *Simulation*, 2003, **79**, (12), pp. 740–747

110 Garny, A., Kohl, P., and Noble, D.: 'Cellular open resource (COR): a public CellML based environment for modeling biological function', *Int. J. Bifurcation Chaos*, 2003, **13**, (12), pp. 3579–3590

111 Hedley, W.J., Nelson, M.R., Bullivant, D., Cuellar, A., Ge, Y., Grehlinger, M., Jim, K., Lett, S., Nickerson, D., Nielsen, P., and Yu, H.: 'CellML specification', 2001. Available on the Internet at http://www.cellml.org/public/specification/20010810/cellml_specification.html

112 Hedley, W.J., Nelson, M.R., Bullivant, D.P., and Nielson, P.F.: 'A short introduction to CellML', *Philos. Trans. R. Soc. Lond. A*, 2001, **359**, pp. 1073–1089

113 Physiome Sciences, Inc.: 'CellML Home Page', 2001, http://cellml.org/

114 Kiehl, T.R., Mattheyses, R.M., and Simmons, M.K.: 'Hybrid simulation of cellular behavior' (Technical Report, GE Global Research, 2003). Available on the Internet at http://www.crd.ge.com/cooltechnologies/biblio/2002grc294_bib.jsp

115 Kiehl, T.R., Mattheyses, R.M., and Simmons, M.K.: 'Hybrid simulation of cellular behavior', *Bioinformatics*, 2004, **20**, (3), pp. 316–322

116 Bernauer, S., Croft, D., Gardina, P., Minch, E., De Rinaldis, M., and Vatcheva, I.: 'Case study: data management strategies in an integrated pathway tool', *Appl. Bioinformatics*, 2004, **3**, pp. 63–75

117 Minch, E., De Rinaldis, M., and Weiss, S.: 'pathSCOUT: exploration and analysis of biochemical pathways', *Bioinformatics*, 2003, **19**, (3), pp. 431–432

118 Ortoleva, P.J., Berry, E., Ensman, L., Fan, J., Fontus, M., Friel, M., Harris, R., Hubbard, K., Jarymowycz, L., Navid, A., Park, A.J., Premkumar, P., Sayyed-Ahmad, A., Dong-Hoon, S., Shreif, Z., Stanley, F., Tuncay, K., and Weitzke, E.: 'Karyote', 2004, http://biodynamics.indiana.edu/cyber_cell/

119 Weitzke, E.L., and Ortoleva, P.J.: 'Simulating cellular dynamics through a coupled transcription, translation, metabolic model', *Comput. Biol. Chem.*, 2003, **27**, (4-5), pp. 469–480

120 Aleman-Meza, B.: 'Advances in numerical simulation of kinetics reactions equations' (Technical Report, Computer Science Department, University of Georgia, Athens, Georgia, USA, 2001)

121 Aleman-Meza, B., Yu, Y., Schüttler, H.-B., Arnold, J., and Taha, T.R.: 'Kinsolver', 2004, http://lsdis.cs.uga.edu/~aleman/kinsolver/

122 Segre, D., Vitkup, D., and Church, G.M.: 'Analysis of optimality in natural and perturbed metabolic networks', *Proc. Natl. Acad. Sci. USA*, 2002, **99**, (23), pp. 15112–15117

123 Segre, D., Church, G.M., Vitkup, D., and Rindone, W.: 'Minimization of metabolic adjustment (MOMA)', 2004, http://arep.med.harvard.edu/moma/

124 Ginkel, M., Kremling, A., Tränkle, F., Gilles, E.D., and Zeitz, M.: 'Application of the process modeling tool ProMot to the modeling of metabolic networks'. Proc. 3rd MATHMOD Conf., 2000, pp. 525–528

125 Ginkel, M., Kremling, A., Tränkle, F., Gilles, E.-D., and Zeitz, M.: 'ProMoT/DIVA', 2004, http://www.mpi-magdeburg.mpg.de/research/projects/1002/comp_bio/promot

126 Stelling, J., Ginkel, M., Bettenbrok, K., and Gilles, E.D.: 'Towards a virtual biological laboratory', in Kitano, H. (Eds.): Foundations of Systems Biology' (MIT Press, 2001), pp. 189–212

127 Bolouri, H., and Davidson, E.H.: 'Modeling transcriptional regulatory networks', *Bioessays*, 2002, **24**, (12), pp. 1118–1129

128 De Jong, H.: 'Modeling and simulation of genetic regulatory systems: a literature review', *J. Comp. Biol.*, 2002, **9**, (1), pp. 67–103

129 Ascher, U.M., and Petzold, L.R.: 'Computer methods for ordinary differential equations and differential-algebraic equations' (SIAM, Philadelphia, CA, 1998)

130 Gillespie, D.T.: 'Exact stochastic simulation of coupled chemical-reactions', *J. Phys. Chem.*, 1977, **81**, (25), pp. 2340–2361

131 Gibson, M.A., and Bruck, J.: 'Efficient exact stochastic simulation of chemical systems with many species and many channels', *J. Phys. Chem. A*, 2000, **104**, (9), pp. 1876–1889

132 Gillespie, D.T.: 'Approximate accelerated stochastic simulation of chemically reacting systems', *J. Chem. Phys.*, 2001, **115**, (4), pp. 1716–1733

133 Free Software Foundation, 'The GNU lesser general public license', 1999, http://www.fsf.org/licenses/licenses.html

134 Sourceforge.net, 'The SourceForge open source software development web site', 2002, http://www.sourceforge.net/

135 Hucka, M., Finney, A., Kovitz, B., Bornstein, B.J., Shapiro, B.E., Matthews, J., Keating, S., Schilstra, M., Funahashi, A., Taddeo, L., Doyle, J.C., and Kitano, H.: 'The SBML.org Web Site', 2004, http://www.sbml.org

136 Biron, P.V., and Malhotra, A.: 'XML Schema part 2: datatypes (W3C candidate recommendation 24 October 2000)', 2000, http://www.w3.org/TR/xmlschema-2/

137 Fallside, D.C.: 'XML Schema part 0: primer (W3C candidate recommendation 24 October 2000)', 2000, http://www.w3.org/TR/xmlschema-0/

138 Thompson, H.S., Beech, D., Maloney, M., and Mendelsohn, N.: 'XML Schema part 1: structures (W3C candidate recommendation 24 October 2000)', 2000, http://www.w3.org/TR/xmlschema-1/

139 Brownell, D.: 'SAX2' (O'Reilly and Associates, 2002)

140 Finney, A., Hucka, M., Sauro, H.M., Doyle, J.C., and Kitano, H.: 'An overview of the ERATO Systems Biology Workbench project'. Proc. 2nd Workshop on Computation and Biochemical Pathways, Heidelberg, Germany, 2001, pp. 115–122

141 Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J.C., and Kitano, H.: 'The ERATO Systems Biology Workbench: enabling interaction and exchange between software tools for computational biology', in Altman, R.B. (Ed.): Pacific Symposium on Biocomputing, 2002

142 Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J.C., and Kitano, H.: 'The ERATO Systems Biology Workbench: architectural evolution'. Proc. 2nd Int. Conf. on Systems Biology, 2001, pp. 352–361

143 Hucka, M., Sauro, H.M., Finney, A., Bolouri, H., Doyle, J., and Kitano, H.: 'The ERATO Systems Biology Workbench: an integrated environment for multiscale and multitheoretic simulations in systems biology', in Kitano, H. (Ed.): Foundations of Systems Biology, 2001

144 Systems biology workbench development group, 'The systems biology workbench development group', 2001, http://www.sbw-sbml.org/

145 Sauro, H.: 'SBML DLL library', version 1.45, 2003, http://www.cds.caltech.edu/~hsauro/sbml.htm

146 Vass, M.: 'JigCell SBML parser', 2003, http://jigcell.biol.vt.edu/SBML.html

147 Antoniotti, M.: 'CL-SBML', 2004, http://bioinformatics.nyu.edu/~marcoxa/common-lisp/code/cl-sbml/

148 Wolfram, S.: 'The Mathematica Book' (Cambridge University Press, 1999, 4th edn.)

149 Wolfram Research Inc., 'Mathematica', 2003, http://www.wolfram.com/products/mathematica/index.html

150 The Mathworks Inc., 'Using MATLAB' (The MathWorks, Inc., Natik, MA, 1998)

151 The Mathworks Inc., 'The MathWorks, Inc., web site', 2004, http://www.mathworks.com

152 Shapiro, B.E.: 'MathSBML', 2004, http://sbml.org/mathsbml.html

153 Shapiro, B.E., Hucka, M., Finney, A., and Doyle, J.C.: 'MathSBML: a package for manipulating SBML-based biological models', *Bioinf. Adv. Acc.*, 2004c

154 Kanehisa, M., and Goto, S.: 'KEGG: Kyoto encyclopedia of genes and genomes', *Nucleic Acids Res.*, 2000, **28**, (1), pp. 27–30

155 Kanehisa, M., Goto, S., Kawashima, S., and Nakaya, A.: 'The KEGG databases at GenomeNet', *Nucleic Acids Res.*, 2002, **30**, (1), pp. 42–46

156 Karp, P.D., Krummenacker, M., Paley, S., and Wagg, J.: 'Integrated pathway/genome databases and their role in drug discovery', *Trends Biotechnol.*, 1999, **17**, p. 275

157 Romero, P.R., and Karp, P.D.: 'Using functional and organizational information to improve genome-wide computational prediction of transcription units on pathway-genome databases', *Bioinformatics*, 2004, **20**, (5), pp. 709–717

158 Funahashi, A., Tanimura, N., Morohashi, M., and Kitano, H.: 'CellDesigner: a process diagram editor for gene-regulatory and biochemical networks', *BioSilico*, 2003, **1**, pp. 159–162

159 Funahashi, A., Tanimura, N., Morohashi, M., and Kitano, H.: 'CellDesigner', 2004, http://www.systems-biology.org/002/

160 Kitano, H.: 'A graphical notation for biological networks', *BioSilico*, 2003, **1**, pp. 169–176

161 Clark, J.: 'XSL transformations (XSLT) version 1.0 (W3C recommendation 16 November 1999)', 1999, http://www.w3.org/TR/xslt

162 Skonnard, A., and Gudgin, M.: 'Essential XML Quick Reference: A Programmer's Reference to XML, XPath, XSLT, XML Schema, SOAP, and More' (Addison-Wesley, 2001)

163 Nielsen, P., Hunter, P., Bullivant, D., Cuellar, A., Lloyd, C., Nickerson, D., and Halstead, M.: 'CellML model repository', 2004, http://www.cellml.org/examples/repository/index.html

164 Zucker, J.: 'Biocyc2SBML', 2003, http://genome.dfci.harvard.edu/~zucker/BPHYS/biocyc-open/

165 Novak, B., and Tyson, J.J.: 'Modeling the control of DNA replication in fission yeast', *Proc. Natl. Acad. Sci., USA.*, 1997, **94**, pp. 9147–9152

166 Tyson, J.J.: 'Modeling the cell division cycle: cdc2 and cyclin interactions', *Proc. Natl. Acad. Sci. USA*, 1991, **88**, pp. 7328–7332

167 De Young, G.W., and Keizer, J.: 'A single-pool inositol 1,4,5-Trisphosphate-receptor-based model for agonist-stimulated oscillations in Ca2+ concentration', *Proc. Natl. Acad. Sci. USA*, 1992, **89**, pp. 9895–9899

168 Kholodenko, B.N.: 'Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades', *Eur. J. Biochem.*, 2000, **267**, pp. 1583–1588

52

Syst. Biol., Vol. 1, No. 1, June 2004

169 Levchenko, A., Bruck, J., and Sternberg, P.W.: 'Scaffold proteins may biphasically affect the levels of mitogen-activated protein kinase signaling and reduce its threshold properties', *Proc. Natl. Acad. Sci. USA*, 2000, **97**, (11), pp. 5818–5823

170 Wolf, D.M., and Arkin, A.P.: 'Motifs, modules and games in bacteria', *Curr. Op. Microbiol.*, 2003, **6**, (2), pp. 125–134

171 Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U.: 'Network motifs: simple building blocks of complex networks', *Science*, 2002, **298**, pp. 824–827

172 Shen-Orr, S.S., Milo, R., Mangan, S., and Alon, U.: 'Network motifs in the transcriptional regulation network of Escherichia coli', *Nature Gen.*, 2002

173 Neves, S.R., and Iyengar, R.: 'Modeling of signaling networks', *BioEssays*, 2002, **24**, (12), pp. 1110–1117

174 Davidson, E.H., Rast, J.P., Oliveri, P., Ransick, A., Calestani, C., Yuh, C.H., Minokawa, T., Amore, G., Hinman, V., Arenas-Mena, C., Otim, O., Brown, C.T., Livi, C.B., Lee, P.Y., Revilla, R., Schilstra, M.J., Clarke, P.J., Rust, A.G., Pan, Z., Arnone, M.I., Rowen, L., Cameron, R.A., McClay, D.R., Hood, L., and Bolouri, H.: 'A provisional regulatory gene network for specification of endomesoderm in the sea urchin embryo', *Dev. Biol.*, 2002, **246**, (1), pp. 162–190

175 Goldstein, B., Faeber, J.R., Hlavacek, W.S., Blinov, M.L., Redondo, A., and Wolfsy, C.: 'Modeling the early signaling events mediated by FceRI', *Mol. Immun.*, 2002, **137**, pp. 1–7

176 Schürr, A.: 'PROGRES for Beginners', 1997, http://www-i3. informatik.rwth-aachen.de/research/projects/progres/main.html

177 Loriaux, P., Pettigrew, M., Lam, E., Cheng, B.N., Riches, M., Shanks, D., Keane, J.F., Sarkar, A., Borriello, G., Moon, R.T., and Franza, B.R.: 'Balsa', 2004, http://www.csi.washington.edu/teams/ modeling/projects/BALSA/index.html

178 Kirkwood, T.B.L., Boys, R., Wilkinson, D., Gillespie, C., Proctor, C., and Shanley, D.: 'Basis', 2003, http://www.basis.ncl.ac.uk/

179 Alur, R., Beals, C., Belta, C., Dunlap, P., Goulian, M., Ivancic, F., Kumar, V., Mintz, M., Pappas, G., Rubin, H., and Schug, J.: 'BioCharon', 2003, http://www.cis.upenn.edu/olf/biocomp/

180 Alur, R., Belta, C., Ivancic, F., Kumar, V., Rubin, H., Schug, J., Sokolsky, O., and Webb, J.: 'Visual programming for modeling and simulation of biomolecular regulatory networks'. Presented at Int. Conf. on High Performance Computing, 2002

181 Belta, C., Finin, P., Habets, L.C.G.J.M., Halasz, A., Imielinksi, M., Kumar, V., and Rubin, H.: 'Understanding the bacterial stringent response using reachability analysis of hybrid systems'. Proc. 7th Int. Workshop on Hybrid Systems: Computation and Control (HSCC), 2004, pp. 111–125

182 Webb, J., Belta, C., Goulian, M., Ivancic, F., Kumar, V., Rubin, H., Schug, J., Sokolsky, O., Gowen, S., Owen, A., Rubin, M., and Welber, L.: 'Bio sketch pad', 2004, http://bio.bbn.com/biospice/biosketchpad/ index.html

183 McCollum, M., and Lancaster, J.: 'BioSpreadsheet', version 1.0, Knoxville, 2003, http://biocomp.ece.utk.edu/

184 Kolpakov, F.: 'BioUML—Framework for visual modeling and simulation of biological systems'. Presented at Int. Conf. on Bioinformatics of Genome Regulation and Structure (BGRS), 2002, Akademgorodok, Novosibirsk, Russia

185 Kolpakov, F., Zhvaleev, V., Tyazhev, I., Onegov, O., and Kel, A.: 'BioUML', 2004, http://www.biouml.org/

186 Schwacke, J., and Voit, E.: 'BSTLab: A MATLAB toolbox for biochemical systems theory', 2003, https://bioinformatics.musc.edu/ bstlab/

187 Kurata, H., Matoba, N., and Shimizu, N.: 'CADLIVE for constructing a large-scale biochemical network based on a simulation-directed notation and its application to yeast cell cycle', *Nucleic Acids Res.*, 2003, **31**, (14), pp. 4071–4084

188 Kurata, H.: 'Cadlive', 2004, http://kurata21.bse.kyutech.ac.jp/cadlive/ download.html

189 Dhar, P., Meng, T.C., Somani, S., Ye, L., Sairam, A., Chitre, M., Hao, Z., and Sakharkar, K.: 'Cellware', 2004, http://www.bii.a-star. edu.sg/sbg/cellware

190 Dhar, P., Meng, T.C., Somani, S., Ye, L., Sairam, A., Chitre, M., Hao, Z., and Sakharkar, K.: 'Cellware—a multi-algorithmic software for computational systems biology', *Bioinformatics*, 2004, **20**, pp. 1319–1321

191 Shannon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T.: 'Cytoscape: a software environment for integrated models of biomolecular interaction networks', *Genome Res.*, 2003, **13**, (11), pp. 2498–2504

192 Shannon, P., Ozier, O., Markiel, A., Ramage, D., Amin, N., Wang, J., and Chang, M.: 'Cytoscape', 2003, http://cytoscape.org

193 Ramsey, S., and Bolouri, H.: 'Dizzy', 2004, http://labs. systemsbiology.net/bolouri/software/Dizzy/

194 Weimar, J.R.: 'ecellJ', 2003, http://www.jweimar.de/ecellJ

195 Cox, C.D., Peterson, G.D., Allen, M.S., Lancaster, J.M., McCollum, J.M., Austin, D., Yan, L., Sayler, G.S., and Simpson, M.L.: 'Analysis of noise in quorum sensing', *OMICS*, 2003, **7**, (3), pp. 317–334

196 Cox, C.D., Peterson, G.D., Sayler, G.S., Simpson, M.L., and McCollum, J.M.: 'Exact stochastic simulator (ESS)', 2004, http:// biocomp.ece.utk.edu/tools.html

197 Sauro, H.S.: 'JDesigner: A simple biochemical network designer', 2001, http://www.cds.caltech.edu/~hsauro/

198 Allen, N., Vass, M., Zwolak, J., Ramakrishan, N., Shaffer, C., Sible, J., Tyson, J.J., and Watson, L.: 'The eukaryotic cell cycle collaborative problem-solving environment group', 2003, http://gnida.cs.vt.edu/ ~cellcyclepse/

199 Allen, N.N., Calzone, L., Chen, K.C., Ciliberto, A., Ramakrishnan, N., Shaffer, C.A., Sible, J.C., Tyson, J.J., Vass, M.T., Watson, L.T., and Zwolak, J.W.: 'Modeling regulatory networks at Virginia Tech', *OMICS*, 2003, **7**, (3), pp. 285–299

200 Allen, N.A., Shaffer, C.A., Vass, M.T., Ramakrishnan, N., and Watson, L.T.: 'Improving the development process for eukaryotic cell cycle models with a modeling support environment'. Proc. Winter Simulation Conf., 2003, pp. 782–790

201 Allen, N.A., Shaffer, C.A., Vass, M.T., Ramakrishnan, N., and Watson, L.T.: 'Improving the development process for eukaryotic cell cycle models with a modeling support environment', *Simulation*, 2003, **79**, (12), pp. 674–688

202 Allen, N.N., Calzone, L., Chen, K.C., Ciliberto, A., Ramakrishnan, N., Shaffer, C.A., Sible, J.C., Tyson, J.J., Vass, M.T., Watson, L.T., and Zwolak, J.W.: 'JigCell', 2003, http://jigcell.biol.vt.edu

203 Vass, M., Carroll, J.M., and Shaffer, C.A.: 'Supporting creativity in problem solving environments'. Proc. 4th Creativity & Cognition Conf., Loughborough, UK, 2002, pp. 31–37

204 Bassingthwaighte, J.B.: 'Strategies for the Physiome Project', *Ann. Biomed. Eng.*, 2000, **28**, (8), pp. 1043–1058

205 Butterworth, E., Lawson, E., Raymond, G., Li, Z., Kellen, M., Packer., A., Schwab, A., and Bassingthwaighte, J.B.: 'JSim', 2003, http://nsr.bioeng.washington.edu/PLN/Members/butterw/JSIMDOC1.6/ JSim_Home.stx/view

206 Snoep, J.L., and Olivier, B.G.: 'Java web simulation (JWS); a web based database of kinetic models', *Mol. Biol. Rep.*, 2002, **29**, (1–2), pp. 259–263

207 Snoep, J.L., and Olivier, B.G.: 'JWS online cellular systems modeling and microbiology', *Microbiology*, 2003, **149**, pp. 3045–3047

208 Snoep, J., and Olivier, B.: 'Java web simulation', 2004, http://jjj. biochem.sun.ac.za/index.html

209 Olivier, B.G., and Snoep, J.L.: 'Web-based kinetic modelling using JWS Online', *Bioinf. Adv. Acc.*, 2004

210 Soergel, D., George, B., Morgan-Linial, R., Brent, R., and Endy, D.: 'Monod, a tool to support collaborative modeling of biological processes' (Unpublished manuscript, Molecular Sciences Institute, Berkeley, California, USA, 2003)

211 Soergel, D., George, B., Morgan-Linial, R., Brent, R., and Endy, D.: 'Monod', 2004, http://monod.molsci.org/

212 Brown, C.T., Rust, A.G., Clarke, P.J.C., Pan, Z., Schilstra, M.J., De Buysscher, T., Griffin, G., Wold, B.J., Cameron, R.A., Davidson, E.H., and Bolouri, H.: 'New Computational Approaches for Analysis of cis-Regulatory Networks', *Dev. Biol.*, 2002, **246**, pp. 86–102

213 Schilstra, M., and Bolouri, H.: 'NetBuilder', 2002, http://strc.herts.ac. uk/bio/maria/NetBuilder/index.html

214 Sreeni, 'PathArt', 2004, http://www.jubilantbiosys.com

215 Gilman, A., Jacobsen, J., Chan, L., and Arkin, A.: 'PathwayBuilder', 2004, http://biospice.lbl.gov/software/NRMS/description.html

216 Patel, M., and Nagl, S.: 'MicroCore: mapping genome expression to cell pathways and networks', *Comp. Funct. Gen.*, 2004, **5**, (1), pp. 75–78

217 Patel, M., and Nagl, S.: 'SCIpath', 2004, http://www.ucl.ac.uk/ oncology/MicroCore/microcore.htm

218 Weinstein, H., Campagne, F., Skrabanek, L., Srdanovic, M., Chan, E., Martin, M., Shi, L., Mukherjee, P., Chang, V., Honig, M., Iyengar, R., Neves, S., Cerami, E., Chiang, B., Fevre, F.L., Ram, P., and Wilson, M.: 'SigPath', 2004, http://icb.med.cornell.edu/crt/SigPath/index.xml

219 Divalentin, P., and Pettigrew, M.: 'SigTran', 2004, http://www.csi. washington.edu/teams/modeling/projects/sigtran/index.html

220 Antoniotti, M., Ugel, N., Paxia, S., and Mishra, B.: 'Simpathica', 2002, http://bioinformatics.nyu.edu/Projects/Simpathica/

221 Antoniotti, M., Policriti, A., Ugel, N., and Mishra, B.: 'XS-systems: eXtended S-Systems and algebraic differential automata for modeling cellular behavior'. Proc. 9th Int. Conf. on High Performance Computing (HiPC), Bangalore, India, 2002, pp. 431–442

222 Antoniotti, M., Policriti, A., Ugel, N., and Mishra, B.: 'Model building and model checking for biochemical processes', *Cell Biochemistry and Biophysics*, 2003, **38**, (3), pp. 271–286

223 Antoniotti, M., Policriti, A., Ugel, N., and Mishra, B.: 'Reasoning about biochemical processes', *Cell Biochem. Biophys.*, 2003, **38**, pp. 271–286

224 Rost, U., and Kummer, U.: 'SimWiz: A Java package for the visualization of biochemical simulation processes', 2004, http:// projects.villa-bosch.de/bcb/software/software/Ulla/SimWiz/

225 Kierzek, A.M.: 'STOCKS: STOChastic kinetic simulations of biochemical systems with gillespie algorithm', *Bioinformatics*, 2002, **18**, (3), pp. 470–481

226 Puchalka, J., and Kierzek, A.M.: 'STOCKS', 2004, http://www.sysbio. pl/stocks/

227 Warren, L.: 'Trelis', 2003, http://sourceforge.net/projects/trelis

228 Duncan, J., Arnstein, L., and Li, Z.: 'Teranode Corporation launches first industrial-strength research design tools for the life sciences at DEMO 2004', 2004, http://www.teranode.com/about/pr_2004021601. php

229 Teranode Inc., 'VLX Design Suite', 2004, http://www.teranode.com/ products

230 Sauro, H.M., and Fell, D.A.: 'Scamp—A metabolic simulator and control analysis program', *Math. Comput. Model.*, 1991, **15**, (12), pp. 15–28

231 Sauro, H.M.: 'Scamp—a general-purpose simulator and metabolic control analysis program', *Comput. Appl. Biosci.*, 1993, **9**, (4), pp. 441–450

232 Sauro, H.M.: 'WinScamp', 2003, http://www.cds.caltech.edu/ ~hsauro/Scamp/scamp.htm

Syst. Biol., Vol. 1, No. 1, June 2004

53