SBML at 18: More Sophisticated, but with Room to Grow - 2018 Version

List Authors here

August 4, 2018

Abstract

The rise of systems biology brought up an increase in the number, size and complexity of mathematical models used in biology. To reproduce results and re-use models, we need to be able to exchange precise and unambiguous descriptions of their structure and biological meaning. The Systems Biology Markup Language is a community-developed standard format to encode the structure of systems models, coming with a rich software toolkit. SBML Level 3 provides a modular structure, with a core supporting reaction-based models extended by packages that allow the description of a variety of model types, such as constraint-based models, reaction-diffusion models, logic or rule-based models, and provide additional capabilities, for instance building multi-scale models or encode graphical representations. Over the past two decades, SBML has transformed the way systems biologists build and interact with models, and played an important role in increasing model quality, sharing and re-use. However, the landscape of modeling is changing, with the increasing development of multi-scale models of whole cells and organs, and new data such as single cells and live imaging triggering novel ways of integrating data and models. SBML Level 3 provides the foundation needed to support such a revolution.

Introduction

The use of systems modeling and numerical simulations in biology can be traced to the mid-20th century, notably with the works of Hodgkin and Huxley on the molecular basis of neuronal transmission (Hodgkin and Huxley, 1952) and Britton Chance on the mechanism of catalase action (Chance et al., 1952). Since then, the number and variety of mathematical models has steadily grown in all fields of life sciences. These models can be simulated, analyzed, tested, and compared to experimentally-derived data. The insights gained can then be used to confirm or infirm hypothesis, suggest additional experiments, and refine models (Le Novère, 2015). The availability of genome-wide functional information, sophisticated modeling methods, and easily accessible computer power, led to the rise of systems biology as a new domain of research (Kitano, 2000; Ideker et al., 2001). For the increasing number of models to be used and re-used efficiently, they needed to be expressed using precisely-defined digital formats that can be communicated directly between different software systems, databases, and people. This realization around the turn of the millennium drove the creation of the Systems Biology Markup Language (SBML, `http://sbml.org`), an open, machine-readable format for the representation of computational models.

SBML originated as a byproduct of another project, the Systems Biology Workbench (SBW, Hucka et al., 2001), designed to enable separate simulation-oriented software packages to interact. The effort needed something that did not exist at the time, namely a tool-*independent* way to represent computational models, removing the potential for translation errors and assuring a common starting point for analyses and simulations. Moreover, the community needed a commonly-accepted format to distribute models as supplementary materials for publications. The decision to create a neutral format based on XML (bray2008, a novel technology at the time) was articulated during a workshop organized at the California Institute of Technology in Pasadena, California, in the year 2000. The initial structure was inspired by a previous effort aimed at exchanging metabolic models (Kell and Mendes, 2008). After further revisions, discussions and software implementations, the final specification for SBML Level 1, Version 1, was released in March, 2001 (Hucka et al., 2003).

The format then evolved in a community-driven fashion, benefiting from the efforts of many people worldwide over more than a decade and a half. A guiding principle in its development has been to seek consensus between different viewpoints and the needs of different groups, to find a middle ground that would be—while perhaps not a perfect solution—an *acceptable* and *usable* solution. It was always understood that more types of models existed than SBML initially supported when it was first introduced. However, achieving community consensus on a more straightforward set of simpler features for which support could be readily implemented was deemed a more accessible and more pragmatic strategy. A deliberate decision was taken to add more advanced capabilities later, and SBML thus evolved in stages.

SBML Level 2 was published in 2002, introducing among other things support for discrete events and complex metadata, as well as re-use of other standards such as MathML and Unicode. The format came with a rich toolkit that facilitates its support in modeling and simulation software and its use by end-users (Box 1). While it was initially developed to support non-spatial compartmental models of biochemical reaction networks using chemical kinetics (Hucka et al., 2002), SBML developers soon saw the use for a broader range of other model types, modeling paradigms, and research areas. In addition to reaction-diffusion models, alternative modeling frameworks rose in the past decade, such as constraint-based, logic and rule-based modeling. The progressive development of SBML Level 2 was thus accompanied by in-depth discussions, resulting in the release of SBML Level 3 in 2010 (Hucka et al., 2015) that introduced a profound change in structure enabling the support needed for a much larger variety of models.

Structure of SBML

SBML aims at describing the primary structures of a systems model. Since its inception, SBML's main strength has been encoding kinetic models for dynamic simulations, for instance of metabolic or cell signaling pathways. The constructs provided by SBML Core are in particular well suited for representing reactions between well-mixed pools of reactants in spatially homogeneous compartments (Figure 1B), including initial conditions, kinetic laws, and all the necessary mathematical relationships between model variables. Listing reactions separately allows software systems to exchange the biological structure of the models together with their mathematical structure precisely and unambiguously. Such a granular structure increases model re-usability. Some SBML models might even use no mathematics at all, and only carry the reaction structure. In fact, SBML is often used to exchange reaction maps (e.g. Oda et al., 2005). Finally, encoding reactions separately considerably facilitates the modification of existing models by adding, replacing or removing reactions, without having to reverse-engineer the model from mathematical equations and make all the required modifications, an error-prone procedure.

However, besides reactions, SBML provides generic facilities to encode a large diversity of mathematical models. All mathematical constructs in SBML are encoded using a subset of MathML (Ausbrooks et al., 2003). Values of model variables may be fixed or changed by mathematical expressions, either before or during simulation, continuously or in response to discrete events. One can also compute the rate of change of a variable. Therefore, SBML can support models described purely with mathematical equations, for example, sets of differential equations — although this should be discouraged when we know the biological mechanisms in play. Moreover, SBML is not restricted to a particular field of the life sciences or a certain level of description. The meaning of the model components is up to the modeler and might depend on the domain. For instance, the variables can be populations of molecules, cells, or even organisms, and the mathematical constructs can represent relationships at any scale. Finally, any SBML element can be extended with annotations which can cover capabilities not yet included in the public specifications, or tool-specific structures.

Although some models deal with dimensionless variables, units can be added to most relevant SBML constructs. In addition to add a layer of physical knowledge to the model, units can be used to validate mathematical equations and help converting models between modeling frameworks. SBML supports a large set of base units from the international systems of units, and provide mechanisms to create custom ones. Units are optional. However, if used, none should be missing or unit validation is turned off in supporting tools.

Any element of a model can be documented using human readable notes encoded in XHTML (W3C HTML Working Group, 2002). To comply with the MIRIAM guidelines (Le Novère et al., 2005), metadata can also be added to model elements via controlled annotations using W3C's semantic web technologies. A layer of biological meaning can be added by using standard Identifiers.org URIs (Juty et al., 2012) to create cross-references to external terminologies and databases, for instance linking a model's species definitions to entries in UniProt (The UniProt Consortium, 2017) if it represents a protein or to ChEBI (Hastings et al., 2013) if it represents a simple chemical. Gene Ontology terms (Ashburner et al., 2000) can be attached to compartments, species, and all mathematical objects representing biological process and functions. Clerical data such as personal details or timestamps facilitates logging, tracking and versioning.

SBML does not specify which simulation framework to use. A given model can (for example) be simulated as a stochastic system using algorithms such as Gillespie (1977), which keeps each process separate, or via the numerical solution of differential equations representing all the processes affecting a given molecular species, or something else entirely. This abstract approach allows software tools to translate a model into whatever internal form they use. In fact, SBML

does not specify what to do with a model, for instance which simulation to run, how to run them and how to present and interpret the results. Other formats exist for this purpose such as SED-ML (Waltemath et al., 2011).

SBML development anticipated the conflicting pressures faced by (academic) software developers to produce stable software that meets the needs of their users — usually focused on a subset of the modelling landscape —, within the funding constraints that often restricts development of such software. The resulting *Levels* of SBML attempt to achieve a consistent language at a certain level of complexity and co-exist. Within a *Level*, *Versions* introduce gradual changes and maintain backward compatibility. *Level 3 packages* have further broadened the scope of SBML without rendering existing software obsolete and allows tools to develop support for only those aspects of *Level 3* that are relevant to their particular tool.

Figure 1: A close look at SBML. A) Global structure of an SBML file. In this example, the model description declares the use of the Layout package. Model's structures include the descriptions of model variables, as well as their relationships. Package elements can refer to elements declared in SBML core or other packages. Elements of the same type are defined in ListOf containers, e.g., model parameters in the ListOfParameters. B) Links between symbols used in the mathematical constructs and the elements describing the reactions, the molecular species, and their localization.*

Box 1. Software infrastructure supporting SBML

Application Programming Interface

Libraries that support reading, writing, manipulating, validating, and transforming SBML.

1. LibSBML (Bornstein et al., 2008) (http://sbml.org/Software/libSBML) is written in C++ and offers language interfaces for C, C++, C#, Java, JavaScript, MATLAB, Octave, Perl, PHP, Python, Ruby, and R.
2. JSBML (Rodriguez et al., 2015) (http://sbml.org/Software/JSBML) is a pure-Java based implementation.

Both libraries:

- Support all levels and versions of SBML
- Support all L3 packages
- Use LGPL License

Software Guide

A collection (> 290 to date) of software applications, libraries, and online services that support SBML. Users can browse the guide http://sbml.org/SBML_Software_Guide as:

- A tabular form that highlights supported SBML features
- A categorized list that displays a summary
- An online gallery of screenshots (where provided)

Test Suite

The *SBML Test Suite* (https://sbml.org/Software/SBML_Test_Suite) facilitates developers in supporting SBML. It provides

1. Thousands of test cases for
   - Semantic support (using results from both deterministic and stochastic simulation)
   - Syntactic support
   - All Levels/Versions of SBML and relevant L3 packages
2. A graphical front end that enables test cases to be filtered by Level/Version and a range of test tags.
3. An online database where test results can be uploaded and compared with results from other simulators.

Validation

Ensuring correct use of the standard, validation is available via

1. API libraries
2. *SBML online validator* (https://sbml.org/Facilities/Validator/)
3. SBML validator web service

Validation ensures compliance with

- XML syntax
- SBML Validation Rules published as part of each accepted SBML specification
- Syntax of L3 packages under development

SBML Level 3's modularity and breadth

SBML Level 2 was perfectly suited for encoding chemical kinetics models in homogeneous compartments, a field of modeling where methods and best practices were fairly standard. To cover the wider breadth of modeling in systems biology and support larger and more complex models SBML Level 3 introduced the concept of modularity, with a *core* derived from SBML Level 2 and *Packages* (Figure 1A). *Packages* augment the capabilities of SBML *core* by extending components, adding new components, and changing the meaning or scope of components (within certain limits). That way, SBML Level 3 supports many model types more naturally than if they had to be shoehorned into core SBML constructs. New packages can be developed independently, within dedicated communities as the need arise. This has been the case for logic modeling (Naldi et al., 2015) and constraint-based modeling (Ebrahim et al., 2015). Such a development can take place at a pace that suits their developers and package can be adopted when people are ready.

The list of Level 3 packages declared by models guide their interpretation and handling by software applications. For instance, software tools can detect the presence of features that they may not support, and thus inform users why their tool may not be able to work with a given model. Some Level 3 packages do not affect the mathematical interpretation of a model and can be entirely ignored in a simulation setting (e.g., the Layout package for storing diagrams). However, even when an application does not support a package that adds crucial details to a model, it can nevertheless make some interpretation of fundamental aspects of the model by understanding Level 3 Core and perhaps other packages used by the model. For example, the reaction network of a reaction-diffusion model might still provide useful information even without the spatial context.

Though a modular approach has benefits, it is also not without potential pitfalls. Among the main risks are the fragmentation of the community, and incompatibility of packages, due to complex dependencies of features. The SBML community addressed the former by maintaining communications and interactions between supporters of the various packages while API libraries (see Box Software) can handle *some* combinations of SBML Level 3 packages today and hide some of the complexity.

Twelve SBML Level 3 packages are under active consideration today (Table 1). Together, they provide new structural capabilities (grouping of model elements into sets, model composition, creation of arrays of elements, use of distributions and sampling, and extension of the mathematical constructs), support for new types of models (constraint-based model, qualitative models, rule-based models, spatial models and multi-agent models), and model representation (layout and rendering of model maps) (Figure 2). To date, six packages have been fully developed into consensus specifications and feature at least two software implementations (Box 2). Another three packages have a draft specification sufficiently advanced that some software tools implemented support.

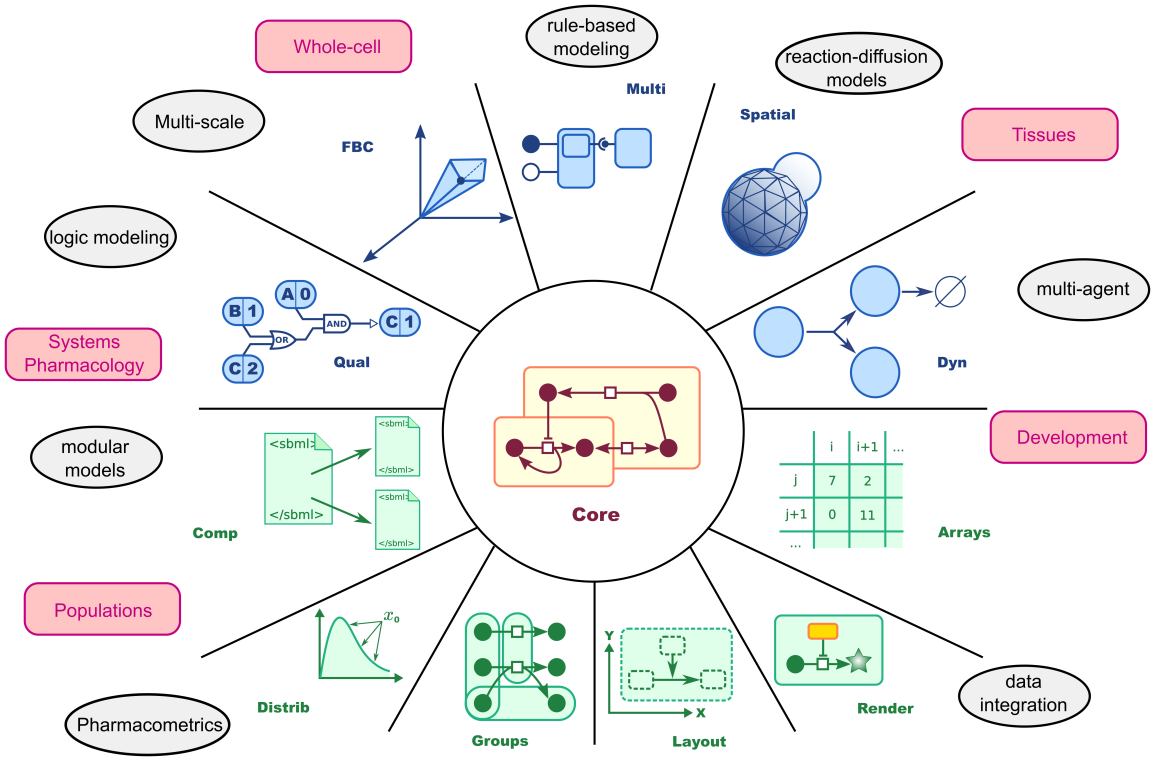


Figure 2: Extending the Level 3 Core with packages bringing new syntactical constructs (in green) and covering new modeling approaches (in blue) will make SBML better suited to encode the types of modelling (in grey) needed for large and complex models used in physiology, development and disease modeling.

Table 1: Status of Level 3 packages. Symbols: ✓ = done; 🕒 = in progress; n/a= not applicable.

Package name	Description	Specification Status	libSBML Support	JSBML Support	Test Suite
Groups	Grouping elements for conceptual purposes. Groups have no mathematical meaning and are not intended to affect simulations.	✓	✓	✓	n/a
Hierarchical Model Composition	Support for defining models composed of other models. Those "submodels" can be stored in the same file or as separate files.	✓	✓	✓	✓
Arrays	Support for defining arrays of components, such as arrays of species or arrays of reactions. (SBML Level 3 core supports only scalar values, not arrays.)	🕒	✓	✓	🕒
Distributions	Support for defining statistical distributions for quantitative model components (SBML Level 3 Core does not provide a way to indicate distributions of numerical values.	🕒	✓	✓	🕒
Extended math	Package that adds support for MathML constructs beyond the subset allowed in SBML Core.	🕒	🕒	🕒	🕒
Flux Balance Constraints	Constraint-based (also known as steady-state) models.	✓	✓	✓	✓
Qualitative Models	Supports direct expression of Boolean, Petri Net, and similar models in which species variables do not represent quantities but states.	✓	✓	✓	🕒
Multistate, Multicomponent, and Multicompartment Species	Supports the definition of species that possess complex features such as states or binding sites, and supports the use of rules-based models.	✓	✓	✓	🕒
Spatial Processes	Supports the definition of spatially inhomogeneous geometries and processes such as diffusion.	🕒	✓	✓	🕒
Dynamical Processes	Creation, destruction, and movement of entities during simulation, which is especially useful for multicellular models.	🕒	✓	✓	🕒
Layout	Storing position and size of model components for drawing network diagrams. The separate Rendering package specifies details of visual appearance.	✓	✓	✓	n/a
Rendering	Extends the Layout package by storing graphical symbols and styles, curves, colors, and gradients in network diagrams.	🕒	✓	✓	n/a

Box 2. Published SBML Level 3 packages

Comp The *Hierarchical Model Composition* package (Smith et al., 2015) allows users to build a model from other complete models or from submodels, as a way to manage complexity or construct composite models. Submodels can be defined within the main SBML file or linked from external model definitions. Multiple instances of a given model definition (i.e. template) can be linked from an enclosing model, to avoid duplication of elements and reuse model parts. The package provides defined interfaces (ports) for optional black-box encapsulation. A hierarchical model can be 'flattened' into a single monolithic model using libSBML (Bornstein et al., 2008).

FBC The *Flux Balance Constraints* package (Olivier and Bergmann, 2018) provides means for encoding constraint-based models and optimizations, e.g., Flux Balance Analysis (Bordbar et al., 2014). Constructs are added for defining the list of objectives for minimization or maximization, for setting flux bounds on reactions, and defining gene-reaction mappings. Additional information such as chemical formula and charge enable further model analyses including calculation of reaction mass balances, electron leaks, or implausible sources of matter.

Groups The *Groups* package (Hucka and Smith, 2016) allows indicating conceptual connections between elements in a model. Components can be grouped together to indicate classification, partonomy, or merely a collection of things. The meaning of the group can be elaborated through the use of annotations on the group structure. Groups may contain either the same or different types of SBML objects, and groups may be nested if desired. There is no predefined behavioral semantic associated with groups, and they cannot influence the mathematical interpretation of a model.

Multi The *Multistate, Multicomponent and Multicompartment Species* package (Zhang and Meier-Schellersheim, 2018) adds functionality for representing entities composed of multiple components that can exist in multiple states. Rules can be defined by specifying how reactions depend on the particular states of the entities and their locations. To this end, the package adds syntactic constructs for species types, compartment types, features, binding sites, and bonds. By introducing 'patterns' which describe entire families of molecule complexes sharing certain properties, this 'rule-based, domain-detailed' modeling approach (Blinov et al., 2004) strongly simplifies the definition of models.

Qual The *Qualitative Models* package (Chaouiya et al., 2013) adds constructs to encode models whose dynamics can be represented by discrete reachable states connected by state transitions denoting qualitative updates of model elements. Examples incluse logic regulatory networks (Boolean or multi-valued (Abou-Jaoudé et al., 2016)) or Petri nets (Chaouiya, 2007). Qualitative species are used to associate discrete levels of activities with entity pools, whereas transitions define the possible changes between states in the transition graph.

Layout & Render The *Layout* (Gauges et al., 2015) and *Rendering* (Bergmann et al., 2017) packages define the syntax for storing models as graphical diagrams of networks or pathways. The layout package defines a way to encode the positions and size of graphical elements such as nodes and lines in a diagram, while information about colors, styles, fonts, etc., are defined by the rendering package. The separation accomplishes several goals: First, applications can offer multiple styles for visualizing the same layout of a network map. Second, it simplifies the implementation because most of the essential aspects of a reaction network diagram can be expressed using the layout package, and thus tools do not need to implement a full graphics environment. Features for customizing the look-and-feel of a diagram are left to the separate rendering package.

SBML as a community standard

SBML is a community standard, entirely developed by its users, and not a format imposed by any formal entity, whether a research organization, a governmental agency or an official standardization body. Over the years, the community designs rules to set-up its governance, maintains and develops the specifications, and supports the use of the language.

SBML has been successful in large part because its development has followed a bottom-up approach driven by goals expressed in the community. This engages the researchers and software developers who constitute SBML's foremost stakeholders. It furthermore ensures that development is driven by needs and allows fast feedback from users to developers. The development of SBML and its Level 3 packages is shepherded by the SBML Editors, a group of community-elected volunteers, serving terms of three years (See Box 3 for a more detailed description of the process leading to the release of a new package). SBML editors write or review SBML specification documents, organize discussions and votes on specific technical issues, and enact the decisions of the community. Besides, the SBML Team, a group of people with dedicated financial supports (of which the main one is a current grant from the US National Institutes of Health), develops the core SBML supporting software (Box 1).

SBML hosts specific mailing lists for people who want to be informed about changes related to SBML, such as updates of the SBML specifications, libraries and software support. The main changes to the specifications are discussed by the community through the SBML mailing list `sbml-discuss` or the lists dedicated to specific packages. Detailed information about the mailing lists is available at `http://sbml.org/Forums/`. Interested persons can also contact the SBML Editors directly at `sbml-editors@googlegroups.com` and the SBML Team at `sbml-team@googlegroups.com`.

Many changes are proposed and discussed by the community at the annual SBML meetings. The community comes together twice a year within the context of the COMBINE (Computational Modeling in Biology Network, `http://co.mbine.org`) meetings: The HARMONY hackathons take place in the first half of each year and focus on the support of existing SBML, in particular via the development of libraries, tools and specifications. The COMBINE forums take place in the second half of each year and focus on the presentation of novel relevant tools and the discussion of proposed features. In addition to these general meetings, special working groups meet regularly to drive package development. It is worth noting that the resulting SBML development has stimulated collaborative work and the creation of consortia, leading to better awareness and communication within groups interested in specific modeling frameworks. A good example is the CoLoMoTo effort, partially launched by and responsible for the development of the Qualitative Modeling Package (Naldi et al., 2015).

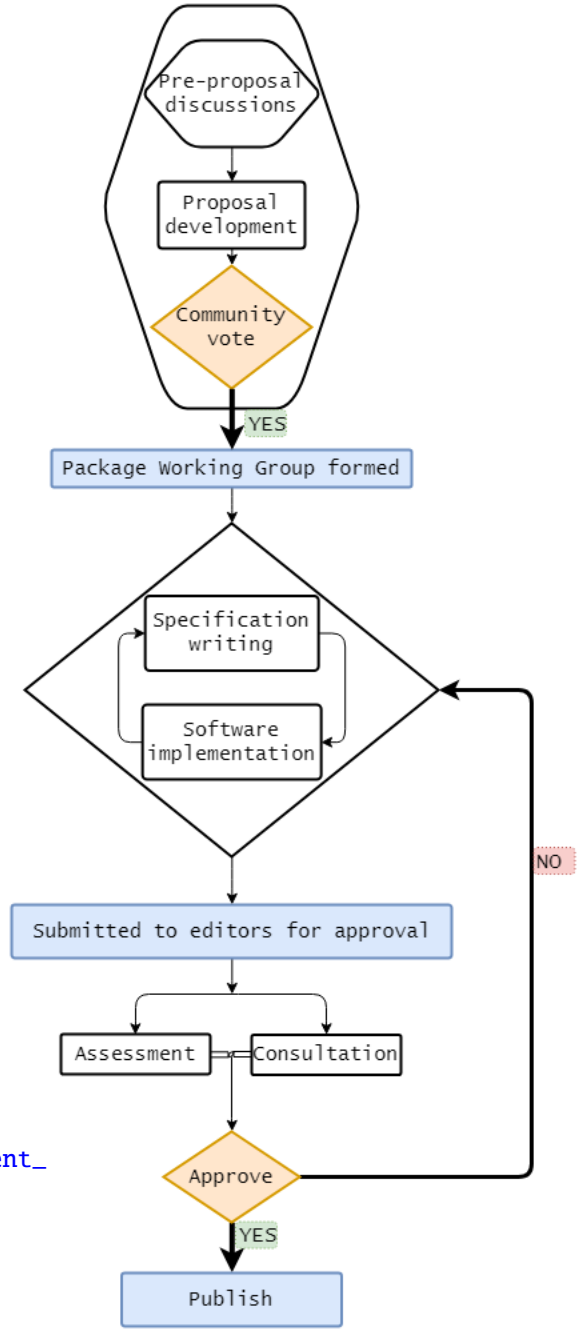
Box 3. Package development process

Package proposal stage The SBML Editors follow the *SBML Development Process*, which includes a set of procedures that allows anyone to propose a new SBML package. Proposals may be as detailed as the authors wish, but must address the need being served by the package, and the general plan for how the package will address that need. Proposals are first voted on by the SBML community using a set of criteria that indicate that the area the package intends to address is appropriate, and that the proposed approach seems viable.

Package development stage Once a proposal has been approved a package working group (PWG) is formed and a dedicated mailing list established. All interested parties are encouraged to subscribe. Members of the PWG work on both specification and implementation; iterating on both following decisions made at each point. Once members have finalized the specification, they submit it to the SBML Editors, including a covering letter detailing the implementations.

Editor approval stage The SBML Editors review the specification for style and clarity. They establish the exchange of models between implementations and, where appropriate, check that the results of simulation/analysis are reproducible and accept the page as official if all the criteria are satisfied.

Full details of the development process are available at http://sbml.org/Documents/SBML_Development_Process/SBML_Development_Process_for_SBML_Level_3.



Impact of SBML

Both the process of developing SBML and the language itself had a profound impact on the field of systems modeling. The SBML development process helped to structure the domain by directly and continuously involving the community of developers and users of modeling software. The frequent meetings improved feedback from modelers to software developers. They also increased the awareness of existing tools - as did the Software Guide (see Box 1). Furthermore, developing SBML instilled a culture of model sharing and re-use in systems biology. The ability to exchange models between software tools, and re-use them, as such or as a starting point for new modeling efforts, changed the way people built and used models. It also created new activities centered on the models themselves, including automatic model generation, analysis of model structures, model retrieval, and integration of models with other types of data. Existing software tools allow modelers to use SBML in all aspects of a modeling project, including creation (manual or automated), manipulation, annotation, comparison, merging, parametrization, simulation/analysis, comparison of analysis results, network motif discovery, system identification, and omics data integration, visualization of models, and more (see Box 4).

Before the advent of SBML, it was extremely challenging to exchange models between different modeling tools since most were using incompatible scripting languages, or even hard-coded the mathematics. The rise of systems biology led to an increase in models' size and complexity that made rewriting a model from scratch even more difficult and error-prone. Modeling projects relying on a single software tool are becoming the exception. SBML permits the use of a single description throughout the modeling life cycle (Box 4). While the exchange of chemical kinetics models in SBML has been a staple of computational systems biology for a few years now, it also recently became possible for qualitative and rule-based models with the Qual and Multi package (Box 2). For example, CellNOpt (Terfve et al., 2012) provides a set of optimal Boolean models that best explains the causal relationships between components of a signaling transduction network and associated data. Dynamical properties of these models can be studied with GINsim (Chaouiya et al., 2012), varying the updating schemes, identifying the model attractors, mutant behaviors, etc.

Encoding SBML also facilitated the comparison of analyses obtained from different software tools. If a deviation is identified, modelers and software developers can further investigate the reasons. It has been shown that wrong interpretation of genome-scale models by different software was alleviated by the use of proper SBML (Ebrahim et al., 2015). In rule-based modeling, the combinatorial numbers of reactions makes impractical to check the correctness of all expended rules. Two rule-based modeling tools, such as *Simmune* (Zhang et al., 2013) and *BionetGen* (Faeder et al., 2009) can nevertheless easily simulate the same models by supporting the SBML Multi package. A simple comparison of the total numbers of well-defined species from the same starting point can give good confidence that the models are defined with correct rules and that software tools behave accurately. As a consequence, using sets of SBML-encoded models has become the norm to assess the accuracy of system modeling software, initially using BioModels as a source of curated models (Bergmann and Sauro, 2008), and now using the SBML test suite (Box 1).

SBML eased automated process of mathematical models so they became just another type of data in the life sciences, that one can generate, process, share, etc. SBML is nowadays used as import/export format by many databases of mathematical models (Chelliah et al., 2014; King et al., 2015; Peters et al., 2017), but also by pathway databases (Caspi et al., 2015; Mi et al., 2016; Fabregat et al., 2017) as well as reaction databases (Wittig et al., 2017; Placzek et al., 2017). SBML is also used to share models by more generic data management platforms in systems biology (Wolstencroft et al., 2016). The continuous development and reuse of existing models in systems biology requires sound model management strategies. SBML facilitates tasks such as model storage,

version control (Scharm et al., 2015), quality and validity checks. The comparison of model structure or semantic annotations has also led to the development of several methods to quantify model similarities (Henkel et al., 2016) that can then be used to improve the relevance of model searches (Henkel et al., 2010; Schulz et al., 2011). Once model elements can be compared, one can align, combine and merge different models (Krause et al., 2010).

Box 4. Examples of SBML use cases

One of the deepest impacts SBML has on computational systems biology is to facilitate, or even enable, collaborative work. We illustrate here three use cases were using SBML facilitates or enables research projects.

SBML throughout the model life-cycle Encoding a model in SBML permits to use different software tools for the tasks to which they are best suited. A signaling pathway can be designed graphically using CellDesigner (Funahashi et al., 2008). The resulting model can then be semi-automatically annotated using the online tool semanticSBML (Krause et al., 2010). Experimental kinetic information can be retrieved in SBML from the SABIO-Reaction Kinetics database (Wittig et al., 2017). COPASI (Hoops et al., 2006) allows users to estimate parameters and to simulate the model with various algorithms. Other SBML-supporting tools provide capabilities such as identifiability or bifurcation analysis such as Tellurium (Choi et al., 2016) and PySCeS (Olivier et al., 2005). Each step of the modeling, simulation and analysis procedure can be documented using the notes attached to every model component. The model can even be turned into a publishable document using SBML2LaTeX (Dräger et al., 2009).

Pipeline for automated model building The ability to precisely identify model components using semantic annotations facilitates the building of automated pipelines for model generation, using existing models in conjunction with databases of molecular phenotypes or reaction kinetics (Li et al., 2010). Models can also be generated *de novo* from data resources, as shown by the Path2Models project (Büchel et al., 2013). Path2Models produced 143,000 models SBML models (all fully annotated) for over 2,600 organisms using pathway database data. Metabolic pathways were encoded in SBML Core while signalling pathways were encoded with SBML Qual (Chaouiya et al., 2013). Moreover, constraint-based models of genome-scale reconstruction were provided for each organism. Pipelines have now been built that can systematically generate alternative models for different tissue-types (Thiele et al., 2013) and patients (Uhlen et al., 2017), a pivotal stepping-stone towards personalized precision medicine.

Development, sharing, and re-use of genome-scale models Constraint-based modeling approaches such as Flux Balance Analysis and its derivatives permit to use whole-genome reconstructions together with experimental molecular phenotypes in order to predict how mutations or different environments affect metabolic landscapes, and to predict drug targets and biomarkers (Savinell and Palsson, 1992; O'Brien et al., 2015). With the availability of genome-scale metabolic reconstructions (Edwards and Palsson, 1999), the use of genome-scale metabolic flux models has been growing exponentially (Bordbar et al., 2014). A recent development in the field has been the curation by the community of consensus metabolic models, in particular for human (Brunk et al., 2018). Those community efforts rely heavily on SBML for encoding and sharing the models, including annotations, which are crucial to document the curation process and use the reconstructions later, and also visual representation using the Layout (Gauges et al., 2015) and Render (Bergmann et al., 2017) packages. The Flux Balance Constraint package (Olivier and Bergmann, 2018) enables encoding of the information required for model optimization and flux calculation. Unambiguous encoding in SBML has been shown to be key for interpreting the model and precisely computing fluxes (Ebrahim et al., 2015).

Forthcoming challenges

Over the best part of two decades, SBML has supported mathematical modeling in systems biology, structuring the corresponding scientific community and fostering a culture of openness and sharing. This was helped by the emphasis on relatively simple chemical kinetics representations of biomolecular pathways, and the prevalence of ordinary differential equations (Hübner et al., 2011). The structure of the models could then easily be exchanged between SBML-supporting software. Although this modeling approach remains very important, the field is evolving fast, presenting challenges that the community — and SBML — must face.

The first noticeable evolution lies in the size of the published models. While a model in BioModels' "literature" branch contained on average 30 mathematical relationships in 2005, it features about 900 as the time of writing. All types of models tend to grow. However, the main driver of this inflation is currently the increasing use of genome-scale metabolic models (Bordbar et al., 2014). Those models can now be produced semi-automatically for many organisms (Büchel et al., 2013; Magnúsdóttir et al., 2017). Moreover, methods have been developed to use several such models conjointly, either to model several cell-types (Bordbar et al., 2011) or populations of cells (Damiani et al., 2017). It is, therefore, reasonable to anticipate models of ecosystems soon, such as microbiomes and their host. Model size will also become an issue as more models of tissues and organs are exchanged and reused, for instance developed with software like CHASTE (Mirams et al., 2013) and ComputCell3D (Swat et al., 2012). Explorations have shown that using SBML packages such as Comp (Watanabe and Myers, 2014) and Arrays (Watanabe and Myers, 2016) may provide solutions.

Because of the variety of biological phenomena amenable to mathematical modeling, their scales, and their characteristic properties, it is likely that a much broader variety of modeling approaches will become mainstream, beside the numerical or logical simulation of biochemical reactions. Many of those approaches have long been used in modeling biological process but remained confined within communities of specialists. However, methods such as multi-agent and lattice approaches are now needed to represent evolving cell populations, cell migration, and deformation. Modeling the development of tissues and organ function may also require to combine them with reaction-diffusion models, or multi-physics approaches. Statistical modeling will need to complement traditional instance-based systems if we want to take into account patient variability or information coming from single-cell measurements. Not only new modeling approaches are increasingly being used, but the coupling of different modeling approaches within the same simulation experiment is also becoming more frequent. Biomolecular reactions modeled using ODEs, Poisson processes and Flux Balance Analyses were coupled in the first whole-cell model (Karr et al., 2012). At the organ level, liver lobules were modeled using a combination of metabolism and multi-agent models (Schliess et al., 2014). Several approaches mixing modeling of cell mechanical properties and gene regulatory networks or signalling networks were used to study morphogenesis (Tanaka et al., 2015; Delile et al., 2017). The coupling of different approaches can be done within a single hybrid model, simulated by a given software tool, or each model can be simulated using a different software, with a dynamic synchronization at runtime (Mattioni and Le Novère, 2013).

Published and proposed SBML L3 packages already provide support for different modeling approaches (Figure 2). Software exists that allows using conjointly model parts encoded with different L3 packages. For example, the software FlexFlux (Marmiesse et al., 2015) allows the use of a constraint-based metabolic model encoded in SBML FBC with a logic model of regulatory genomics encoded in SBML Qual to run dynamical Flux Balance Analysis. More packages will be needed in the future. We believe that those packages will have to be grassroots, ensuring that they fulfill the needs of users and get prompt software support. Because of the increase in size,

and the diversity of approaches, using monolithic models is becoming increasingly difficult. The Hierarchical Model Composition package already provides a way to encode models in SBML out of separate building blocks or from existing *bona fide* models. A complementary approach is the coupling of "different" models at runtime, for instance, using SED-ML (*vide infra*). Using modular models is natural when encoding multi-scale models (Chew et al., 2014). Such models are easier to maintain, in particular in the case of collaborative development. Finally, a modular approach allows using alternate versions of submodels depending on the need, for instance showing different granularity.

Models are increasingly developed using quantitative experimental measurements. Data has long been used to estimate parameters of moderate size (Mendes and Kell, 1998). Novel approaches are being developed to take care of very large kinetics models (Villaverde et al., 2018). Methods also exist to take into account the variability of experimental results (Liepe et al., 2010). Experimental data is also used to constrain models, such as genome-scale reconstructions. While most methods so far used transcriptomics data (Machado and Herrgård, 2014), for instance, to develop personalized patient models (Uhlen et al., 2017), multi-omics datasets start to be considered (Ebrahim et al., 2016). Use of patient data in conjunction with models can also support drug discovery pipelines (Jerby and Ruppin, 2012). Finally, new technological developments are revolutionizing life sciences and health, and are expected to have a large impact on systems modeling. Improvement in live imaging permit to provide quantities, distributions and velocities of biological entities over time, including single cell and single molecules tracking (Lipkow and Odde, 2008; Griffin et al., 2011). Omics measurement at single-cell resolution also changes the way we parameterize models or validate simulations, increasing the importance of variability, whether intrinsic noise or meaningful distributions.

These emerging needs take place in an evolving landscape of the computational systems biology landscape where the structural model moves from being the principal object of study to become a knowledge aggregator and even integrator. When SBML emerged, it was natural that a model was self-contained. In the future, such a postulate might be challenged. So far, a typical SBML-encoded model came with uniquely defined values parameters, whether initial values for state variables or parameters of mathematical expressions. However, the same "model" can be used with different parametrizations depending on the needs, some of the parameters being distributions, lists or ranges rather than unique values. The semantic annotation of SBML elements has become increasingly important over the years, a bedrock for many of the analyses using SBML-encoded models. It was proposed to move such annotations to separate, linked, files (Neal et al., 2018). Other formats have been developed that can complement SBML, and some consolidations and increased coordination could happen. For instance, graphical representations of models can be encoded in SBGN-ML (Van Iersel et al., 2012). SBML always aimed at defining the model in a declarative way, irrespective of the simulation framework used for its interpretation. SED-ML was later developed to encode what to do with the model, including initialization, simulation and processing of results (Waltemath et al., 2011). Finally, with the rise of gene engineering and synthetic biology, SBML could be combined with other standards such as SBOL (Galdzicki et al., 2014), which allows encoding of genetic designs. For instance, preliminary works showed that SBML can be enriched with SBOL to provide models of DNA components' behavior (Roehner and Myers, 2014). Files in all those formats can now be distributed in single archives, encoded with the OMEX file format (Bergmann et al., 2014). One can thus share all the materials needed to understand and reuse a systems biology project involving modeling and simulation.

We believe that SBML will continue to play a pivotal role in this complex environment. Other formats are often being aligned with SBML, borrowing its object-oriented data structures, which map the complexity of biological systems to distinctly comprehensible components, an also its rich semantic annotation system. Its development also shapes and unites the modeling

community.

Conclusion

SBML was born 18 years ago, as systems biology emerged as a branch of life sciences. The format and supporting software tools were instrumental to the growth of the domain. When modeling and simulation became increasingly used to gain insights into biological phenomena, SBML allowed researchers to publish models in an open, well-supported, and reusable format. Today, scientists can build, manipulate, annotate, store, reuse, publish, and connect models to each other and to basic data sources. In effect, SBML turned models into a kind of data, and moved modeling in biology from an art to an engineering procedure.

The landscape of modeling in systems biology is changing, and SBML will have to change with it. SBML made possible much of the research pursued by the authors of this article, and also helped us structure our thoughts about our models and the biology they represent. We wish SBML continues to be a source of inspiration for many researchers, especially those new to the field. In return, may they help develop the next generation of the language to support more comprehensive, richer, and more diverse models, and expand the reach of systems modeling towards entire cells, organs, and organisms.

1. Acknowledgments

Software packages and libraries are developed under the primary NIH grant (GM070923).

References

Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D. and Chaouiya, C. (2016) Logical modeling and dynamical analysis of cellular networks. *Frontiers in Genetics*, **7**.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T. et al. (2000) Gene Ontology: tool for the unification of biology. *Nature Genetics*, **25**, 25–29.

Ausbrooks, R., Buswell, S., Carlisle, D., Dalmas, S., Devitt, S., Diaz, A., Froumentin, M., Hunter, R., Ion, P., Kohlhase, M., Miner, R., Poppelier, N., Smith, B., Soiffer, N., Sutor, R. and Watt, S. (2003) Mathematical Markup Language (MathML) Version 2.0 (Second Edition). *Tech. rep.* URL: https://www.w3.org/TR/MathML2/.

Bergmann, F. T., Adams, R., Moodie, S., Cooper, J., Glont, M., Golebiewski, M., Hucka, M., Laibe, C., Miller, A. K., Nickerson, D. P. et al. (2014) Combine archive and omex format: one file to share all information to reproduce a modeling project. *BMC Bioinformatics*, **15**, 369.

Bergmann, F. T., Keating, S. T., Gauges, R., Sahle, S. and Wengler, K. (2017) SBML Level 3 Package: Render, Version 1 Release 1. *Tech. rep.* URL: https://identifiers.org/combine.specifications/sbml.level-3.version-1.render.version-1.release-1.

Bergmann, F. T. and Sauro, H. M. (2008) Comparing simulation results of sbml capable simulators. *Bioinformatics*, **24**, 1963–1965.

Blinov, M. L., Faeder, J. R., Goldstein, B. and Hlavacek, W. S. (2004) Bionetgen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, **20**, 3289–3291.

Bordbar, A., Feist, A. M., Usaite-Black, R., Woodcock, J., Palsson, B. Ø. and Famili, I. (2011) A multi-tissue type genome-scale metabolic network for analysis of whole-body systems physiology. *BMC Systems Biology*, **5**, 180.

Bordbar, A., Monk, J. M., King, Z. A. and Palsson, B. Ø. (2014) Constraint-based models predict metabolic and associated cellular functions. *Nature Reviews Genetics*, **15**, 107–20.

Bornstein, B. J., Keating, S. M., Jouraku, A. and Hucka, M. (2008) LibSBML: an API library for SBML. *Bioinformatics*, **24**, 880–881.

Brunk, E., Sahoo, S., Zielinski, D., Altunkaya, A., Dräger, A., Mih, N., Gatto, F., Nilsson, A., Gonzalez, G., Aurich, M., Prlić, A., Sastry, A., Danielsdottir, A., Heinken, A., Noronha, A., Rose, P., Burley, S., Fleming, R., Nielsen, J., Thiele, I. and Palsson, B. (2018) Recon3d enables a three-dimensional view of gene variation in human metabolism. *Nature Biotechnology*, **36**.

Büchel, F., Rodriguez, N., Swainston, N., Wrzodek, C., Czauderna, T., Keller, R., Mittag, F., Schubert, M., Glont, M., Golebiewski, M., van Iersel, M., Keating, S., Rall, M., Wybrow, M., Hermjakob, H., Hucka, M., Kell, D. B., Müller, W., Mendes, P., Zell, A., Chaouiya, C., Saez-Rodriguez, J., Schreiber, F., Laibe, C., Dräger, A. and Le Novère, N. (2013) Path2Models: large-scale generation of computational models from biochemical pathway maps. *BMC Systems Biology*, **7**, 116.

Caspi, R., Billington, R., Ferrer, L., Foerster, H., Fulcher, C. A., Keseler, I. M., Kothari, A., Krummenacker, M., Latendresse, M., Mueller, L. A., Ong, M. Q., Paley, S., Subhraveti, P., Weaver, D. S. and Karp, P. D. (2015) The MetaCyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic Acids Research*, **44**, D471–D480.

Chance, B., Greenstein, D. S., Higgins, J. and Yang, C. (1952) The mechanism of catalase action. ii. electric analog computer studies. *Archives of Biochemistry and Biophysics*, **37**, 322–339.

Chaouiya, C. (2007) Petri net modelling of biological networks. *Briefings in Bioinformatics*, **8**, 210–219.

Chaouiya, C., Bérenguier, D., Keating, S. M., Naldi, A., Van Iersel, M. P., Rodriguez, N., Dräger, A., Büchel, F., Cokelaer, T., Kowal, B., Wicks, B., Gonçalves, E., Dorier, J., Page, M., Monteiro, P. T., von Kamp, A., Xenarios, I., de Jong, H., Hucka, M., Klamt, S., Thieffry, D., Le Novère, N., Saez-Rodriguez, J. and Helikar, T. (2013) Sbml qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. *BMC Systems Biology*, **7**, 135.

Chaouiya, C., Naldi, A. and Thieffry, D. (2012) Logical modelling of gene regulatory networks with ginsim. In *Bacterial Molecular Networks*, 463–479. Springer.

Chelliah, V., Juty, N., Ajmera, I., Ali, R., Dumousseau, M., Glont, M., Hucka, M., Jalowicki, G., Keating, S., Knight-Schrijver, V., Lloret-Villas, A., Natarajan, K., Pettit, J.-B., Rodriguez, N., Schubert, M., Wimalaratne, S., Zhou, Y., Hermjakob, H., Le Novère, N. and Laibe, C. (2014) Biomodels: ten-year anniversary. *Nucleic Acids Research*, **43**, D542–D548.

Chew, Y. H., Wenden, B., Flis, A., Mengin, V., Taylor, J., Davey, C. L., Tindal, C., Thomas, H., Ougham, H. J., De Reffye, P., Stitt, M., Williams, M., Muetzelfeldt, R., Halliday, K. J. and Millar, A. J. (2014) Multiscale digital arabidopsis predicts individual organ and whole-organism growth. *Proceedings of the National Academy of Sciences*, **111**, E4127–E4136.

Choi, K., Medley, J. K., Cannistra, C., Konig, M., Smith, L., Stocking, K. and Sauro, H. M. (2016) Tellurium: a python based modeling and reproducibility platform for systems biology. *bioRxiv*, 054601.

Damiani, C., Di Filippo, M., Pescini, D., Maspero, D., Colombo, R. and Mauri, G. (2017) popfba: tackling intratumour heterogeneity with flux balance analysis. *Bioinformatics*, **33**, i311–i318.

Delile, J., Herrmann, M., Peyriéras, N. and Doursat, R. (2017) A cell-based computational model of early embryogenesis coupling mechanical behaviour and gene regulation. *Nature Communications*, **8**, 13929.

Dräger, A., Planatscher, H., Wouamba, D. M., Schröder, A., Hucka, M., Endler, L., Golebiewski, M., Müller, W. and Zell, A. (2009) SBML2LaTeX: Conversion of SBML files into human-readable reports. *Bioinformatics*, **25**, 1455–1456.

Ebrahim, A., Almaas, E., Bauer, E., Bordbar, A., Burgard, A. P., Chang, R. L., Dräger, A., Famili, I., Feist, A. M., Fleming, R. M. T., Fong, S. S., Hatzimanikatis, V., Herrgård, M. J., Holder, A., Hucka, M., Hyduke, D., Jamshidi, N., Lee, S. Y., Le Novère, N., Lerman, J. A., Lewis, N. E., Ma, D., Mahadevan, R., Maranas, C., Nagarajan, H., Navid, A., Nielsen, J., Nielsen, L. K., Nogales, J., Noronha, A., Pal, C., Palsson, B. O., Papin, J. A., Patil, K. R., Price, N. D., Reed, J. L., Saunders, M., Senger, R. S., Sonnenschein, N., Sun, Y. and Thiele, I. (2015) Do Genome-scale Models Need Exact Solvers or Clearer Standards? *Molecular Systems Biology*, **11**, 831.

Ebrahim, A., Brunk, E., Tan, J., O'brien, E. J., Kim, D., Szubin, R., Lerman, J. A., Lechner, A., Sastry, A., Bordbar, A., Feist, A. M. and Palsson, B. O. (2016) Multi-omic data integration enables discovery of hidden biological regularities. *Nature Communications*, **7**, 13091.

Edwards, J. S. and Palsson, B. O. (1999) Systems properties of the haemophilus influenzaerd metabolic genotype. *Journal of Biological Chemistry*, **274**, 17410–17416.

Fabregat, A., Jupe, S., Matthews, L., Sidiropoulos, K., Gillespie, M., Garapati, P., Haw, R., Jassal, B., Korninger, F., May, B., Milacic, M., Roca, C., Rothfels, K., Sevilla, C., Shamovsky, V., Shorser, S., Varusai, T., Viteri, G., Weiser, J., Wu, G., Stein, L. and Hermjakob, H. (2017) The reactome pathway knowledgebase. *Nucleic Acids Research*, **46**, D649–D655.

Faeder, J. R., Blinov, M. L. and Hlavacek, W. S. (2009) Rule-based modeling of biochemical systems with bionetgen. In *Systems biology*, 113–167. Springer.

Funahashi, A., Matsuoka, Y., Jouraku, A., Morohashi, M., Kikuchi, N. and Kitano, H. (2008) Celldesigner 3.5: a versatile modeling tool for biochemical networks. *Proceedings of the IEEE*, **96**, 1254–1265.

Galdzicki, M., Clancy, K. P., Oberortner, E., Pocock, M., Quinn, J. Y., Rodriguez, C. A., Roehner, N., Wilson, M. L., Adam, L., Anderson, J. C., Bartley, B. A., Beal, J., Chandran, D., Chen, J., Densmore, D., Endy, D., Grünberg, R., Hallinan, J., Hillson, N. J., Johnson, J. D., Kuchinsky, A., Lux, M., Misirli, G., Peccoud, J., Plahar, H. A., Sirin, E., Stan, G.-B. . B., Villalobos, A., Wipat, A., Gennari, J. H., Myers, C. J. and Sauro, H. M. (2014) The synthetic biology open language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nature Biotechnology*, **32**, 545.

Gauges, R., Rost, U., Sahle, S., Wengler, K. and Bergmann, F. T. (2015) The Systems Biology Markup Language (SBML) Level 3 Package: Layout, Version 1 Core. *Journal of Integrative Bioinformatics*, **12**, 267. URL: http://journal.imbio.de/article.php?aid=267.

Gillespie, D. T. (1977) Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, **81**, 2340–2361.

Griffin, E. E., Odde, D. J. and Seydoux, G. (2011) Regulation of the mex-5 gradient by a spatially segregated kinase/phosphatase cycle. *Cell*, **146**, 955–968.

Hastings, J., de Matos, P., Dekker, A., Ennis, M., Harsha, B., Kale, N., Muthukrishnan, V., Owen, G., Turner, S., Williams, M. et al. (2013) The chebi reference database and ontology for biologically relevant chemistry: enhancements for 2013. *Nucleic Acids Research*, **41**, D456–D463.

Henkel, R., Endler, L., Peters, A., Le Novère, N. and Waltemath, D. (2010) Ranked retrieval of computational biology models. *BMC Bioinformatics*, **11**, 423.

Henkel, R., Hoehndorf, R., Kacprowski, T., Knüpfer, C., Liebermeister, W. and Waltemath, D. (2016) Notions of similarity for systems biology models. *Briefings in Bioinformatics*, **19**, 77–88.

Hodgkin, A. L. and Huxley, A. F. (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, **117**, 500–544.

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P. and Kummer, U. (2006) CopasiâĂŤa complex pathway simulator. *Bioinformatics*, **22**, 3067–3074.

Hübner, K., Sahle, S. and Kummer, U. (2011) Applications and trends in systems biology in biochemistry. *FEBS Journal*, **278**, 2767–857.

Hucka, M., Bergmann, F. T., Dräger, A., Hoops, S., Keating, S. M., Le Novère, N., Myers, C. J., Olivier, B. G., Sahle, S., Schaff, J. C., Smith, L. P., Waltemath, D. and Wilkinson, D. J. (2015) Systems Biology Markup Language (SBML) Level 2 Version 5: Structures and Facilities for Model Definitions. *Journal of Integrative Bioinformatics*, **12**, 271. URL: http://journal.imbio.de/article.php?aid=271.

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C. and Kitano, H. (2002) The ERATO Systems Biology Workbench: Enabling interaction and exchange between software tools for computational biology. In *Pacific Symposium on Biocomputing* (eds. R. B. Altman, A. K. Dunker, L. Hunter, K. Lauderdale and T. E. Klein), vol. 7, 450–461. World Scientific Press.

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. . H. S., Hunter, P. J., Juty, N. S., Kasberger, J. L., Kremling, A., Kummer, U., Le Novère, 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. M. and Wang, J. (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524–531.

Hucka, M., Sauro, H. M., Finney, A., Bolouri, H., Doyle, J. and Kitano, H. (2001) The ERATO Systems Biology Workbench: An integrated environment for multiscale and multitheoretic simulations in systems biology. In *Foundations of Systems Biology* (ed. H. Kitano), chap. 6, 125–143. MIT Press.

Hucka, M. and Smith, L. P. (2016) SBML Level 3 package: Groups, Version 1 Release 1. *Journal of Integrative Bioinformatics*, **13**, 8–29.

Ideker, T., Galitski, T. and Hood, L. (2001) A new approach to decoding life: systems biology. *Annual Review of Genomics and Human Genetics*, **2**, 343–372.

Jerby, L. and Ruppin, E. (2012) Predicting drug targets and biomarkers of cancer via genome-scale metabolic modeling. *Clinical Cancer Research*, **18**, 5572–5584.

Juty, N., Le Novère, N. and Laibe, C. (2012) Identifiers.org and MIRIAM registry: community resources to provide persistent identification. *Nucleic Acids Research*, **40**, D580–D586.

Karr, J. R., Sanghvi, J. C., MacKlin, D. N., Gutschow, M. V., Jacobs, J. M., Bolival, B., Assad-Garcia, N., Glass, J. I. and Covert, M. W. (2012) A whole-cell computational model predicts phenotype from genotype. *Cell*, **150**, 389–401.

Kell, D. B. and Mendes, P. (2008) The markup is the model: reasoning about systems biology models in the semantic web era. *Journal of Theoretical Biology*, **252**, 538–543.

King, Z. A., Lu, J., Dräger, A., Miller, P., Federowicz, S., Lerman, J. A., Ebrahim, A., Palsson, B. O. and Lewis, N. E. (2015) Bigg models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Research*, **44**, D515–D522.

Kitano, H. (2000) Perspectives on systems biology. *New Generation Computing*, **18**, 199–216.

Krause, F., Uhlendorf, J., Lubitz, T., Schulz, M., Klipp, E. and Liebermeister, W. (2010) Annotation and merging of sbml models with semanticsbml. *Bioinformatics*, **26**, 421–422.

Le Novère, N. (2015) Quantitative and logic modelling of molecular and gene networks. *Nature Reviews Genetics*, **16**, 146–158.

Le Novère, N., Finney, A., Hucka, M., Bhalla, U. S., Campagne, F., Collado-Vides, J., Crampin, E. J., Halstead, M., Klipp, E., Mendes, P., Nielsen, P., Sauro, H., Shapiro, B. E., Snoep, J. L., Spence, H. D. and Wanner, B. L. (2005) Minimum information requested in the annotation of biochemical models (MIRIAM). *Nature Biotechnology*, **23**, 1509–1015.

Li, P., Dada, J. O., Jameson, D., Spasic, I., Swainston, N., Carroll, K., Dunn, W., Khan, F., Malys, N., Messiha, H. L., Simeonidis, E., Weichart, D., Winder, C., Wishart, J., Broomhead, D. S., Goble, C. A., Gaskell, S. J., Kell, D. B., Westerhoff, H. V., Mendes, P. and Paton, N. W. (2010) Systematic integration of experimental data and models in systems biology. *BMC Bioinformatics*, **11**, 582.

Liepe, J., Barnes, C., Cule, E., Erguler, K., Kirk, P., Toni, T. and Stumpf, M. P. (2010) Abc-sysbio — approximate bayesian computation in python with gpu support. *Bioinformatics*, **26**, 1797–1799.

Lipkow, K. and Odde, D. J. (2008) Model for protein concentration gradients in the cytoplasm. *Cellular and Molecular Bioengineering*, **1**, 84–92.

Machado, D. and Herrgård, M. (2014) Systematic evaluation of methods for integration of transcriptomic data into constraint-based models of metabolism. *PLoS Computational Biology*, **10**, e1003580.

Magnúsdóttir, S., Heinken, A., Kutt, L., Ravcheev, D. A., Bauer, E., Noronha, A., Greenhalgh, K., Jäger, C., Baginska, J., Wilmes, P., Fleming, R. M. T. and Thiele, I. (2017) Generation of genome-scale metabolic reconstructions for 773 members of the human gut microbiota. *Nature Biotechnology*, **35**, 81–89.

Marmiesse, L., Peyraud, R. and Cottret, L. (2015) Flexflux: combining metabolic flux and regulatory network analyses. *BMC Systems Biology*, **9**, 93.

Mattioni, M. and Le Novère, N. (2013) Integration of biochemical and electrical signaling-multiscale model of the medium spiny neuron of the striatum. *PloS One*, **8**, e66811.

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

Mi, H., Huang, X., Muruganujan, A., Tang, H., Mills, C., Kang, D. and Thomas, P. D. (2016) Panther version 11: expanded annotation data from gene ontology and reactome pathways, and data analysis tool enhancements. *Nucleic Acids Research*, **45**, D183–D189.

Mirams, G. R., Arthurs, C. J., Bernabeu, M. O., Bordas, R., Cooper, J., Corrias, A., Davit, Y., Dunn, S.-J., Fletcher, A. G., Harvey, D. G., Marsh, M., Osborne, J., Pathmanathan, P., Pitt-Francis, J., Southern, J., Zemzemi, N. and Gavaghan, D. (2013) Chaste: an open source c++ library for computational physiology and biology. *PLoS Computational Biology*, **9**, e1002970.

Naldi, A., Monteiro, P. T., Müssel, C., for Logical Models, C., Tools, Kestler, H. A., Thieffry, D., Xenarios, I., Saez-Rodriguez, J., Helikar, T. and Chaouiya, C. (2015) Cooperative development of logical modelling standards and tools with colomoto. *Bioinformatics*, **31**, 1154–1159.

Neal, M. L., König, M., Nickerson, D., Mısırlı, G., Kalbasi, R., Dräger, A., Atalag, K., Chelliah, V., Cooling, M., Cook, D. L., Crook, S., de Alba, M., Friedman, S. H., Garny, A., Gennari, J. H., Gleeson, P., Golebiewski, M., Hucka, M., Juty, N., Le Novère, N., Myers, C., Olivier, B. G., Sauro, H. M., Scharm, M., Snoep, J. L., Touré, V., Wipat, A., Wolkenhauer, O. and Waltemath, D. (2018) Harmonizing semantic annotations for computational models in biology. *bioRxiv*. URL: https://www.biorxiv.org/content/early/2018/01/23/246470.

O'Brien, E. J., Monk, J. M. and Palsson, B. Ø. (2015) Using genome-scale models to predict biological capabilities. *Cell*, **161**, 971–987.

Oda, K., Matsuoka, Y., Funahashi, A. and Kitano, H. (2005) A comprehensive pathway map of epidermal growth factor receptor signaling. *Molecular Systems Biology*, **1**, 2005.0010.

Olivier, B. G. and Bergmann, F. T. (2018) SBML Level 3 Package: Flux Balance Constraints version 2. *Journal of Integrative Bioinformatics*, **15**, 20170082.

Olivier, B. G., Rohwer, J. M. and Hofmeyr, J.-H. S. (2005) Modelling cellular systems with pysces. *Bioinformatics*, **21**, 560–561.

Peters, M., Eicher, J. J., van Niekerk, D. D., Waltemath, D. and Snoep, J. L. (2017) The JWS online simulation database. *Bioinformatics*, **33**, 1589–1590.

Placzek, S., Schomburg, I., Chang, A., Jeske, L., Ulbrich, M., Tillack, J. and Schomburg, D. (2017) Brenda in 2017: new perspectives and new tools in brenda. *Nucleic Acids Research*, **45**, D380–D388.

Rodriguez, N., Thomas, A., Watanabe, L., Vazirabad, I. Y., Kofia, V., Gómez, H. F., Mittag, F., Matthes, J., Rudolph, J. D., Wrzodek, F., Netz, E., Diamantikos, A., Eichner, J., Keller, R., Wrzodek, C., Fröhlich, S., Lewis, N. E., Myers, C. J., Le Novère, N., Palsson, B. Ø., Hucka, M. and Dräger, A. (2015) JSBML 1.0: providing a smorgasbord of options to encode systems biology models. *Bioinformatics*, **31**, 3383–3386.

Roehner, N. and Myers, C. J. (2014) A methodology to annotate systems biology markup language models with the synthetic biology open language. *ACS Synthetic Biology*, **3**, 57–66.

Savinell, J. M. and Palsson, B. Ø. (1992) Network analysis of intermediary metabolism using linear optimization. i. development of mathematical formalism. *Journal of Theoretical Biology*, **154**, 421–454.

Scharm, M., Wolkenhauer, O. and Waltemath, D. (2015) An algorithm to detect and communicate the differences in computational models describing biological systems. *Bioinformatics*, **32**, 563–570.

Schliess, F., Hoehme, S., Henkel, S. G., Ghallab, A., Driesch, D., Böttger, J., Guthke, R., Pfaff, M., Hengstler, J. G., Gebhardt, R., HÃd'ussinger, D., Drasdo, D. and Zellmer, S. (2014) Integrated metabolic spatial-temporal model for the prediction of ammonia detoxification during liver damage and regeneration. *Hepatology*, **60**, 2040–2051.

Schulz, M., Krause, F., Le Novere, N., Klipp, E. and Liebermeister, W. (2011) Retrieval, alignment, and clustering of computational models based on semantic annotations. *Molecular Systems Biology*, **7**.

Smith, L. P., Hucka, M., Hoops, S., Finney, A., Ginkel, M., Myers, C. J., Moraru, I. and Liebermeister, W. (2015) SBML Level 3 package: Hierarchical Model Composition, Version 1 Release 3. *Journal of Integrative Bioinformatics*, **12**, 268.

Swat, M. H., Thomas, G. L., Belmonte, J. M., Shirinifard, A., Hmeljak, D. and Glazier, J. A. (2012) Multi-scale modeling of tissues using compucell3d. In *Methods in Cell Biology*, vol. 110, 325–366. Elsevier.

Tanaka, S., Sichau, D. and Iber, D. (2015) Lbibcell: a cell-based simulation environment for morphogenetic problems. *Bioinformatics*, **31**, 2340–2347.

Terfve, C., Cokelaer, T., Henriques, D., MacNamara, A., Goncalves, E., Morris, M. K., van Iersel, M., Lauffenburger, D. A. and Saez-Rodriguez, J. (2012) Cellnoptr: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC Systems Biology*, **6**, 133.

The UniProt Consortium (2017) Uniprot: the universal protein knowledgebase. *Nucleic Acids Research*, **45**, D158–D169.

Thiele, I., Swainston, N., Fleming, R. M., Hoppe, A., Sahoo, S., Aurich, M. K., Haraldsdottir, H., Mo, M. L., Rolfsson, O., Stobbe, M. D. et al. (2013) A community-driven global reconstruction of human metabolism. *Nature Biotechnology*, **31**, 419–425.

Uhlen, M., Zhang, C., Lee, S., Sjöstedt, E., Fagerberg, L., Bidkhori, G., Benfeitas, R., Arif, M., Liu, Z., Edfors, F., Sanli, K., von Feilitzen, K., Oksvold, P., Lundberg, E., Hober, S., Nilsson, P., Mattsson, J., Schwenk, J. M., Brunnström, H., Glimelius, B., SjÃűblom, T., Edqvist, P.-H., Djureinovic, D., Micke, P., Lindskog, C., Mardinoglu, A. and Ponten, F. (2017) A pathology atlas of the human cancer transcriptome. *Science*, **357**, eaan2507.

Van Iersel, M. P., Villéger, A. C., Czauderna, T., Boyd, S. E., Bergmann, F. T., Luna, A., Demir, E., Sorokin, A., Dogrusoz, U., Matsuoka, Y., Funahashi, A., Aladjem, M. I., Mi, H., Moodie, S. L., Kitano, H., Le Novère, N. and Schreiber, F. (2012) Software support for SBGN maps: SBGN-ML and LibSBGN. *Bioinformatics*, **28**, 2016–2021.

Villaverde, A., Froehlich, F., Weindl, D., Hasenauer, J. and Banga, J. R. (2018) Benchmarking optimization methods for parameter estimation in large kinetic models. *BioRxiv*.

W3C HTML Working Group (2002) Xhtml™1.0 the extensible hypertext markup language (second edition). *Tech. rep.* URL: https://www.w3.org/TR/xhtml1/.

Waltemath, D., Adams, R., Bergmann, F. T., Hucka, M., Kolpakov, F., Miller, A. K., Moraru, I. I., Nickerson, D., Sahle, S., Snoep, J. L. and Le Novère, N. (2011) Reproducible computational biology experiments with SED-ML–the simulation experiment description markup language. *BMC Systems Biology*, **5**, 198.

Watanabe, L. H. and Myers, C. J. (2014) Hierarchical stochastic simulation algorithm for SBML models of genetic circuits. *Frontiers in Bioengineering and Biotechnology*, **2**, 55.

— (2016) Efficient analysis of systems biology markup language models of cellular populations using arrays. *ACS Synthetic Biololy*, **5**, 835–841.

Wittig, U., Rey, M., Weidemann, A., Kania, R. and Müller, W. (2017) SABIO-RK: an updated resource for manually curated biochemical reaction kinetics. *Nucleic Acids Research*, **46**, D656–D660.

Wolstencroft, K., Krebs, O., Snoep, J. L., Stanford, N. J., Bacall, F., Golebiewski, M., Kuzyakiv, R., Nguyen, Q., Owen, S., Soiland-Reyes, S., Straszewski, J., van Niekerk, D., Williams, A., Malmström, L., Rinn, B., MÃijller, W. and Goble, C. (2016) Fairdomhub: a repository and collaboration environment for sharing systems biology research. *Nucleic Acids Research*, **45**, D404–D407.

Zhang, F., Angermann, B. R. and Meier-Schellersheim, M. (2013) The simmune modeler visual interface for creating signaling networks based on bi-molecular interactions. *Bioinformatics*, **29**, 1229–1230.

Zhang, F. and Meier-Schellersheim, M. (2018) Sbml level 3 package: Multistate, multicomponent and multicompartment species, version 1, release 1. *Journal of Integrative Bioinformatics*, **15**, 20170077.