frontiers in
BIOENGINEERING AND BIOTECHNOLOGY

REVIEW ARTICLE
published: 08 December 2014
doi: 10.3389/fbioe.2014.00061

Improving collaboration by standardization efforts in systems biology

Andreas Dräger*[1,2]* and Bernhard Ø. Palsson[1]

[1] Systems Biology Research Group, Department of Bioengineering, University of California, San Diego, La Jolla, CA, USA
[2] Cognitive Systems, Center for Bioinformatics Tübingen (ZBIT), Department of Computer Science, University of Tübingen, Tübingen, Germany

Edited by:
Daniel Machado, University of Minho, Portugal

Reviewed by:
Padraig Gleeson, University College London, UK
Taishin Nomura, Osaka University, Japan
Herbert M. Sauro, University of Washington, USA

****Correspondence:***
Andreas Dräger, University of California, Atkinson Hall, Room 2506, 9500 Gilman Drive # 0412, La Jolla, CA 92093-0412, USA
e-mail: andraeger@eng.ucsd.edu

Collaborative genome-scale reconstruction endeavors of metabolic networks would not be possible without a common, standardized formal representation of these systems. The ability to precisely define biological building blocks together with their dynamic behavior has even been considered a prerequisite for upcoming synthetic biology approaches. Driven by the requirements of such ambitious research goals, standardization itself has become an active field of research on nearly all levels of granularity in biology. In addition to the originally envisaged exchange of computational models and tool interoperability, new standards have been suggested for an unambiguous graphical display of biological phenomena, to annotate, archive, as well as to rank models, and to describe execution and the outcomes of simulation experiments. The spectrum now even covers the interaction of entire neurons in the brain, three-dimensional motions, and the description of pharmacometric studies. Thereby, the mathematical description of systems and approaches for their (repeated) simulation are clearly separated from each other and also from their graphical representation. Minimum information definitions constitute guidelines and common operation protocols in order to ensure reproducibility of findings and a unified knowledge representation. Central database infrastructures have been established that provide the scientific community with persistent links from model annotations to online resources. A rich variety of open-source software tools thrives for all data formats, often supporting a multitude of programing languages. Regular meetings and workshops of developers and users lead to continuous improvement and ongoing development of these standardization efforts. This article gives a brief overview about the current state of the growing number of operation protocols, mark-up languages, graphical descriptions, and fundamental software support with relevance to systems biology.

Keywords: model formats, modeling guidelines, ontologies, model databases, network visualization, software support

1. INTRODUCTION

Since its emergence in the 1960s systems biology has always been tightly related to the availability of powerful computational resources. While at the beginning of research in the field and its applications quick and simple script-based solutions were sufficient, the bar for publication and review has been drastically raised (Sauro et al., 2003). It has been realized that individual scripts, which are specific to certain computational environments and that are not very reproducible are of small benefit for the scientific community and progress of the field (Lloyd et al., 2004). The development of standardized data formats, models, and computational methods have paved the way toward the evolution and maturation of systems biology into a main-stream field of research (Macilwain, 2011). Sufficient annotation and metadata of models, experiments, and other data enhance the reproducibility of

Abbreviations: ANSI, American National Standards Institute; API, application programing interface; BRAIN, brain research through advancing innovative neurotechnologies; CAD, computer-aided design; COPASI, complex pathway simulator; CSS, cascading style sheets; DAE, differential-algebraic equation; DIN, Deutsches Institut für Normung; FBA, flux balance analysis; fbc, flux balance constraints; GO, gene ontology; HTML, hyper text mark-up language; IEEE, Institute of Electrical and Electronics Engineers; IETF, internet engineering task force; ISML, *in silico* mark-up language; JSON, JavaScript object notation; KiSAO, kinetic simulation algorithm ontology; LEMS, low entropy model specification; MAMO, mathematical modeling ontology; MIASE, minimum information about a simulation experiment; MIBBI, minimal information for biological and biomedical research; MIRIAM, minimal information required in the annotation of models; NCBI, National Center for Biotechnology Information; NuML, numerical mark-up language; OBO, open biomedical ontologies; ODE, ordinary differential equation; OMEX, open modeling exchange format; OMG, object management group; OSB, open-source brain; OWL, web ontology language; PDE, partial differential equation; PharmML, pharmacometrics mark-up language; PHML, physiological hierarchy mark-up language; RDF, resource description framework; SBGN, systems biology graphical notation; SBGN-ML, systems biology graphical notation mark-up language; SBML, systems biology mark-up language; SBOL, synthetic biology open language; SBRML, systems biology result mark-up language; SBW, systems biology workbench; SED-ML, simulation experiment description mark-up language; SVG, scalable vector graphics; SWIG, simplified wrapper and interface generator; TEDDY, terminology for the description of dynamics; URI, uniform resource identifier; W3C, world wide web consortium; XML, eXtended mark-up language.

results (Wolstencroft et al., 2011). For individual areas of research, different models are required, hence different standards for their encoding. Research in constraint-based modeling (Bordbar et al., 2014) deals with the encoding of the stoichiometric matrix and flux bounds, whereas, dynamic metabolic modeling (Dräger and Planatscher, 2013a) is usually based on building ordinary differential equation systems, model calibration, and parameter estimation (Dräger et al., 2009a; Kronfeld et al., 2009; Dräger and Planatscher, 2013b). Spatial-temporal simulations require encoding three-dimensional geometries and partial differential equation systems (Moraru et al., 2008).

It can hence be observed that the modeling community in systems biology has diversified. One reason for this development is that main parts of funding for these standardization attempts originate from ambitious large-scale projects, each having has different requirements. These efforts include, for example, goal of specifically reconstructing all reactions in specific organisms, such as human or yeast, resulting in giant reaction networks (Duarte et al., 2007; Herrgård et al., 2008; Rolfsson et al., 2011; Thiele et al., 2013) or systematically representing the complete knowledge about biochemical reactions available today (Büchel et al., 2013a). Trans-European projects like SysMO (Booth, 2007) want to comprehensively record and describe dynamic molecular processes in unicellular microorganisms and to present all processes in the form of computerized mathematical models. The German Virtual Liver Network (Holzhütter et al., 2012) aims to mathematically explain all phenomena in the human liver across multiple cell types and levels of organization. The Physiome project attempts to achieve a full quantitative description of all physiological dynamics and functional behaviors of the intact human body (Hunter and Borg, 2003). The US BRAIN (Brain Research through Advancing Innovative Neurotechnologies) Initiative aims to support the development of new technologies for classifying the anatomical constituents for the brain and to allow simultaneous recording from an unprecedented number of neurons simultaneously. The EUs Human Brain Project seeks to develop the infrastructure for creating computational models of brain regions at multiple scales on high-performance computing platforms (Shepherd et al., 1998; Markram et al., 2011; Kandel et al., 2013). Thereby, medical applications become increasingly important (Büchel et al., 2013b; Grillner, 2014).

Common to all these consortia is that with the increasing number of active researchers and collaborators the exchange, reproduction, and accessibility of models, data, and further information in specific online databases play a major role (Brazma et al., 2006; Schellenberger et al., 2010; Wolstencroft et al., 2011; Yu et al., 2011; Chelliah et al., 2013). Just like the documentation of source code, the careful annotation of models and data are also necessary to achieve a fruitful collaboration. The more meta information that is provided, the easier the model can be comprehended, modified, simulated, and analyzed (Waltemath et al., 2013). The use of standard formats is highly recommended for the publication of results even if not required by the prospective journal.

In addition, new fields and areas of application are emerging, for instance, pharmacometric models or synthetic biology (Endler et al., 2009; Galdzicki et al., 2011; Müller and Arndt, 2012). There is therefore no one-size-fits-all solution that would be equally suitable for all fields of research. The standardization community therefore needs to continuously catch up with these developments in the actual modeling community and to reinvent itself over and over again. Recent approaches have suggested to modularize modeling languages by introducing highly specialized *packages* for modeling aspects that can otherwise not be represented in the main data format (Chaouiya et al., 2013).

The structure of how standards are defined has also matured. Brazma et al. (2006) describe that four steps are required for the development of a standard: (i) data and information need to be collected about the domain of interest that are relevant for an unambiguous transfer and interpretation as well as conceptual model design, (ii) the model needs to be formalized, (iii) an exchange format must be defined, and (iv) software support must be implemented. Nearly all modeling formats described in this article now follow this suggestion and are based on a minimum information requirement description (Taylor et al., 2008). These documents define what kind of information has to be stored in a respective model in order to guarantee that the model can be reused and understood by other researchers. In this way, the information requirement and the corresponding modeling standard are decoupled, exchangeable, and independent. The minimum information requirement is usually complemented with a specific ontology, i.e., a hierarchical collection of field-specific terms and their definitions (Courtot et al., 2011). These terms can be associated to model components and descriptions. In addition, elaborate and persistent annotation frameworks have been developed, which allow the modeler to precisely express, what individual model components are and how they are to be understood (Juty et al., 2012, 2013). The development of standards, minimal information requirements, and ontologies needs to be orthogonal to existing respective standards. **Table 1** and **Figure 1** give an overview about the relationship amongst various standards discussed in this article.

The structural representation of the model [for instance, SBML by Hucka et al. (2004) or CellML by Cuellar et al. (2006)], its application and analysis [SED-ML by Waltemath et al. (2011b)], its (graphical) display [SBGN guidelines by Le Novère et al. (2009)], and features should be accurately discriminated and encoded in

Table 1 | Standards with relevance for modeling in systems biology.

	Model	Procedures	Results
Representation formats	BioPAX, CellML, NeuroML, PharmML, SBML (including extension packages), SBGN-ML, SBOL	SED-ML	NuML, SBRML
Graphical display	CellML visualization, SBGN, SBOL visual		
Minimal information requirements	MIRIAM	MIASE	
Mathematical semantics	SBO, MAMO	KiSAO	TEDDY
Biological semantics	MIRIAM	MIRIAM	MIRIAM

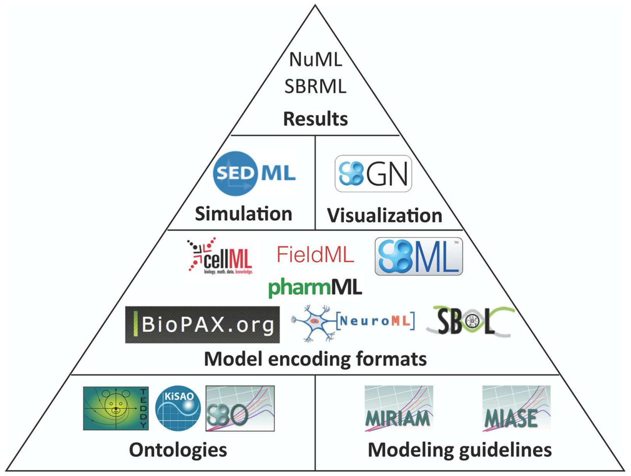


FIGURE 1 | Standards overview. Hierarchically organized controlled vocabularies, so-called *ontologies* and modeling guidelines build the basis for model encoding formats. These formats can refer to terms from ontologies and their organization is in accordance with the modeling guidelines. Recommendations for a visual representation of models as well as the execution of individual models in numerical simulation or optimization are separated from the structural models. Numerical results can be encoded in further standard data formats.

distinct formats. Depending on the concrete modeling format, structural models can also include mathematical formulations, but not their interpretation framework (such as the algorithm to solve the model or the simulation end time). Recently, a new archive format has been proposed in order to link and distribute these independent modeling aspects all together in a single file (Bergmann et al., 2014).

Much effort has also been invested in software support and the creation of infrastructures for diverse standards. For each data format, a specific library has been implemented for reading and writing files as well as for manipulating components of the format in memory (Bornstein et al., 2008; Miller et al., 2010; Demir et al., 2013). Often, language-bindings for diverse programing environments are provided, but sometimes specific libraries have been developed in order to support certain programing languages (Dräger et al., 2011). These parsing libraries help developers to use and exploit the individual standards. Often these libraries provide interfaces to corresponding ontologies and controlled vocabulary annotations (Courtot et al., 2011). However, the interpretation, analysis, drawing, etc. of models cannot be facilitated by these libraries. Higher level software has been implemented to support model building, display, simulation, etc. (Deckard et al., 2006; Keller et al., 2013). Sometimes, this is done in the form of plug-ins to more general frameworks, and often there are diverse stand-alone or web-based tools for various purposes (König et al., 2012; Krause et al., 2013).

When the first XML- or OWL-based exchange formats for models were proposed, developers of existing software tools were often involved, and their individual software was adapted in order to fit the standard. Nowadays, with many standards being well established, software is specifically tailored with respect to the standards. The stringent elaboration and clear distinction between models, purpose, simulation, and annotation can also be a source of inspiration for young researchers who enter the field. In the long-term, using standard formats can lower the expenses for software development because they allow the reuse of existing tools in new applications. Moreover, with the many available tools for standard formats, less research time is needed for the interconversion of tool-specific files, making it much easier to collect information from diverse sources (Demir et al., 2010).

While international and national standardization bodies, such as OMG, W3C, IEEE, ANSI, IETF, DIN, etc., usually approve standards and release specifications, the situation is different in systems biology, where *de facto* standards are established by the scientific community (Brazma et al., 2006). The fast-moving nature and ongoing development of research makes this approach necessary.

However, keeping track of the growing number of model formats and standards for diverse purposes has become more and more difficult. This review article gives a broad overview of a wide range of currently existing modeling standards, formats, and online repositories, and a selection of software solutions for systems biology and related fields of research. The aim of this article is to highlight specific standards, their usability, and application in order to give the reader an up-to-date picture of model definition, encoding, and availability in systems biology.

2. MATERIAL AND METHODS

2.1. MODELING GUIDELINES

Modeling formats give us the syntax of models (Juty et al., 2012). In order to enhance accessibility of data and to facilitate the reuse of models, several modeling guidelines have been proposed, which are discussed in this section. These guidelines are often called "Minimum Information of/for," which should express that without at least this form of information optimal use and reproducibility of results cannot be guaranteed. More information can always be provided on top of the minimal requirements. The guidelines are hence a form of checklists that describe which kind of information to include and often go back to the idea of the MIBBI project (Minimal Information for Biological and Biomedical research) proposed by Taylor et al. (2008). The open biomedical ontologies (OBO) foundry[1] maintains orthogonal (non-overlapping) collections of controlled vocabularies, which provide the semantics for models. The most well-known ontology is probably the gene ontology (GO) by (Ashburner et al., 2000).

2.1.1. Minimum information required in the annotation of models

Reuse of models can be compromised if inconsistent identifier systems are used for individual components. For instance, when merging models, it is necessary to match overlapping components. If a molecule is identified as water in one model and as H_2O in another such a matching is already difficult for automated procedures. To solve such problems, the minimum information required in the annotation of models (MIRIAM) guidelines has been proposed as a general model curation checklist (Le Novère et al., 2005). The MIRIAM registry (Laible and Le Novère, 2007) goes further

[1] http://obofoundry.org

and provides a connection between controlled vocabularies (Courtot et al., 2011) and formats, tools, and databases. Most modeling standards provide mechanisms to attach MIRIAM annotations to their components. These annotations are structured based on a subject-predicate-object scheme. Here, the subject is the identifier of the model element. The predicate is one of several predefined qualifiers, e.g., hasPart or is. The object should be a web resources pointing to an identifiers.org address (Juty et al., 2012, 2013), for instance http://identifiers.org/kegg.compound/C00001 for water. This Uniform Resource Identifier (URI) is therefore composed of the prefix identifiers.org/, the definition of the data collection (in this example kegg.compound) followed by the delimiter and finally the record identifier (here C00001). Using such an identifiers.org address instead of directly pointing to an entry in ChEBI (Brooksbank et al., 2013; Hastings et al., 2013), MetaCyc (Caspi et al., 2014), KEGG (Kanehisa and Goto, 2000), or any other of the more than 30 currently supported data collections has several advantages. Should the original resource location or address schema change, the identifiers.org site will point to the new location. identifiers.org also measures the uptime of mirror servers for identical records and preferably directs to the most reliable mirror.

2.1.2. Minimum information about a simulation experiment

The minimum information about a simulation experiment (MIASE) project (Waltemath et al., 2011a) aims to unambiguously define how to reproduce the results of a model simulation. For stochastic models, the results should be within an acceptable small range from the original results, and for deterministic models, the results should be identical. This requirements checklist also supports the review process of scientific publications. Relevant ontologies (Courtot et al., 2011) for MIASE are the kinetic simulation algorithm ontology (KiSAO) that defines the method to use, the terminology for the description of dynamics (TEDDY), and the mathematical modeling ontology (MAMO).

2.1.3. Ontologies

2.1.3.1. Kinetic simulation algorithm ontology. The KiSAO gathers computational methods that can be used to simulate a model in a certain way (Courtot et al., 2011). It contains, for instance, definitions of several differential equation solvers for numerical calculations. Organizing these algorithms in a hierarchical structure allows tools to automatically select the most similar solver within their collection of implemented methods.

2.1.3.2. SBO. The Systems Biology Ontology is a collection of terms that describe the structure of a model, its components, modeling frameworks, and processes (Courtot et al., 2011). By using terms from this ontology, the semantics of individual parts of a model can be made explicit. This is often of particular importance if elements can participate in processes where they can have multiple roles, such as catalysts or inhibitors.

2.1.3.3. Mathematical modeling ontology. The recently developed ontology (MAMO, see http://bioportal.bioontology.org/ontologies/MAMO) has complemented and refined the *modeling framework* branch of SBO. Both ontologies are intended to cross link each other. While SBO mainly focuses on the entities and parameters in the model and describes the relationships among them, MAMO has been developed in order to precisely define and categorize *types* of mathematical models (e.g., *ODE*) and their characteristics (e.g., *discrete* vs. *continuous*) as well as types of readouts (such as *time-course analysis*) and variables (such as *dependent variable*).

2.1.3.4. Terminology for the description of dynamics. The TEDDY defines a formal way to specify how the numerical results of a dynamic system behave when a simulation experiment is conducted (Knüpfer et al., 2006; Courtot et al., 2011). In this way, a machine-readable representation of such a description can be automatically generated upon simulation and be stored along with the model. When querying a database of numeric results, this terminology can help to find models with a desired behavior, such as ongoing oscillations.

2.2. MODELING FORMATS

Reconstructing computational models based on a textual description in a publication can be difficult, because required information, such as a clear definition of the units of all components, can be lacking, the language might be imprecise or ambiguous, or a combined explanation of simulation procedure and actual model hamper the implementation of the model (Cooling, 2010; Dräger et al., 2010). In cases, where models are distributed in form of source code implemented for a specific run-time environment or programing language, executing these programs can be a challenge because of diverse dependencies to operating systems or required third-party libraries (Lloyd et al., 2004). In this section, we will discuss several formats that encode systems biological models in different ways with the aim to overcome this problem.

2.2.1. Systems biology mark-up language

The Systems Biology Mark-up Language (Finney and Hucka, 2003; Hucka et al., 2003, 2004)[2] is a hierarchical XML-based format consisting of several lists of components, such as compartments (finite spaces), (reactive) species, parameters (constants or variables), reactions with kinetic laws, user-defined functions and rules, events, units, and many more. SBML has been developed as a model exchange format that covers a wide range of modeling approaches used today (Hucka et al., 2004), including dynamic and steady-state metabolic networks as well as gene-regulatory and signaling networks (Lambeck et al., 2010; Vlaic et al., 2013). The term *reaction* should no longer be seen as a strict (bio-)chemical reaction. It is rather a process with inputs and outcomes. Specific annotation with SBO terms and MIRIAM identifiers clarify the purpose of all elements. The reactions implicitly define a differential equation system, whose explicit structure needs to be assembled at simulation time or prior to simulation. The rationale behind this design decision is that the same model can be interpreted in terms of a different modeling framework, such as stochastic simulation, etc.

The libraries libSBML (Bornstein et al., 2008) and JSBML (Dräger et al., 2011) facilitate the implementation of import and

[2] http://sbml.org

export functions of SBML models in customized software solutions. While libSBML provides bindings to a large variety of programing languages based on the wrapper generator SWIG (Beazley, 1996), the JSBML library has been specifically developed for the platform-independent Java™ language. Both libraries strive to attain a high degree of compatibility. Specific API libraries have also been implemented for working with SBML under MATLAB™ (Keating et al., 2006) and Mathematica™ (Shapiro et al., 2004, 2007).

It has been recognized that the interpretation and simulation of SBML models can be quite challenging and that different simulation environments can yield divergent results on identical input files (Bergmann and Sauro, 2008). For this reason, a comprehensive test suite of manually created SBML models has been established including reference results. This test suite can be used as a benchmark test case for simulation routines.

SBML handles the increasing diversification of modeling approaches and community requirements with the development of several specific and orthogonal packages, which can be used in addition or separately from the core format. The following extension packages have already been released: hierarchical model composition (comp) (Smith et al., 2013b), flux balance constraints (fbc) (Orth et al., 2010; Bergmann and Olivier, 2013), three-dimensional arrangement of elements in diagrams (layout) (Gauges et al., 2006), and qualitative relationships (qual) (Chaouiya et al., 2013). Draft specifications are available for the following extensions: arrays, sampling of values from statistical distributions (distrib), dynamic creation and destruction of structures during a simulation (dyn), grouping of elements (groups), entity pools with multiple states and complex composition of species (multi), drawing graphical representations of a model (render), indication of those model elements that are changed by packages (req), and spatial processes and geometries (spatial). For an up-to-date list and more detailed explanation of available extension packages, see http://sbml.org/Community/Wiki.

2.2.2. CellML

The XML-based model storage and exchange format CellML[3] has been developed for the IUPS Physiome project with the aim to facilitate reuse of models or their components in a software-independent manner (Lloyd et al., 2004; Cooling, 2010). CellML eases the creation of new models based on parts of existing models and hence accelerates the cumbersome model building process (Cooling et al., 2010). CellML models contain structural information about the organization of the model (components, connections, and units), mathematical equations (arbitrary MathML) to quantitatively describe biological processes, and metadata that link model components to online resources. An important design feature of CellML allows components and parameters to be shared across models via import statements and well-defined interfaces. This also allows users to structure their models into multiple files, similar as can be done with HTML pages, and increases reusability of individual black-box models, but also requires a strict decoupling of components. CellML uses RDF tags for semantic annotations and allows for hierarchical groupings of components. A set of software tools is available to edit CellML models, including an API implementation (Miller et al., 2010) or the graphical modeling environments OpenCell (Lloyd, 2013) and OpenCOR (Nickerson et al., 2013). CellML can be inter-converted from and to SBML and to the scripting language Antimony (Schilstra et al., 2006; Smith et al., 2013a). The rates of change of all components are explicit in CellML. When adding components or connections to a model, these rates of change would need to be updated. With the help of interfaces modelers can avoid this cumbersome update process (Cooling, 2010).

2.2.3. FieldML

FieldML[4] is an XML-based model interchange standard, which has been developed with a focus on the euHeart and Physiome projects and is currently available in version 0.5 (Britten et al., 2013). The main purpose of the format is to encode geometric models in explicit or implicit mathematical form with respect to biological and medical phenomena with spatial-temporal variation, such as the simulation of power fields and gradients. FieldML focuses on fields over multiple discrete indices and multivariate fields with discrete or continuous variables as well as interpolation functions. With these approaches, it is possible to model muscle contraction as part of cardiac mechanics, blood flows, and other multi-scale processes. Other applications include the modeling of patient-specific clinical images with the help of specific annotations and fitting of models to fields. Similar approaches are also planned for the spatial extension for SBML (Schaff et al., 2013). A powerful C++ API with wrappers for Java, Fortran, and Python as well as a software plug-in for the physiome model repository (PMR) support FieldML and provide several high-level functions for model building and simulation (Yu et al., 2011). Version 0.5 already includes model composition over multiple files and data sources.

2.2.4. BioPAX

The motivation for the creation of the BioPAX[5] format (Biological Pathway Exchange) was the aim to unify the various co-existing pathway encoding formats of numerous online databases (Demir et al., 2010). This format is intended to facilitate the communication between diverse software systems and also serves as a common knowledge representation of pathways. With BioPAX the structure of metabolic, signaling, and gene-regulatory pathways can be encoded, including relationships between elements (such as genes or molecules) as well as diverse states (such as post-translational modifications). A growing number of pathway databases and software tools provide BioPAX files as import or export formats (Shannon et al., 2003; Funahashi et al., 2008; Demir et al., 2010; Kelder et al., 2011) and BioPAX is useful to integrate information from heterogeneous sources, to support visualization, and analyses. The definition of BioPAX is the result of a continuous community effort. The BioPAX language is organized in levels that increasingly add features to the language definition. BioPAX is based on OWL and it is implemented as an ontology. An online

[3] http://www.cellml.org

[4] http://physiomeproject.org/software/fieldml

[5] http://www.biopax.org

validator can be used to check the correctness of BioPAX files. All elements within a BioPAX file can be annotated using controlled vocabularies and MIRIAM (Laible and Le Novère, 2007; Juty et al., 2012). For writing, reading, manipulating, and analyzing the API library Paxtools (Demir et al., 2013)[6] has been created and is freely available. Quantitative relationships and temporal sequences of events do not belong to the objectives of BioPAX. However, since it is also possible to encode qualitative relationships in SBML (Chaouiya et al., 2013), BioPAX can be converted to SBML without loss of information (Büchel et al., 2012).

2.2.5. *NeuroML*

The object-oriented mark-up language NeuroML (Gleeson et al., 2010)[7] has been developed as a standard to specifically encode, share, and store computational models of information transfer in neurosciences (Goddard et al., 2001). The aim of the language is to cover diverse structural levels beginning at individual neuron cell membranes and ranging to entire neural networks. This XML-based language encodes biophysically detailed neuronal and network models including ion channels, synapses, and the anatomical connectivity of neurons and how these elements underlie the complex electrical behavior of the brain (Gleeson et al., 2010). Therefore, from the very beginning, modularity, portability, and clarity were the main language requirements (Goddard et al., 2001). Supporting high-performance simulations and creating software frameworks for neuroinformatics are the aims of the language (Beeman, 2013). To this end, NeuroML 2 has been built on the Low Entropy Model Specification (LEMS) language (Cannon et al., 2014), which hierarchically defines structure and dynamics of a large variety of biological models. For parsing, writing, and manipulating NeuroML and LEMS files, the Python APIs libNeuroML and PyLEMS as well as the Java™ APIs jNeuroML and jLEMS are available (Vella et al., 2014). The original idea to link sub-modules of processes in NeuroML to models encoded in SBML or CellML (Gleeson et al., 2010) has since been further elaborated. The LEMS libraries allow users to import SBML models and can also export SED-ML (Waltemath et al., 2011b) files for reproducible simulation experiments. The main repository for NeuroML is Open-Source Brain (Gleeson et al., 2013).

2.2.6. *ISML and PHML*

The XML-based language ISML (*insilicoML*) allows users to describe biophysiological models that cross multiple scales and levels. This format is fully compatible to CellML 1.0, but incorporates a specific ontology of physiological functions (Asai et al., 2008). A large collection of models in ISML can be obtained from an online database at http://www.physiome.jp. The physiological hierarchy mark-up language (PHML) has been designed as the successor of ISML (Asai et al., 2013). PHML defines each biological or biophysical element as a module, which can be encapsulated and linked through ports. This concept hierarchically structures the language. Furthermore, PHML can integrate SBML models as sub-cellular phenomena (Asai et al., 2012).

2.2.7. *PharmML*

The Pharmacometrics Mark-up Language PharmML (Moodie et al., 2013)[8] belongs to the most recent languages in the family of XML-based standards for biomedical computation and is currently under development. The purpose of this new language is to exchange and store pharmacometric models, which includes studies, trials, simulations, estimation, and exploration. It will support metadata, non-linear mixed effects models, serve as an encoding platform for new approaches and elements, as well as support model-based analysis. The developers want to ensure backwards compatibility with existing relevant standards in order to use existing software tools. Use-case scenarios are, for instance, the kinetics of tumor growth, observation models, or trial design for treatment-dosing-related data.

2.2.8. *Synthetic biology open language*

The Synthetic Biology Open Language[9] also belongs to the latest modeling standards (Galdzicki et al., 2014). This RDF-based format has been designed in a community process in order to facilitate the creation of synthetic biology components by providing an exchange format for software tools. As a specialty, SBOL comes with a specific graphical representation for promoters, their regulators, and many additional genetic structures (see **Figure 2**).

2.3. STANDARDS FOR MODEL SIMULATION PROCEDURES

Defining the structure of a model does not give any information about reproducible simulation experiments. In order to perform the identical simulation of the model as described in a corresponding research article, the exact name of the numerical solving algorithm, step size, error tolerance, etc. must be precisely defined. The purpose of the Simulation Experiment Description Mark-up Language (SED-ML)[10] is to provide a standardized, machine-readable, platform-independent data format for this purpose (Waltemath et al., 2011b). SED-ML follows the MIASE guidelines (Waltemath et al., 2011a) and hence enables users to attach both a model as well as the description of its intended use to a publication, which could also simplify review processes. It therefore contributes to the reproducibility aspect in science, where only stochastic approaches might diverge within a small range from published data. The XML-based language SED-ML is organized in levels and can describe

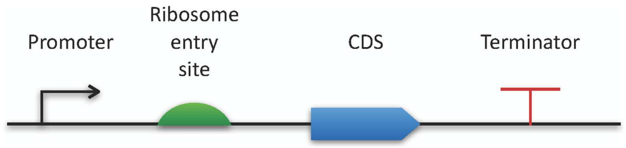


FIGURE 2 | SBOL visual. The horizontal bar represents a DNA molecule to which various features can be visually attached. Here, a few examples are applied for demonstration purposes. A full specification and an exhaustive list of all available symbols can be found online at http://www.sbolstandard.org/visual.

[6] http://www.biopax.org/paxtools/
[7] http://www.neuroml.org
[8] http://www.ddmore.eu
[9] http://www.sbolstandard.org
[10] http://sed-ml.org

multiple simulation experiments within the same file. Language components can be annotated using MIRIAM resources (Laible and Le Novère, 2007). A key idea of SED-ML is not to distribute concrete implementations of simulation procedures, but rather to use ontologies such as KiSAO (Courtot et al., 2011) to refer to the method and its settings. Since this ontology has a hierarchical structure, it is possible to apply related simulation algorithms in case a required method is not implemented in a certain software tool. Structural model changes prior to simulation and post-processing steps of the results (such as converting between amounts and concentration units) as well as the presentation of the output can also be defined (Waltemath et al., 2013). The model can, in principle, be encoded in an arbitrary standardized format and addressed through URI links. SED-ML does not provide an encoding of the simulation results itself, but can be used in combination with numerical mark-up language (NuML) or SBRGML (Dada et al., 2010). An extension to SED-ML has been proposed in order to also support sampling sensitivity analysis simulation experiments (Miller et al., 2012). Some simulation environments have already adopted this young format (Olivier et al., 2005; Myers et al., 2009; Kolpakov et al., 2011; Keller et al., 2013). A workflow editor (SED-ED), API libraries (libSedML, jlibSEDML), and a simplified scripting language (Antimony) are also available (Smith et al., 2009; Adams, 2012).

2.4. GRAPHICAL MODEL REPRESENTATION FORMATS

The visual representation of biochemical pathways has a long tradition. Displays of biological circuit diagrams and reaction pathways can be found in numerous textbooks and a plethora of publications. Databases such as KEGG (Kanehisa and Goto, 2000) or MetaCyc (Caspi et al., 2014) take this up and provide displays of biological networks in their specific layout and style, which follows many traditional aspects. In order to display and draw similar maps, several programs have been developed, for instance, CellDesigner (Funahashi et al., 2008), JDesigner (Sauro et al., 2003), TinkerCell (Chandran et al., 2009), VCell (Resasco et al., 2012), or Cytoscape (Shannon et al., 2003) with its diverse plug-ins (König et al., 2012; Gonçalves et al., 2013). We now discuss recommendations for the display of pathways and standardized data formats for exchanging these maps.

2.4.1. SBGN and SBGN-ML

The myriad of graphical notations that are being used can lead to confusion or ambiguity. The development of a unified and standardized notation has thus become necessary (Le Novère et al., 2009). The Systems Biology Graphical Notation[11] effort aims to make the display of biological networks exchangeable between software tools and at the same time to clearly define the meaning of specific nodes and arcs in such networks in order to ease their interpretation and automated processing. Therefore, the number of graphical symbols is intentionally limited in order to keep the learning curve flat and to create a visually, syntactically, and semantically consistent schema, which is modular in size and complexity (Le Novère et al., 2009). The SBGN neither defines layout (placement and adjustment) nor style (such as line thickness or color) of objects. In order to represent the current needs for such a display, it is organized in levels, so that in the future new versions can be proposed. The specifications of the SBGN are organized in three different languages, each of which has been designed for certain use-case scenarios and has inherent strengths and weaknesses. (i) In process-description diagrams (Kitano et al., 2005; Funahashi et al., 2008), the level of detail is very high and these maps show sequences of processes, which also involve temporal causality (see **Figure 3A**). These maps are well suited for metabolic pathways, but not for the consistent display of the combinatorial complexity of several proteins with many phosphorylation states (van Iersel et al., 2012). (ii) Activity flow charts (van Iersel et al., 2012) are much more abstract and neglect many molecular mechanisms. By design, these maps introduce a certain ambiguity and can hence be used to describe effects whose precise underlying mechanisms are either not know or not relevant (see **Figure 3B**). In this type of diagram, stimulation and inhibition, effects of perturbation, and the activity of components can be displayed. Activity flow charts are thus suitable for the display of causality chains (van Iersel et al., 2012). (iii) The entity-relationship diagrams (Kohn et al., 2006) are particularly useful when the temporal sequence of events does not play the main role, but precise molecular interactions are to be displayed (see **Figure 3C**). These maps are more concise than process-diagrams for protein modifications and interactions, but less capable of representing reactions (van Iersel et al., 2012).

In order to specifically store and exchange SBGN maps in XML files, the Mark-up Language SBGN-ML has been developed (van Iersel et al., 2012). The main requirement for this format is its simplicity, i.e., it should be easy to draw and to interpret. Most significantly, SBGN-ML is not tied to any of the network representation standards. While, this format does not include rendering information, it has been proposed to incorporate a rendering extension, similar as can be done with SBML files. In contrast to the SBML layout extension, this format is focused on the concepts of SBGN only and can be validated against the SBGN specifications. The API library libSBGN[12] facilitates the import and export of SBGN-ML files. The code of libSBGN has been automatically created from an XML Schema Definition file (XSD), which significantly reduces the implementation effort, makes native language implementations in C++ and Java™ possible, and can be used for Schematron validation. A growing number of libSBGN-based software tools support the SBGN-ML format, such as the VANTED (Junker et al., 2006) plug-in SBGN-Ed (Czauderna et al., 2010), the Cytoscape (Shannon et al., 2003) plug-in CySBGN (Gonçalves et al., 2013), the online tool BioGrapher (Krause et al., 2013), the model generator KEGGtranslator (Wrzodek et al., 2013), or the visual editor CellDesigner (Funahashi et al., 2008).

2.4.2. Visualization of CellML

For CellML, a specialized interactive framework has been developed for the display of models (Wimalaratne et al., 2009). This

[11] http://sbgn.org

[12] http://libsbgn.sourceforge.net

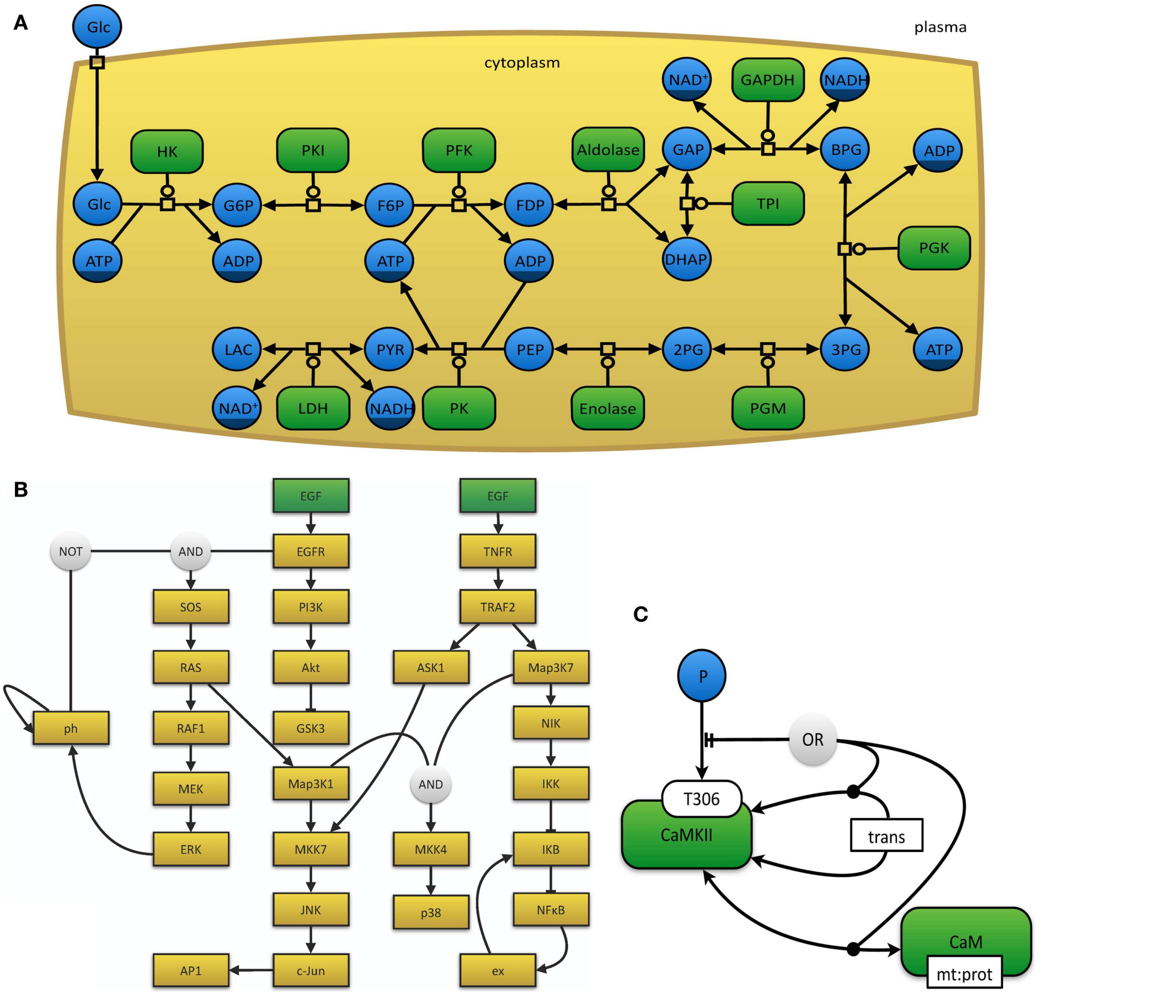


FIGURE 3 | (A) The glycolysis in human erythrocytes, simplified from Dräger (2011). This example network depicts the reaction steps from extracellular glucose to intracellular lactose as a chain of enzyme-catalyzed reactions in SBGN PD notation. Metabolites that occur multiple times in the map, such as ATP or NAD$^+$, have darker clone markers on the bottom. Simple molecules are displayed as circles, whereas, macromolecules appear as rounded rectangles. Reactions are indicated as square process nodes. **(B)** This activity flow diagram displays the interaction of two mammalian signaling pathways that are stimulated by epidermal growth factor (EGF) and tumor necrosis factor alpha (TNFα) and their influence on the nuclear factor κ-light-chain-enhancer of activated B cells (NFκB) and mitogen-activated protein kinases (MAPK) cascades. Adapted from Chaouiya et al. (2013) and generated with the program CellNOpt (Terfve et al., 2012). Here, external stimuli are colored in green. **(C)** This figure displays an example for an entity-relationship diagram, in which Ca^{2+}/calmodulin-dependent protein kinase II (CaMKII) is precluded if it dimerizes or if it binds to the protein calmodulin. Adapted from Le Novere et al. (2011).

framework can either depict the physical model, i.e., the actual components of the CellML format, or the biological interpretation. CellML hence provides its own two-dimensional visual language for both concepts, which can be used in programs to link between the display and the underlying data structure, and also for dynamic image manipulation. For both kinds of displays, a small set of distinct glyphs are defined: entities, processes, and roles. While the physical display tends to be very complex, the biological view is much more straightforward. The developers of the CellML visualization scheme interact with the SBGN team (Wimalaratne et al., 2009). On the longer term, it is intended to combine ideas from SBGN (Le Novère et al., 2009) and the CellML display. Currently, not all concepts of the CellML display can be expressed in SBGN (Wimalaratne et al., 2009).

2.4.3. Layouts in SBML

Layouts can directly be stored in SBML models with the help of the layout extension (Gauges et al., 2006). With this extension, it is possible to attach information about position and size of objects, such as reactive species, compartments, or reaction arcs. Text labels can also be placed. The SBML layout package is based on boundary boxes and defines neither shapes nor colors of objects, but it can be further extended with additional rendering information (Deckard et al., 2006; Shen et al., 2010). Tools such as SBML2LaTeX or SBML2Ti*k*Z (Dräger et al., 2009b; Shen et al., 2010) can interpret layouts stored in this extended SBML to be consistent with SBGN process-diagram maps. In general, these two SBML extensions allow users to store arbitrary forms of network representations. Programs such as KEGGtranslator

(Wrzodek et al., 2011, 2013) use the layout extension to preserve initial layouts from the KEGG database in SBML files. In combination with the SBML extension for qualitative models (Chaouiya et al., 2013), it is also possible to create activity flow networks. In contrast to the SBML layout extension, no standardized way has been proposed to directly store SBGN-ML layouts inside of SBML files. However, the recent COMBINE format (Bergmann et al., 2014) allows users to store files of diverse forms all together within one archive file (see Section 2.6).

2.5. REPRESENTATION OF NUMERICAL RESULTS

In order to store the results of numerical simulation, specific file formats have been proposed. The Systems Biology Result Mark-up Language (SBRML, Dada et al., 2010) has been succeeded by the NuML[13]. This new format has been developed as a standardized exchange and archiving format for the results of numerical methods. This new language has been designed as a format that is usable in various disciplines besides systems biology. The C++ library libNUML can be used for parsing, manipulating, and writing the information of NuML data structures.

2.6. COMBINE FORMAT

The COMBINE format aims to distribute diverse modeling, documentation, and data files together within one single Open Modeling Exchange format (OMEX) file (Bergmann et al., 2014). The format is basically a ZIP archive, i.e., a compressed datatype, which contains an XML-based manifest file and an optional metadata file in RDF format. While the structure of the manifest file is well-defined, there are only recommendations for the metadata file. If present, metadata should at least include information about the author of the OMEX file in form of a vCard and follow the structure proposed by the Dublin Core Metadata Initiative. The manifest file contains structured links to all included files together with a definition URI that describes the filetype. Thus, diverse types of files can be included, even publications, plots, models, graph definitions, etc. Just for the sake of significant data compression, it is already recommended to store models inside of OMEX files (file extension *.omex). Even though the COMBINE archive format belongs to the most recent datatypes of the systems biology community, it is already supported by a number of tools and also the library libCombineArchive for dealing with it (Java™ and C#).

2.7. ONLINE MODEL REPOSITORIES AND DATABASES

One important aspect of model exchange and reusability is the availability and distribution of models that have already been published or that are currently under review. Since a growing number of journals require the online availability of models along with a publication, it is important to be familiar with a number of online resources that are now available. In this section, we will discuss the different aims and features of selected online model repositories, which are summarized in **Table 2**.

2.7.1. BiGG

An important resource for Biochemically, Genetically, and Genomically structured genome-scale metabolic network reconstruction is the BiGG database (Schellenberger et al., 2010). The main focus of this knowledge-base is to facilitate the bottom-up genome-scale reconstruction of metabolic networks. Inclusion of every known reaction of an entire organism constitutes the ultimate goal of BiGG. To this end, it integrates published genome-scale metabolic networks into one resource and applies a standard nomenclature for all of their components. Among these networks are several important model organisms, such as *E. coli* and *H. sapiens*, as well as further main branches of life (Duarte et al., 2007; Feist et al., 2007; Thiele et al., 2013). All models are manually curated and all reactions are atom-balanced. These networks also include gene–protein associations, which can be used to relate the activity of genes via Boolean logic to reactions and hence to perform knock-out or knock-down experiments *in silico*. BiGG offers various options to search, browse, and display networks. Manually curated maps can be downloaded in SVG format for a multitude of pathways. There are often several such maps available for one organism. Various build-in functions (such as decompartmentalization, orphan detection, gap filling, etc.) support the modeling process. With its SBML export function, it provides the basis for further steps in the modeling pipeline, particularly constraint-based analyses by the COBRA platform (Becker et al., 2007; Ebrahim et al., 2013). As the first database specific to constraint-based models, it precedes the SBML extension for fbc,

[13] http://code.google.com/p/numl/

Table 2 | Relevant online databases.

Database	URL	Provides	Comments
BiGG	http://bigg.ucsd.edu	SBML	COBRA models
BioModels	http://www.ebi.ac.uk/biomodels-main/	CellML, SBML, PDF, VCML, and other formats	Main repository for SBML models
JWS	http://jjj.biochem.sun.ac.za/	JWS format, SBML	Online simulation facility
ModelDB	http://senselab.med.yale.edu/modeldb/	Various kinds of model data files	Focus on neuroscience
Open-source brain	http://www.opensourcebrain.org	NeuroML and PyNN	Interactive model development repository
PMR2	http://models.cellml.org	CellML	Project management platform with connection to JWS
SEEK	http://www.sysmo-db.org	Models in diverse formats, publications, and presentations	Focus on collaboration, connection to JWS
WikiPathways	http://www.wikipathways.org	BioPAX, PathVisio, and image formats	Interactive web 2.0 tool for biochemical pathways

but provides COBRA-specific model extensions that can be easily converted (Bornstein et al., 2008).

2.7.2. BioModels database

BioModels database (Chelliah et al., 2013) is an open-source project, whose license model allows free commercial and academic use. Individual authors can submit their models to this database. A team of curators further improves the models, for instance by making the annotations in the model consistent with respect to MIRIAM guidelines (Juty et al., 2012). Large parts of the database content have been imported from collaborative repositories, such as the CellML model repository (Yu et al., 2011). The web interface of BioModels database provides a large variety of services based on embedded tools, e.g., for the simulation or graphical display of models. The main format of BioModels database is SBML, but models can be downloaded in a wide variety of formats, most of which have been automatically converted from the SBML files. It is also possible to obtain an exhaustive model report about each model (Dräger et al., 2009b) that describes the details of each model component in a human-readable way. Since the database was launched in 2005, it has been observed that not only are the number of models significantly increasing, but also their complexity. It now contains a large number of models, each describing the same biological process, but with higher levels of detail. With the growing size of the database the search for a model of interest has become a problem by itself (Schulz et al., 2011). With the help of metadata stored along with each model and the actual content of the models, sophisticated ranking procedures have been designed based on information theory aiming to retrieve models from the database for a given query (Henkel et al., 2010). The metadata include the submission and modification data, the authors of the model, and references. The user can browse through the models based on several characteristics, including the model name, publication identifier, or a GO-based (Ashburner et al., 2000) classification. Besides the curation of models, the main purpose of this repository includes the reproduction of model simulation results as given by the original publication (Waltemath et al., 2013).

2.7.3. CellML physiome model repository 2

The CellML physiome model repository 2 (PMR2) is the most important resource for CellML models at different states of their curation (Yu et al., 2011). It uses a Plone-based model management system that is organized in workspaces. This allows its users to collaboratively develop models based on a version-control system and also facilitates the modular development of models. The models stored in this database cover a large variety of processes, including signal transduction and metabolic pathways, electrophysiological and cell cycle models, immunological models, and models describing muscle contraction or mechanical phenomena. The idea of collaborative model development brings with it one important feature: PMR2 keeps track of a detailed version history of all models. Plug-ins to the system facilitate the presentation of models in various ways and also enable the import and export of diverse modeling formats, including SBML or FieldML besides the native database format CellML. In addition, the plug-in technique makes the database extendable. A search function returns models of all curation states. The main focus of this database is to provide a version-controlled repository for the collaborative model development and presentation of model information, here called *exposures*.

2.7.4. JWS online model repository

Another popular model resource is the JWS Online Model Repository (Snoep and Olivier, 2003). When JWS was launched as the first central model database in 2003 the standards SBML and CellML were still in their early development and not as well established. The repository itself is tightly related to the JWS online simulator (Olivier and Snoep, 2004), a particularly useful resource for educational purposes. Since then, the database has been continuously extended. Its native data format is SBML. Models can be queried based on a list of predefined characteristics (Waltemath et al., 2013), including metadata such as author, publication, organism, or model type as well as a list of categories (for instance, cell cycle or metabolism). The purpose of JWS is to provide a user-friendly online repository of kinetic models of biological systems in combination with an application that facilitates the simulation of these models. The aim of this infrastructure is to ease the review process of papers describing these kinds of models. As a result of its integration into the SEEK platform (Wolstencroft et al., 2011) a large number of collaborative projects use JWS as their default modeling platform.

2.7.5. SEEK platform

The open-source SEEK platform benefits from the ability to offer JWS as its integrated simulation tool to its users. The SEEK platform goes beyond just being a model database. This web-based tool has been designed as a pragmatic data management solution for the exchange of very diverse kinds of data relevant for research in systems biology. Besides mathematical models, it also covers the exchange of experimental data, scientific protocols, and personal information about members of large research consortia (Wolstencroft et al., 2011). It allows its users to record the outcomes of experiments. One of its most important features is the ability to link between data, models, and publications, as well as to tag all uploaded items. This platform has originally been developed for the European SysMO consortium (Booth, 2007), and is also used in several other National and European projects, such as the German Virtual Liver Network (Holzhütter et al., 2012). The preferred modeling data format of SEEK is SBML with MIRIAM annotations.

2.7.6. ModelDB

ModelDB (Migliore et al., 2003; Hines et al., 2004) belongs to the seven databases of SenseLab (NeuronDB, CellPropDB, ModelDB, Olfactory Receptor Database, OdorDB, OdorMapDB, and BrainPharm). SenseLab aims to provide a neural, genomics/genetics, proteomics, and imaging information resource for the neuroscience community and the interested public (Crasto et al., 2007). The database does not explicitly require a standard data format. Instead, authors are welcome to upload their models in arbitrary formats. As a result, the database is very flexible, but model reuse can take extra time to convert the desired model in a format for a particular execution environment (Waltemath et al., 2013).

2.7.7. Open-source brain

Inspired by the open-source movement, the collaboration-oriented open-source brain (OSB) repository has been established (Gleeson et al., 2013). All models in this repository can be commented, debugged, and extended by registered users. This platform therefore complements repositories, such as ModelDB (Hines et al., 2004), which focus on distributing published models, with the aim to drive the advance of models at all stages of its development. An integrated WebGL-based 3D explorer allows users to view cells and networks in NeuroML 2 format within their browser. OSB is well integrated and links to ongoing research projects such as OpenWorm[14].

2.7.8. WikiPathways

The WikiPathways project (Kelder et al., 2011) provides a Web 2.0 wiki-based platform for the online curation of biological pathways. The idea for this platform is that manually curated pathways are of higher quality than automatically created ones. Motivating the scientific community to share knowledge would thus increase the quality of available pathway information. To this end, WikiPathways provides an interactive zoom-able pathway viewer that comes with a pathway diagram description, hyper-links, and detailed information as well as literature references. Users can also annotate the pathways with ontology terms. It is possible to submit private pathway information that is shared later with the public, for instance, as part of the review process, or if current knowledge about certain processes is limited. As a major feature, WikiPathways provides stable hyper-links to all pathways, which is useful in order to use the platform as a reference. Its content can be downloaded in many export formats under the terms of the Creative Commons license. The BioPAX standard (Demir et al., 2010) is thereby its most important format. Internally, it uses GPML, an XML standard that is compatible with many modeling tools, including Cytoscape (Shannon et al., 2003).

3. RESULTS

3.1. INTEROPERABILITY OF STANDARDS

3.1.1. Path2Models

An important driving force for improved interoperability and exchange of diverse data formats and standards was the community project path2models (Büchel et al., 2013a). The aim of this project was to automatically create draft models of biological processes based on the knowledge stored in the databases KEGG (Kanehisa and Goto, 2000), MetaCyc (Caspi et al., 2014), SABIO-RK (Wittig et al., 2014), and PID (Schaefer et al., 2009). The extraction of information from these databases required the development of new algorithms in order to capture a large variety of special cases (Wrzodek et al., 2011, 2013; Büchel et al., 2012) due to the different scope of the source databases. In order to also encode qualitative networks in SBML, the standard needed to be extended (Chaouiya et al., 2013). The draft SBML models had to be quality controlled and enriched with further kinetic information for reactions for which the SABIO-RK database did not yet provide experimentally determined rate laws (Dräger et al., 2008, 2010). Drafts of whole organism models were created by combining individual organism-specific pathway models (Swainston et al., 2011).

The main purposes of the KEGG databases are to provide a comprehensive, textbook-like educational view on the knowledge about a large variety of biological pathways. For modeling purposes, however, the information needs to be presented in a different way (Wrzodek et al., 2013). Reactions cannot be lumped together for the purpose of a better visual presentation, but have to be made explicit. The model must be as specific as possible, i.e., organism-specific variations must be reflected in pathways.

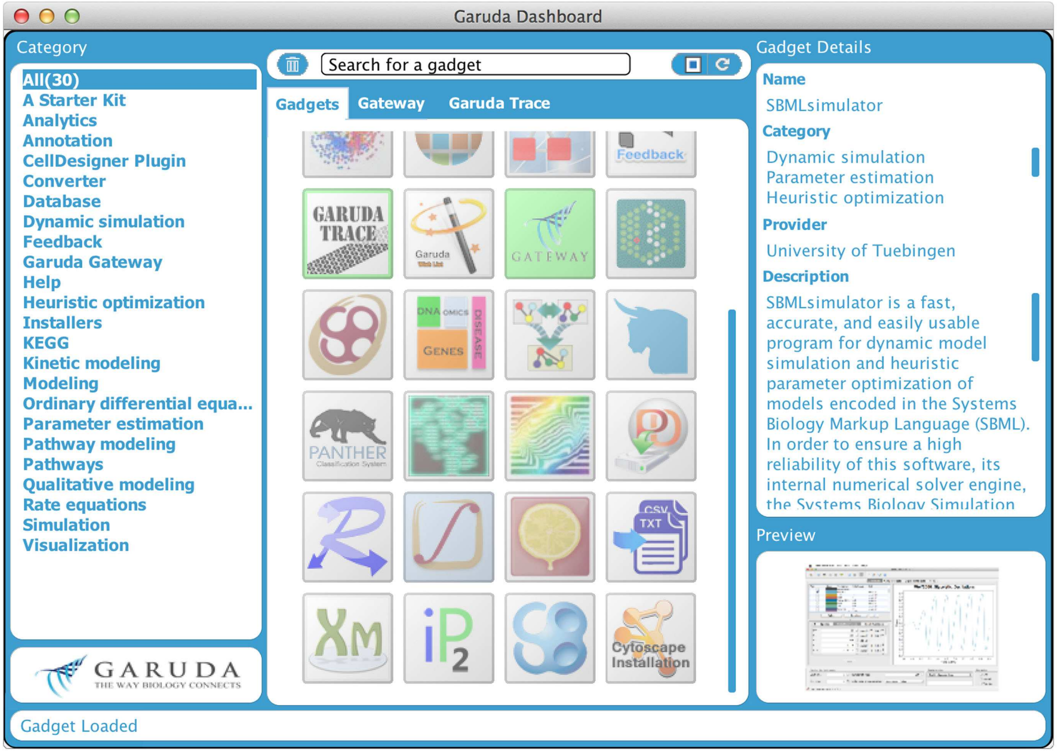
New algorithms also needed to be proposed in order to generate SBGN-ML files directly from KEGG (Czauderna et al., 2013). On the one hand, the manually created pathway maps in KEGG can be much better comprehended by human beholders than automatic layouts. However, in order to obtain an unambiguous representation of knowledge, the initial KEGG layout needs to be modified and subject to several constraints with respect to the esthetics of the result.

Such a large-scale endeavor, which resulted in more than 140,000 pathway maps that are all available from BioModels Database (Chelliah et al., 2013), was only feasible with the help of automatic procedures. Overall, this effort can be seen as a showcase application, which demonstrated the usefulness of data standardization, source code exchange, and software development in a large collaborative community project.

3.1.2. Workbench and workflow approaches

Even though several data storage and exchange formats have been defined and software has been developed to import and export those formats, it is still difficult to work with a large number of different programs and in diverse environments. It can be of particular interest to process intermediate results from one program in another software package or to work with software on different computers with different operating systems. Furthermore, software is often written in diverse programing languages and compiled in diverse environments. Code reuse is still quite limited. All this can hamper building complex analysis pipelines. To address these problems, the systems biology workbench (SBW, Sauro et al., 2003) and the Garuda effort (Ghosh et al., 2011) have been launched. SBW is a software framework for communication between heterogeneous application components. It provides a broker to which each SBW-enabled software needs to register. This broker enables the software to be executed on different machines. Information can be sent from one program to the other through a specific protocol, which provides a fast binary encoded message system. SBW therefore allows programs to use each other's capabilities. In contrast, Garuda is similar to an "App Store" for systems biology (see http://www.garuda-alliance.org/). It provides a common platform, from which diverse applications (gadgets) can be launched (see **Figure 4**). Garuda gadgets can call each other and send their output the next gadget or receive input from other gadgets. A powerful workflow would be to create a model with KEGGtranslator (Wrzodek et al., 2011, 2013), which can forward its result to the rate law generator SBMLsqueezer (Dräger et al., 2008, 2010), which in turn launches SBMLsimulator (Keller et al., 2013)in order to run a simulation and parameter calibration on

[14] http://openworm.org



Garuda Dashboard

Category

All(30)
A Starter Kit
Analytics
Annotation
CellDesigner Plugin
Converter
Database
Dynamic simulation
Feedback
Garuda Gateway
Help
Heuristic optimization
Installers
KEGG
Kinetic modeling
Modeling
Ordinary differential equa...
Parameter estimation
Pathway modeling
Pathways
Qualitative modeling
Rate equations
Simulation
Visualization

Search for a gadget

Gadgets | Gateway | Garuda Trace

Gadget Details

Name
SBMLsimulator

Category
Dynamic simulation
Parameter estimation
Heuristic optimization

Provider
University of Tuebingen

Description
SBMLsimulator is a fast, accurate, and easily usable program for dynamic model simulation and heuristic parameter optimization of models encoded in the Systems Biology Markup Language (SBML). In order to ensure a high reliability of this software, its internal numerical solver engine, the Systems Biology Simulation

Preview

Gadget Loaded

FIGURE 4 | Garuda dashboard. This is the main screen of Garuda. The left column lists several categories that group individual gadgets. The icons in the center column allow users to launch applications with a double click. A detailed description of a gadget is displayed in the right column upon click on an icon.

the resulting model. Garuda provides a nice and easily understandable user interface, the dashboard, from which applications can be launched.

3.2. SOFTWARE SUPPORT

A large variety of software has been developed for many kinds of model building, analysis, drawing, simulation, and format interconversion. In this section, we will only discuss a small number of conceptual categories and particularly important tools. Several reviews specifically focus on available software (e.g., Dandekar et al., 2012; Hamilton and Reed, 2013; Fernández-Castané et al., 2014; Gostner et al., 2014; Koussa et al., 2014; Kramer et al., 2014). **Table 3** gives an overview of selected software. For an up-to-date list and comprehensive information, see, for instance, the dynamic software matrix at http://sysbioapps.dyndns.org/pivot-software-matrix.html.

3.2.1. Visualization and model building

Several tools provide interactive graph-based user interfaces and facilitate import or creation, manipulation, or export of complex pathway structures. Some programs can be extended via plug-ins, e.g., the Biological Network Analyzer BiNA (Gerasch et al., 2014), CellDesigner (Funahashi et al., 2008), or Cytoscape (Shannon et al., 2003). The flexible stand-alone application BiNA (Gerasch et al., 2014) is based on a hierarchical graph concept and provides highly configurable styles for the visualization of regulatory and metabolic network data as well as access to the BN++ pathway data warehouse (Küntzer et al., 2007). The web-modeling tool BioGrapher (Krause et al., 2013) is implemented with HTML5, CSS, and JavaScript and can be used to create SBGN maps. BioGrapher can import several standard formats, including SBML and SBGN-ML, and export SBGN maps in a JSON file format or as images. The VANTED plug-in SBGN-ED supports all three kinds of SBGN maps and is therefore useful for designing and modifying SBGN-ML files (Czauderna et al., 2010). The framework program Cytoscape supports creation, import, and export of SBML and SBGN through plug-ins (König et al., 2012; Gonçalves et al., 2013). The main purpose of the straightforward and user-friendly process-diagram editor CellDesigner is the creation, manipulation, and simulation of SBML models (Matsuoka et al., 2014) with export functions to BioPAX (Mi et al., 2011) and SBGN-ML (van Iersel et al., 2012). CellDesigner can be extended through plug-ins, such as the kinetic law generator SBMLsqueezer (Dräger et al., 2008, 2010). The draft model generator KEGGtranslator (Wrzodek et al., 2011, 2013) automatically downloads contents of the pathway database KEGG (Kanehisa and Goto, 2000) and converts the content to diverse output formats, including SBML with extensions for layout (Gauges et al., 2006) and qual (Chaouiya et al., 2013), SBGN-ML (van Iersel et al., 2012), BioPAX (Demir et al., 2010), and many more. TinkerCell (Chandran et al., 2009) has been developed as a computer-aided design (CAD) tool and provides visual representations for systems biology and synthetic biology. OpenCOR (open-source cross-platform) for working with CellML files can be used through command-line or graphical user interface

Table 3 | Selected relevant software for systems biology.

Program	Main features	Citation
BiNA	Visualization of regulatory and metabolic network data with configurable styles and hierarchical graph concepts; analysis of omics data; data warehouse; plug-in system architecture	Gerasch et al. (2014)
BioGrapher	Web-based tool for creation and editing of SBGN maps with automatic layout algorithms	Krause et al. (2013)
BioUML	Platform for network building, simulation, analysis with full implementation of SBML	Kolpakov et al. (2011)
CellNOpt	Logic-based program for creating and simulating models of signal transduction	Terfve et al. (2012)
Cytoscape	Plug-in-based open-source software platform for visualizing complex networks and their attributes	Shannon et al. (2003)
CellDesigner	Process-diagram editor for gene-regulatory and biochemical networks with plug-in architecture and integrated solvers	Funahashi et al. (2008)
COBRA, COBRApy	Implementations of FBA, gene deletions, flux variability analysis, sampling, and batch simulations for constraint-based models	Schellenberger et al. (2011), Ebrahim et al. (2013)
COPASI	Simulation and analysis of biochemical networks and their dynamics in stochastic and ODE frameworks with support for SBW, parameter estimation, visualization, and several export formats	Hoops et al. (2006)
FASIMU	Command-line based collection of common FBA algorithms for SBML and several kinds of constraints. Its linear programing solvers can be exchanged and numerous constraints be defined	Hoppe et al. (2011)
Flint	An efficient stand-alone solver for PHML and SBML models, which also provides a cloud service	Asai et al. (2013)
GINsim	Simulator for qualitative gene interaction networks with graph-drawing capability, interactive user interface, and support for SBML qual	Gonzalez Gonzalez et al. (2006)
GRN2SBML	Converts the output of network inference procedures to SBML including MIRIAM annotation; access to BioMart central portal; R-package	Vlaic et al. (2013)
iBioSim	Modeling, analysis, design of genetic circuits for systems, and synthetic biology; user-friendly editors for diverse formats; variety of ODE and stochastic simulators; and plotting functions	Myers et al. (2009)
JSim	Building and analysis of quantitative numeric models with focus on physiology and biomedicine; support for ODEs, PDEs, implicit equations, etc.	Butterworth et al. (2014)
libRoad-Runner	C++ library for efficient numerical simulation and analysis of SBML models that provides Python language-bindings, which are integrated into the tellurium environment	Sauro et al. (2013)
libSBMLSim	C-based ODE simulation library for SBML models with explicit and implicit methods, language-bindings, and command-line tool	Takizawa et al. (2013)
Mass-Toolbox	Mathematica framework for kinetic and constraint-based model building and simulation; focus on mass-action kinetics and elementary reaction systems; support for ODE/DAE (incl. delays and events)	Sonnenschein and Palsson (2013)
Module-Master	Identification of *cis*-regulatory modules (CRMs) in sets of co-expressed genes based on transcription factor binding information and multivariate functional relationships between regulators and target genes	Wrzodek et al. (2010)
MOOSE	Multi-scale object-oriented simulation environment for diverse biological systems with a Python scripting interface and support for SBML, NeuroML, GENESIS kkit, and cell.p formats	Dudani et al. (2013)
OpenCOR	Plug-in based cross-platform modeling environment for working with CellML files	Nickerson et al. (2013)
Physio-Designer	Platform for the creation and analysis of PHML models that also allows users to integrate SBML models. It uses Flint as its solver back-end through a cloud service	Asai et al. (2013)
PySCeS	Extendable Python toolbox for time-course simulation, steady-state and stability analysis, metabolic control analysis and many more, support for SBML fbc and SED-ML	Olivier et al. (2005)

(Continued)

Table 3 | Continued

Program	Main features	Citation
SOSlib	C programing library for symbolic and numerical analysis of chemical reaction network models encoded in SBML format	Hindmarsh et al. (2005), Machné et al. (2006)
SBML-simulator	Dynamic model simulation and heuristic parameter optimization of SBML models based on the systems biology simulation core library and EvA2	Kronfeld (2008), Keller et al. (2013)
SBML-squeezer	Context-sensitive generator for kinetic equations of biochemical and gene-regulatory networks with access to SABIO-RK	Dräger et al. (2008, 2010), Dräger (2011)
SBToolbox2	MATLAB™ toolbox with support for SBML, and a large variety of analysis and high-performance simulation functions as well as parameter estimation, sensitivity analyses	Schmidt and Jirstrand (2006), Schmidt (2007)
TinkerCell	Computer-aided design platform for synthetic biology with C and Python API	Chandran et al. (2009)
VANTED	Versatile plug-in based visualization and analysis platform for networks with support for SBGN-ML, sophisticated layout algorithms, and FBA	Junker et al. (2006)
VCell	Modeling and simulation (deterministic and stochastic) of physicochemical and electro-physiological processes with support for irregular spatial distribution of substances in arbitrary geometries	Moraru et al. (2008), Resasco et al. (2012)

(Nickerson et al., 2013). It supports various aspects of modeling, including editing, simulation, and analysis. As a plug-in based program, OpenCOR can be easily extended. One of its most recent plug-ins facilitates the annotation of CellML.

3.2.2. *Constraint-based modeling*

The most important toolboxes for Constraint-Based Reconstruction and Analysis (Bordbar et al., 2014) are the COBRA Toolbox for MATLAB (Schellenberger et al., 2011) and its Python implementation COBRApy (Ebrahim et al., 2013). These toolboxes provide state-of-the-art implementations of flux balance analysis methods, including gene deletions, flux variability analysis, sampling, and batch simulations. Both versions of COBRA incorporate tools to read-in and manipulate constraint-based models, which requires a specific extension of the SBML standard. The Mathematica-based Mass-Toolbox (Sonnenschein and Palsson, 2013)[15] is a complex framework for constraint-based model building and simulation, which can calculate steady-state solutions for complex enzyme reactions and even solve ODE and DAE systems with delays and events. Further important tools for FBA are FASIMU (Hoppe et al., 2011), the VANTED (Junker et al., 2006) plug-in FBA-SimVis (Grafahrend-Belau et al., 2009), and PySCeS (Olivier et al., 2005).

3.2.3. *Dynamic simulation*

The main focus of the Mass-Toolbox (Palsson, 2011; Sonnenschein and Palsson, 2013) is kinetic modeling with a focus on mass-action rate laws and elementary reaction systems. It supports a large variety of analysis methods and high-level plotting commands for phaseportraits, and many more.

The SBToolbox2 (see http://sbtoolbox2.org, Schmidt and Jirstrand, 2006; Schmidt, 2007) provides a powerful and extensible variety of simulation and analysis functions, which smoothly integrate into the MATLAB environment. SBToolbox2 supports SBML and parameter estimation with EvA2 (Kronfeld, 2008).

CellDesigner delivers several third-party tools for interactive model simulation SOSlib (Machné et al., 2006), the Simulation Core Library (Keller et al., 2013), or COPASI (Hoops et al., 2006).

The SBW-enabled complex pathway simulation program COPASI is primarily a stand-alone program, but provides API language-bindings for several programing languages. COPASI can read, write, and understand SBML, but has its own specific modeling language and supports several other export formats. It comprises methods for simulation and analysis of biochemical networks and their dynamics based on ODEs and stochastic systems. Parameter estimation and the visualization of data as well as animated pathways are among its strengths.

The tool SBMLsimulator combines the Simulation Core Library, a comprehensive Java™ API for solving SBML models (Keller et al., 2013) with the optimization framework EvA2 (Kronfeld, 2008) in a self-explanatory user interface and provides a complete implementation of the SBML standard in terms of an ODE framework.

The stand-alone desktop tool BioUML (Kolpakov et al., 2011) is among the few tools that provide a full implementation of the SBML standard in terms of ODE systems and also provides its functions as JavaScript API.

The stand-alone tool iBioSim (Myers et al., 2009) for modeling, analysis, and design of genetic circuits has been developed as an editor and simulator (ODE and stochastic) with applications in systems biology as well as synthetic biology. Besides SBML, it also understands Petri net (LPN) models and has import access to model databases. Experimental data can also be used to infer models in iBioSim.

SOSlib (Machné et al., 2006) is an ODE-based C-API library implementation of SBML that internally uses CVODE (Hindmarsh et al., 2005). The newer C-implementation libSBMLSim (Takizawa et al., 2013) supports even more recent versions of SBML, explicit and implicit integration methods, and bindings to several programing languages. Another alternative is libRoadRunner, a highly performant C++ library for the simulation of

[15] http://opencobra.github.io/MASS-Toolbox/

SBML models, which provides automatically generated language-bindings to Python (Sauro et al., 2013).

The Java-based tool JSim has been designed for building quantitative numeric models as well as the analysis of these models based on given experimental data (Butterworth et al., 2014). It supports ODEs and PDEs, discrete events, and implicit methods. JSim can import and export SBML and import CellML (Smith et al., 2013a).

The Virtual Cell suite VCell is a powerful simulation toolbox for complex biological phenomena (Moraru et al., 2008; Resasco et al., 2012). It includes sophisticated methods for: (i) molecular interactions and transport, (ii) various sub-cellular compartments, (iii) dynamics of membrane potentials, and (iv) arbitrary fluxes and passive cross-membrane transport mechanisms, and supports PDEs in addition to ODEs. It is one of very few tools to incorporate physicochemical and electro-physiological processes and can apply quasi-steady-state approximations to fast reactions. It is also an image processing tool for experimental images.

The simulation environment MOOSE was developed as a reimplementation of the GENESIS neural simulator, and initially used that simulator's model description format. Recently though it has developed support for NeuroML models, and is also capable of dealing with systems biological models (Gleeson et al., 2010; Dudani et al., 2013). New simulation algorithms can be added to MOOSE through a generic framework. It has also been developed with a focus on multi-scale models and simulation in diverse levels of detail (Dudani et al., 2013). For a more comprehensive overview about recent simulation tools with a focus on neuroscience we refer the interested reader to the review by Gleeson (2013).

The stand-alone modeling framework PhysioDesigner (Asai et al., 2013) provides several functions for the creation and analysis of PHML models. SBML models can be incorporated as submodels through PhysioDesigner (Asai et al., 2014), aiming at integrating dynamics at sub-cellular and cellular levels. The simulator Flint can efficiently solve PHML models and provides a cloud service, which allows users to remotely solve their models (Asai et al., 2012). PhysioDesigner uses Flint and submits jobs to this cloud service.

3.2.4. *Regulatory networks*

The inference of regulatory networks is a challenge for many areas of research. The program ModuleMaster (Wrzodek et al., 2010) identifies *cis*-regulatory modules (CRMs) in sets of co-expressed genes based on transcription factor binding information and multivariate functional relationships between regulators and target genes. As an input it uses microarray and clustering experiments and SBML models as output. In order to make the results of network inference procedures such as Net*Gene*rator (Töpfer et al., 2007) reusable in further analysis tools, the program GRN2SBML (Vlaic et al., 2013) has been developed as a converter to SBML. It provides a graphical user interface, access to BioMart Central, and can also be used as an R-package. The program GINsim has been developed for the analysis and simulation of logical models of gene interaction networks (Gonzalez Gonzalez et al., 2006) and has been recently adapted to the SBML qual extension (Chaouiya et al., 2013). The program CellNOpt can be useful for the creation of signal transduction networks based on a logical approach (Terfve et al., 2012), and it also supports SBML qual.

3.3. REGULAR COMMUNITY MEETINGS

Many standards described in this paper are based on community efforts. For this reason, community meetings have been required from their inception. In October 2010, separate workshops were combined in order to better coordinate individual developments and to reduce the necessary amount of traveling for individual researchers. This resulted in two regular annual meetings that brought together the community. The COMBINE (Computational Modeling in Biology Network) is a workshop with scientific presentations, poster sessions, and several break-out sessions, which are used to discuss and coordinate the further development of the "COMBINE Standards" BioPAX, CellML, SBGN, SBML, SBOL, and SED-ML, as well as associated and related standards. The idea of the spring Hackathon on resources for modeling in biology (HARMONY) is to provide room and time for community members to sit down, share code and ideas, program, and discuss. In contrast to the fall event, HARMONY usually has only very few talks and is much more a hands-on practical event, where participants develop new approaches and ideas. For more information about previous meetings see the meeting reports by Le Novère et al. (2011), Waltemath et al. (2014) and the COMBINE homepage[16]. This alternating sequence of complementary meetings leads to a very efficient and progressive development of software and standards.

4. DISCUSSION

In this review article, we have examined diverse modeling standards and data formats that are currently in use within the scientific community together, with databases from where these formats can be obtained. We discussed a selection of useful software packages and modeling approaches for systems biology and related fields. The structuring of individual standards is at present very elaborate: there is usually a modeling, annotation, or documentation recommendation that forms the theoretical basis for a corresponding machine-readable data format and involves specific controlled vocabulary terms for unambiguous specification of individual model components.

Aiming to keep even highly elaborate standards flexible and able to incorporate new findings, the specifications are becoming more and more abstract and modularized. For example, the original *reaction* element in SBML is now seen as a generic *process* whose inputs and outputs no longer strictly have to *represent* substrates and products of biochemical reactions. The idea to develop specific packages for certain needs rather than one monolithic modeling language also follows this trend. The development of all standards involves numerous people, detailed discussions, and careful consideration. This overall procedure ensures that standards mature in an open fashion and allows interested researchers to participate and to contribute to this development. At the same time, it also increases the chance that potential conflicts or inaccuracies can be discovered in early stages of development. With increased use of standards the requirements of the individual format are steadily improved and current limitations are detected and solved. Thanks to the regular meetings and ongoing exchange between the developers of the diverse standards, the individual formats are mutually

[16] http://co.mbine.org

adopting more and more of each other's features. It can therefore be expected that the exchange between different model and pathway representation standards will further increase.

For end-user applications, the goal is that users would no longer have to care about the underlying data format used by a specific software tool. More and more details of the internal structure and organization of underlying formats could be hidden and no detailed knowledge about these formats would be required. Plugins for platforms such as Cytoscape (Shannon et al., 2003) or CellDesigner (Funahashi et al., 2008) can provide complementary functionality for export or import of certain data formats based on a common underlying data structure (König et al., 2012; Gonçalves et al., 2013). The SBW or the Garuda framework provides further ways to increase the interoperability of tools with little effort (Sauro et al., 2003; Ghosh et al., 2011). Many tools could also benefit from the ability of the new COMBINE archive format to bridge separately stored representations or applications of the same model (Bergmann et al., 2014).

The distribution and curation of standardized models, their simulation description, and expected results by centralized databases plays a prominent role. These knowledge bases constitute valuable resources of available information about biological processes and reproducible experiments. They can therefore significantly reduce time and effort needed for the assembly of extended models and create the basis for further research. The ability to easily reproduce new scientific findings with existing simulation workflows facilitates the fast adoption and integration of these findings into new and even further elaborated works. If other researchers are able to run simulations and to comprehend models with minimal effort, it can be expected that these studies will receive higher recognition and lead to more citations compared to distributing models whose outcomes are difficult to reproduce. The distribution of models and data in standard formats amongst their project working groups will not only benefit collaboration partners, but the fine-grained structure of standards for diverse aspects of modeling workflows that is now available can even simplify the review process of scientific papers. If a model is uploaded along with a publication in a standard format, accompanied with a simulation experiment description file and a graphical representation, reviewers can quickly obtain an overview about structure and organization of a model, and even easily check if the findings described in the paper can be reproduced. Thereby, the reviewer can select any numerical tool that supports these data formats and is not restricted to any particular environment.

The development of a standard can be seen as a long-term investment. Unlike in other fields, the community-based bottom-up development of exchange formats is very common in systems biology. Depending on the structure of the field, it can therefore take a long time before the overhead of developing a new standard pays off; on the other hand, standards exist as long as the community has a requirement for them (Brazma et al., 2006). It also seems that the development of standards has become a field of research by itself and is sometimes even seen as the central aspect in modeling (Waltemath et al., 2013). Models and their evaluation are certainly valuable tools for progress in research, but permanently keeping track of all emerging standards can become difficult. The proposed concepts and approaches can only be successful if these are well-known. If standard data formats are developed that are not adopted by the community, the standard will disappear and a simpler solution will gain acceptance. As we go along, new modeling techniques and new finding are established and adopted by the research community (Lerman et al., 2012; O'Brien et al., 2013). Approaches for model encoding and standardization therefore need to continuously evolve with the domain of research that they represent. It is therefore important for the standardization community to continue to closely interact with the modeling community in order to catch up with novel approaches, needs, and requirements. The solutions given to the modeling community must be simple enough in order to be easily adopted, implemented, and applied, but they must also be sophisticated enough in order to capture the complexity of the described systems. Participation of the community in proposing encoding schemes and guideline checklists is essential for the success of the respective standard. Large-scale reconstructions and community projects require data standards and at the same time push their development (Büchel et al., 2013a; Thiele et al., 2013).

While in the past even quick computation in active research required the implementation of some data structures from scratch in customized scripts, the rich variety of software libraries and modeling-specific scripting languages now available drastically simplify these tasks. If an existing software solution cannot be directly applied to solve a specific task, it is at least possible to use standards compliant data structures from the very beginning of a project. Also the quality of available software solutions is progressively increasing. For the distribution of final results, standard formats should be used as the preferred exchange and storage medium in order to ensure reusability and reproducibility of results and findings.

AUTHOR CONTRIBUTIONS

Andreas Dräger developed the conceptual idea for this review and drafted the manuscript. Bernhard Ø. Palsson supervised this work and critically revised the manuscript. Both authors approve the final manuscript.

ACKNOWLEDGMENTS

The authors thank Marc Abrahams, UCSD, for carefully proofreading the manuscript and acknowledge the contribution of Nikolaus Sonnenschein, Novo Nordisk Foundation Center for Biosustainability (CFB). *Funding*: This work was funded by a Marie Curie International Outgoing Fellowship within the European Commission's 7th Framework Program for Research and Technological Development (project AMBiCon, grant number 332020) and a National Institute of Health grant for the continued development of essential SBML software support (NIH, United States, award number GM070923).

REFERENCES

Adams, R. R. (2012). SED-ED, a workflow editor for computational biology experiments written in SED-ML. *Bioinformatics* 28, 1180–1181. doi:10.1093/bioinformatics/bts101

Asai, Y., Abe, T., Oka, H., Okita, M., Hagihara, K.-I., Ghosh, S., et al. (2014). A versatile platform for multilevel modeling of physiological systems: SBML-PHML hybrid modeling and simulation. *Adv Biomed Eng* 3, 50–58. doi:10.1109/EMBC.2013.6610802

Asai, Y., Abe, T., Oka, H., Okita, M., Okuyama, T., Hagihara, K.-I., et al. (2013). "A versatile platform for multilevel modeling of physiological systems: template/instance framework for large-scale modeling and simulation," in *35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Osaka: IEEE.

Asai, Y., Abe, T., Okita, M., Okuyama, T., Yoshioka, N., Yokoyama, S., et al. (2012). "Multilevel modeling of physiological systems and simulation platform: physiodesigner, flint and flint K3 service," in *IEEE/IPSJ 12th International Symposium on Applications and the Internet*. Izmir: IEEE.

Asai, Y., Suzuki, Y., Kido, Y., Oka, H., Heien, E., Nakanishi, M., et al. (2008). Specifications of insilicoML 1.0: a multilevel biophysical model description language. *J. Physiol. Sci.* 58, 447–458. doi:10.2170/physiolsci.RP013308

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., et al. (2000). Gene ontology: tool for the unification of biology. The gene ontology consortium. *Nat. Genet.* 25, 25–29. doi:10.1038/75556

Beazley, D. M. (1996). *SWIG: An Easy to Use Tool for Integrating Scripting Languages with C and C++. Technical Report.* Salt Lake City, UT: Department of Computer Science, University of Utah.

Becker, S. A., Feist, A. M., Mo, M. L., Hannum, G., Palsson, B. Ø., and Herrgård, M. J. (2007). Quantitative prediction of cellular metabolism with constraint-based models: the COBRA toolbox. *Nat. Protoc.* 2, 727–738. doi:10.1038/nprot.2007.99

Beeman, D. (2013). "History of neural simulation software," in *Years of Computational Neuroscience*, , Vol. 9, ed. J. M. Bower (New York: Springer), 33–71.

Bergmann, F. T., Adams, R., Moodie, S., Cooper, J., Glont, M., Golebiewski, M., et al. (2014). *Combine Archive: One File to Share Them All.* Ithaca: Cornell University Library.

Bergmann, F. T., and Olivier, B. G. (2013). "SBML level 3 package: flux balance constraints ('fbc')," in *Technical Report*. Pasadena, CA: Caltech. Available from: http://co.mbine.org/specifications/sbml.level-3.version-1.fbc.version-1.release-1.pdf

Bergmann, F. T., and Sauro, H. M. (2008). Comparing simulation results of SBML capable simulators. *Bioinformatics* 24, 1963–1965. doi:10.1093/bioinformatics/btn319

Booth, I. R. (2007). Sysmo: back to the future. *Nat. Rev. Microbiol.* 5, 566–566. doi:10.1038/nrmicro1719

Bordbar, A., Monk, J. M., King, Z. A., and Palsson, B. Ø. (2014). Constraint-based models predict metabolic and associated cellular functions. *Nat. Rev. Gen.* 15, 107–120. doi:10.1038/nrg3643

Bornstein, B. J., Keating, S. M., Jouraku, A., and Hucka, M. (2008). LibSBML: an API library for SBML. *Bioinformatics* 24, 880–881. doi:10.1093/bioinformatics/btn051

Brazma, A., Krestyaninova, M., and Sarkans, U. (2006). Standards for systems biology. *Nat Rev Genet* 7, 593–605. doi:10.1038/nrg1922

Britten, R. D., Christie, G. R., Little, C., Miller, A. K., Bradley, C., Wu, A., et al. (2013). FieldML, a proposed open standard for the physiome project for mathematical model representation. *Med. Biol. Eng. Comput.* 51, 1191–1207. doi:10.1007/s11517-013-1097-7

Brooksbank, C., Bergman, M. T., Apweiler, R., Birney, E., and Thornton, J. (2013). The European bioinformatics institute's data resources 2014. *Nucleic Acids Res.* 42, D18–D25. doi:10.1093/nar/gkt1206

Büchel, F., Rodriguez, N., Swainston, N., Wrzodek, C., Czauderna, T., Keller, R., et al. (2013a). Large-scale generation of computational models from biochemical pathway maps. *BMC Syst. Biol.* 7:116. doi:10.1186/1752-0509-7-116

Büchel, F., Saliger, S., Dräger, A., Hoffman, S., Wrzodek, C., Zell, A., et al. (2013b). Parkinson's disease: dopaminergic nerve cell model is consistent with experimental finding of increased extracellular transport of α-synuclein. *BMC Neurosci.* 14:136. doi:10.1186/1471-2202-14-136

Büchel, F., Wrzodek, C., Mittag, F., Dräger, A., Eichner, J., Rodriguez, N., et al. (2012). Qualitative translation of relations from BioPAX to SBML qual. *Bioinformatics* 28, 2648–2653. doi:10.1093/bioinformatics/bts508

Butterworth, E., Jardine, B. E., Raymond, G. M., Neal, M. L., and Bassingthwaighte, J. B. (2014). JSim, an open-source modeling system for data analysis. *F1000Res.* 2:288. doi:10.12688/f1000research.2-288.v3

Cannon, R. C., Gleeson, P., Crook, S., Ganapathy, G., Marin, B., Piasini, E., et al. (2014). LEMS: a language for expressing complex biological models in concise and hierarchical form and its use in underpinning NeuroML 2. *Front. Neuroinform.* 8:79. doi:10.3389/fninf.2014.00079

Caspi, R., Altman, T., Billington, R., Dreher, K., Foerster, H., Fulcher, C. A., et al. (2014). The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Res.* 42, D459–D471. doi:10.1093/nar/gkt1103

Chandran, D., Bergmann, F. T., and Sauro, H. M. (2009). TinkerCell: modular CAD tool for synthetic biology. *J. Biol. Eng.* 3, 19. doi:10.1186/1754-1611-3-19

Chaouiya, C., Berenguier, D., Keating, S. M., Naldi, A., van Iersel, M. P., Rodriguez, N., et al. (2013). SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. *BMC Syst. Biol.* 7:135. doi:10.1186/1752-0509-7-135

Chelliah, V., Laibe, C., and Le Novère, N. (2013). "BioModels database: a repository of mathematical models of biological processes," in *In silico Systems Biology*, volume 1021 of *Methods in Molecular Biology*, ed. M. V. Schneider (New York: Springer), 189–199.

Cooling, M. T. (2010). "A primer on modular mass-action modelling with CellML," in *Systems Biology for Signaling Networks*, volume 1 of *Systems Biology*, ed. S. Choi (New York: Springer), 721–750.

Cooling, M. T., Rouilly, V., Misirli, G., Lawson, J., Yu, T., Hallinan, J., et al. (2010). Standard virtual biological parts: a repository of modular modeling components for synthetic biology. *Bioinformatics* 26, 925–931. doi:10.1093/bioinformatics/btq063

Courtot, M., Juty, N., Knüpfer, C., Waltemath, D., Zhukova, A., Dräger, A., et al. (2011). Controlled vocabularies and semantics in systems biology. *Mol. Syst. Biol.* 7, 543. doi:10.1038/msb.2011.77

Crasto, C. J., Marenco, L. N., Liu, N., Morse, T. M., Cheung, K.-H., Lai, P. C., et al. (2007). SenseLab: new developments in disseminating neuroscience information. *Brief. Bioinformatics* 8, 150–162. doi:10.1093/bib/bbm018

Cuellar, A., Nielsen, P., Halstead, M., Bullivant, D., Nickerson, D., Hedley, W., et al. (2006). *CellML 1.1 Specification. Technical report.* Auckland, NZ: Bioengineering Institute, University of Auckland.

Czauderna, T., Klukas, C., and Schreiber, F. (2010). Editing, validating and translating of SBGN maps. *Bioinformatics* 26, 2340–2341. doi:10.1093/bioinformatics/btq407

Czauderna, T., Wybrow, M., Marriott, K., and Schreiber, F. (2013). Conversion of KEGG metabolic pathways to SBGN maps including automatic layout. *BMC Bioinformatics* 14:250. doi:10.1186/1471-2105-14-250

Dada, J. O., Spasic, I., Paton, N. W., and Mendes, P. (2010). SBRML: a markup language for associating systems biology data with models. *Bioinformatics* 26, 932–938. doi:10.1093/bioinformatics/btq069

Dandekar, T., Fieselmann, A., Majeed, S., and Ahmed, Z. (2012). Software applications toward quantitative metabolic flux analysis and modeling. *Brief. Bioinformatics* 15, 91–107. doi:10.1093/bib/bbs065

Deckard, A., Bergmann, F. T., and Sauro, H. M. (2006). Supporting the SBML layout extension. *Bioinformatics* 22, 2966–2967. doi:10.1093/bioinformatics/btl520

Demir, E., Babur, Ö, Rodchenkov, I., Aksoy, B. A., Fukuda, K. I., Gross, B., et al. (2013). Using biological pathway data with paxtools. *PLoS Comput. Biol.* 9:e1003194. doi:10.1371/journal.pcbi.1003194

Demir, E., Cary, M. P., Paley, S., Fukuda, K., Lemer, C., Vastrik, I., et al. (2010). The BioPAX community standard for pathway data sharing. *Nat. Biotechnol.* 28, 935–942. doi:10.1038/nbt.1666

Dräger, A. (2011). *Computational Modeling of Biochemical Networks.* München: Eberhard Karls University of Tübingen.

Dräger, A., Hassis, N., Supper, J., Schröder, A., and Zell, A. (2008). SBMLsqueezer: a CellDesigner plug-in to generate kinetic rate equations for biochemical networks. *BMC Syst. Biol.* 2:39. doi:10.1186/1752-0509-2-39

Dräger, A., Kronfeld, M., Ziller, M. J., Supper, J., Planatscher, H., Magnus, J. B., et al. (2009a). Modeling metabolic networks in *C. glutamicum*: a comparison of rate laws in combination with various parameter optimization strategies. *BMC Syst. Biol.* 3:5. doi:10.1186/1752-0509-3-5

Dräger, A., Planatscher, H., Wouamba, D. M., Schröder, A., Hucka, M., Endler, L., et al. (2009b). SBML2$\LaTeX$: conversion of SBML files into human-readable reports. *Bioinformatics* 25, 1455–1456. doi:10.1093/bioinformatics/btp170

Dräger, A., and Planatscher, H. (2013a). *Encyclopedia of Systems Biology*, Chapter Metabolic Networks. New York, Heidelberg, Dordrecht, London: Springer-Verlag, 1249–1251.

Dräger, A., and Planatscher, H. (2013b). *Encyclopedia of Systems Biology*, Chapter Parameter Estimation, Metabolic Network Modeling. New York, Heidelberg, Dordrecht, London: Springer-Verlag, 1627–1631.

Dräger, A., Rodriguez, N., Dumousseau, M., Drr, A., Wrzodek, C., Le Novère, N., et al. (2011). JSBML: a flexible Java library for working with SBML. *Bioinformatics* 27, 2167–2168. doi:10.1093/bioinformatics/btr361

Dräger, A., Schröder, A., and Zell, A. (2010). *Systems Biology for Signaling Networks*, volume 2, chapter Automating Mathematical Modeling of Biochemical Reaction Networks. New York: Springer.

Duarte, N. C., Becker, S. A., Jamshidi, N., Thiele, I., Mo, M. L., Vo, T. D., et al. (2007). Global reconstruction of the human metabolic network based on genomic and bibliomic data. *Proc. Natl. Acad. Sci. U. S. A.* 104, 1777–1782. doi:10.1073/pnas.0610772104

Dudani, N., Bhalla, U. S., and Ray, S. (2013). MOOSE, the multiscale object-oriented simulation environment. *Encyclopedia of Computational Neuroscience* 1–4. doi:10.3389/neuro.11.006.2008

Ebrahim, A., Lerman, J. A., Palsson, B. Ø., and Hyduke, D. R. (2013). COBRApy: constraints-based reconstruction and analysis for python. *BMC Syst. Biol.* 7:74. doi:10.1186/1752-0509-7-74

Endler, L., Rodriguez, N., Juty, N., Chelliah, V., Laibe, C., Li, C., et al. (2009). Designing and encoding models for synthetic biology. *J. R. Soc. Interface* 6(Suppl. 4), S405–S417. doi:10.1098/rsif.2009.0035.focus

Feist, A. M., Henry, C. S., Reed, J. L., Krummenacker, M., Joyce, A. R., Karp, P. D., et al. (2007). A genome-scale metabolic reconstruction for escherichia coli k-12 mg1655 that accounts for 1260 orfs and thermodynamic information. *Mol. Syst. Biol.* 3, 121. doi:10.1038/msb4100155

Fernández-Castané, A., Fehér, T., Carbonell, P., Pauthenier, C., and Faulon, J.-L. (2014). Computer-aided design for metabolic engineering. *J. Biotechnol.* doi:10.1016/j.jbiotec.2014.03.029

Finney, A., and Hucka, M. (2003). Systems biology markup language: level 2 and beyond. *Biochem. Soc. Trans.* 31(Pt 6), 1472–1473. doi:10.1042/BST0311472

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

Galdzicki, M., Clancy, K. P., Oberortner, E., Pocock, M., Quinn, J. Y., Rodriguez, C. A., et al. (2014). The synthetic biology open language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nat. Biotechnol.* 32, 545–550. doi:10.1038/nbt.2891

Galdzicki, M., Rodriguez, C., Chandran, D., Sauro, H. M., and Gennari, J. H. (2011). Standard biological parts knowledgebase. *PLoS ONE* 6:e17005. doi:10.1371/journal.pone.0017005

Gauges, R., Rost, U., Sahle, S., and Wegner, K. (2006). A model diagram layout extension for SBML. *Bioinformatics* 22, 1879–1885. doi:10.1093/bioinformatics/btl195

Gerasch, A., Faber, D., Küntzer, J., Niermann, P., Kohlbacher, O., Lenhof, H.-P., et al. (2014). Bina: a visual analytics tool for biological network data. *PLoS ONE* 9:e87397. doi:10.1371/journal.pone.0087397

Ghosh, S., Matsuoka, Y., Asai, Y., Hsin, K.-Y., and Kitano, H. (2011). Software for systems biology: from tools to integrated platforms. *Nat Rev Genet* 12, 821–832. doi:10.1038/nrg3096

Gleeson, P. (2013). "Software tools for modelling in computational neuroscience: overview," in *Encyclopedia of Computational Neuroscience*, eds D. Jaeger and R. Jung (New York: Springer), 1–4. doi:10.1007/978-1-4614-7320-6_93-1

Gleeson, P., Crook, S., Cannon, R. C., Hines, M. L., Billings, G. O., Farinella, M., et al. (2010). NeuroML: a language for describing data driven models of neurons and networks with a high degree of biological detail. *PLoS Comput. Biol.* 6:e1000815. doi:10.1371/journal.pcbi.1000815

Gleeson, P., Silver, R. A., and Cantarelli, M. (2013). "Open source brain", in *Encyclopedia of Computational Neuroscience*, eds D. Jaeger and R. Jung (New York: Springer), 1–3. doi:10.1007/978-1-4614-7320-6_595-2

Goddard, N. H., Hucka, M., Howell, F., Cornelis, H., Shankar, K., and Beeman, D. (2001). Towards NeuroML: model description methods for collaborative modelling in neuroscience. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 356, 1209–1228. doi:10.1098/rstb.2001.0910

Gonçalves, E., van Iersel, M., and Saez-Rodriguez, J. (2013). CySBGN: a cytoscape plug-in to integrate SBGN maps. *BMC Bioinformatics* 14:17. doi:10.1186/1471-2105-14-17

Gonzalez Gonzalez, A., Naldi, A., Sánchez, L., Thieffry, D., and Chaouiya, C. (2006). GINsim: a software suite for the qualitative modelling, simulation and analysis of regulatory networks. *BioSystems* 84, 91–100. doi:10.1016/j.biosystems.2005.10.003

Gostner, R., Baldacci, B., Morine, M. J., and Priami, C. (2014). Graphical modeling tools for systems biology. *ACM Comput. Surv.* 47, 1–21. doi:10.1145/2633461

Grafahrend-Belau, E., Klukas, C., Junker, B. H., and Schreiber, F. (2009). FBA-SimVis: interactive visualization of constraint-based metabolic models. *Bioinformatics* 25, 2755–2757. doi:10.1093/bioinformatics/btp408

Grillner, S. (2014). Megascience efforts and the brain. *Neuron* 82, 1209–1211. doi:10.1016/j.neuron.2014.05.045

Hamilton, J. J., and Reed, J. L. (2013). Software platforms to facilitate reconstructing genome-scale metabolic networks. *Environ. Microbiol.* 16, 49–59. doi:10.1111/1462-2920.12312

Hastings, J., de Matos, P., Dekker, A., Ennis, M., Harsha, B., Kale, N., et al. (2013). The ChEBI reference database and ontology for biologically relevant chemistry: enhancements for 2013. *Nucleic Acids Res.* 41, D456–D463. doi:10.1093/nar/gks1146

Henkel, R., Endler, L., Peters, A., Le Novre, N., and Waltemath, D. (2010). Ranked retrieval of computational biology models. *BMC Bioinformatics* 11:423. doi:10.1186/1471-2105-11-423

Herrgård, M. J., Swainston, N., Dobson, P., Dunn, W. B., Arga, K. Y., Arvas, M., et al. (2008). A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology. *Nat. Biotechnol.* 26, 1155–1160. doi:10.1038/nbt1492

Hindmarsh, A. C., Brown, P. N., Grant, K. E., Lee, S. L., Serban, R., Shumaker, D. E., et al. (2005). SUNDIALS: suite of nonlinear and differential/algebraic equation solvers. *ACM T Math Software* 31, 363–396. doi:10.1145/1089014.1089020

Hines, M. L., Morse, T., Migliore, M., Carnevale, N. T., and Hines, M. L. (2004). ModelDB: a database to support computational neuroscience. *J. Comput. Neurosci.* 17, 7–11. doi:10.1023/B:JCNS.0000023869.22017.2e

Holzhütter, H.-G., Drasdo, D., Preusser, T., Lippert, J., and Henney, A. M. (2012). The virtual liver: a multidisciplinary, multilevel challenge for systems biology. *Wiley Interdiscip Rev Syst Biol Med* 4, 221–235. doi:10.1002/wsbm.1158

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., et al. (2006). Copasi – a complex pathway simulator. *Bioinformatics* 22, 3067–3074. doi:10.1093/bioinformatics/btl485

Hoppe, A., Hoffmann, S., Gerasch, A., Gille, C., and Holzhütter, H.-G. (2011). FASIMU: flexible software for flux-balance computation series in large metabolic networks. *BMC Bioinformatics* 12:28. doi:10.1186/1471-2105-12-28

Hucka, M., Finney, A., Bornstein, B. J., Keating, S. M., Shapiro, B. E., Matthews, J., et al. (2004). Evolving a lingua franca and associated software infrastructure for computational systems biology: the systems biology markup language (SBML) project. *Syst Biol (Stevenage)* 1, 41–53. doi:10.1049/sb:20045008

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., et al. (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19, 524–531. doi:10.1093/bioinformatics/btg015

Hunter, P. J., and Borg, T. K. (2003). Integration from proteins to organs: the physiome project. *Nat. Rev. Mol. Cell Biol.* 4, 237–243. doi:10.1038/nrm1054

Junker, B. H., Klukas, C., and Schreiber, F. (2006). VANTED: a system for advanced data analysis and visualization in the context of biological networks. *BMC Bioinformatics* 7:219. doi:10.1186/1471-2105-7-219

Juty, N., Laibe, C., and Le Novère, N. (2013). "Controlled annotations for systems biology," in *In silico Systems Biology*, Vol. 1021, ed. M. V. Schneider (New York: Springer), 227–245.

Juty, N., Le Novère, N., and Laibe, C. (2012). Identifiers.org and MIRIAM registry: community resources to provide persistent identification. *Nucleic Acids Res.* 40, D580–D586. doi:10.1093/nar/gkr1097

Kandel, E. R., Markram, H., Matthews, P. M., Yuste, R., and Koch, C. (2013). Neuroscience thinks big (and collaboratively). *Nat. Rev. Neurosci.* 14, 659–664. doi:10.1038/nrn3578

Kanehisa, M., and Goto, S. (2000). KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* 28, 27–30. doi:10.1093/nar/28.1.27

Keating, S. M., Bornstein, B. J., Finney, A., and Hucka, M. (2006). SBMLToolbox: an SBML toolbox for MATLAB users. *Bioinformatics* 22, 1275–1277. doi:10.1093/bioinformatics/btl111

Kelder, T., van Iersel, M. P., Hanspers, K., Kutmon, M., Conklin, B. R., Evelo, C. T., et al. (2011). WikiPathways: building research communities on biological pathways. *Nucleic Acids Res.* 40, D1301–D1307. doi:10.1093/nar/gkr1074

Keller, R., Dörr, A., Tabira, A., Funahashi, A., Ziller, M. J., Adams, R., et al. (2013). The systems biology simulation core algorithm. *BMC Syst. Biol.* 7:55. doi:10.1186/1752-0509-7-55

Kitano, H., Funahashi, A., Matsuoka, Y., and Oda, K. (2005). Using process diagrams for the graphical representation of biological networks. *Nat. Biotechnol.* 23, 961–966. doi:10.1038/nbt1111

Knüpfer, C., Beckstein, C., and Dittrich, P. (2006). "Towards a semantic description of biomodels: meaning facets – a case study," in *Proceedings of the Second International Symposium on Semantic Mining in Biomedicine (SMBM 2006)*, eds S. Ananiadou, S. Pyysalo, D. Rebholz-Schuhmann, F. Rinaldi, and T. Salakoski (Jena: CEUR-WS), 97–100.

Kohn, K. W., Aladjem, M. I., Weinstein, J. N., and Pommier, Y. (2006). Molecular interaction maps of bioregulatory networks: a general rubric for systems biology. *Mol. Biol. Cell* 17, 1–13. doi:10.1091/mbc.E05-09-0824

Kolpakov, F. A., Tolstykh, N. I., Valeev, T. F., Kiselev, I. N., Kutumova, E. O., Ryabova, A., et al. (2011). "BioUML-open source plug-in based platform for bioinformatics: invitation to collaboration," in *Moscow Conference on Computational Molecular Biology* (Moskow: Department of Bioengineering and Bioinformatics of MV Lomonosov Moscow State University), 172–173.

König, M., Dräger, A., and Holzhütter, H.-G. (2012). CySBML: a cytoscape plugin for SBML. *Bioinformatics* 28, 2402–2403. doi:10.1093/bioinformatics/bts432

Koussa, J., Chaiboonchoe, A., and Salehi-Ashtiani, K. (2014). Computational approaches for microalgal biofuel optimization: a review. *Biomed Res. Int.* 2014, 1–12. doi:10.1155/2014/649453

Kramer, F., Bayerlová, M., and Beißbarth, T. (2014). R-based software for the integration of pathway data into bioinformatic algorithms. *Biology* 3, 85–100. doi:10.3390/biology3010085

Krause, F., Schulz, M., Ripkens, B., Flöttmann, M., Krantz, M., Klipp, E., et al. (2013). Biographer: web-based editing and rendering of SBGN compliant biochemical networks. *Bioinformatics* 29, 1467–1468. doi:10.1093/bioinformatics/btt159

Kronfeld, M. (2008). *EvA2 Short Documentation.* Tübingen: University of Tübingen, Department of Computer Architecture.

Kronfeld, M., Dräger, A., Aschoff, M., and Zell, A. (2009). "On the benefits of multimodal optimization for metabolic network modeling," in *German Conference on Bioinformatics (GCB 2009)*, Volume P-157 of *Lecture Notes in Informatics*, eds I. Grosse, S. Neumann, S. Posch, F. Schreiber, and P. Stadler (Halle: German Informatics society), 191–200.

Küntzer, J., Backes, C., Blum, T., Gerasch, A., Kaufmann, M., Kohlbacher, O., et al. (2007). Bndb – the biochemical network database. *BMC Bioinformatics* 8:367. doi:10.1186/1471-2105-8-367

Laible, C., and Le Novère, N. (2007). MIRIAM resources: tools to generate and resolve robust cross-references in systems biology. *BMC Syst. Biol.* 13:58. doi:10.1186/1752-0509-1-58

Lambeck, S., Dräger, A., and Guthke, R. (2010). "Network inference by considering multiple objectives: insights from in vivo transcriptomic data generated by a synthetic network," in *International Conference on Bioinformatics and Computational Biology, BIOCOMP 2010*, Vol. 2, eds H. R. Arabnia, Q.-N. Tran, R. Chang, M. He, A. Marsh, A. M. G. Solo, et al. (Las Vegas, NV: CSREA Press), 734–742.

Le Novère, N., Finney, A., Hucka, M., Bhalla, U. S., Campagne, F., Collado-Vides, J., et al. (2005). Minimum information requested in the annotation of biochemical models (MIRIAM). *Nat. Biotechnol.* 23, 1509–1515. doi:10.1038/nbt1156

Le Novère, N., Hucka, M., Anwar, N., Bader, G. D., Demir, E., Moodie, S., et al. (2011). Meeting report from the first meetings of the computational modeling in biology network (combine). *Stand. Genomic Sci.* 5, 230–242. doi:10.4056/sigs.2034671

Le Novère, N., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., et al. (2009). The systems biology graphical notation. *Nat. Biotechnol.* 27, 735–741. doi:10.1038/nbt.1558

Le Novere, N., Le Novere, N., Demir, E., Mi, H., Moodie, S., and Villeger, A. (2011). Systems biology graphical notation: entity relationship language level 1 (version 1.2). *Nature Precedings* doi:10.1038/npre.2011.5902.1

Lerman, J. A., Hyduke, D. R., Latif, H., Portnoy, V. A., Lewis, N. E., Orth, J. D., et al. (2012). In silico method for modelling metabolism and gene product expression at genome scale. *Nat. Commun.* 3, 929. doi:10.1038/ncomms1928

Lloyd, C. (2013). Opencell. *Encyclopedia of Systems Biology* 1567–1568. doi:10.1007/978-1-4419-9863-7_1526

Lloyd, C. M., Halstead, M. D. B., and Nielsen, P. F. (2004). CellML: its future, present and past. *Prog Biophys Mol Biol* 85, 433–450. doi:10.1016/j.pbiomolbio.2004.01.004

Machné, R., Finney, A., Müller, S., Lu, J., Widder, S., and Flamm, C. (2006). The SBML ODE solver library: a native API for symbolic and fast numerical analysis of reaction networks. *Bioinformatics* 22, 1406–1407. doi:10.1093/bioinformatics/btl086

Macilwain, C. (2011). Systems biology: evolving into the mainstream. *Cell* 144, 839–841. doi:10.1016/j.cell.2011.02.044

Markram, H., Meier, K., Lippert, T., Grillner, S., Frackowiak, R., Dehaene, S., et al. (2011). Introducing the human brain project. *Procedia Compu Sci* 7, 39–42. doi:10.1016/j.procs.2011.12.015

Matsuoka, Y., Funahashi, A., Ghosh, S., and Kitano, H. (2014). Modeling and simulation using CellDesigner. *Methods Mol. Biol.* 1164, 121–145. doi:10.1007/978-1-4939-0805-9_11

Mi, H., Muruganujan, A., Demir, E., Matsuoka, Y., Funahashi, A., Kitano, H., et al. (2011). BioPAX support in CellDesigner. *Bioinformatics* 27, 3437–3438. doi:10.1093/bioinformatics/btr586

Migliore, M., Morse, T. M., Davison, A. P., Marenco, L., Shepherd, G. M., and Hines, M. L. (2003). ModelDB: making models publicly accessible to support computational neuroscience. *Neuroinformatics* 1, 135–140. doi:10.1385/NI:1:1:135

Miller, A. K., Britten, R. D., and Nielsen, P. M. F. (2012). Declarative representation of uncertainty in mathematical models. *PLoS ONE* 7:e39721. doi:10.1371/journal.pone.0039721

Miller, A. K., Marsh, J., Reeve, A., Garny, A., Britten, R., Halstead, M., et al. (2010). An overview of the cellml api and its implementation. *BMC Bioinformatics* 11:178. doi:10.1186/1471-2105-11-178

Moodie, S. L., Swat, M. J., Kristensen, N. R., and Le Novère, N. (2013). PharmML: the pharmacometrics markup language. Available from: https://sites.google.com/site/pharmmltemp/documentation2/pharmml-specification_0.2.1.pdf

Moraru, I. I., Schaff, J. C., Slepchenko, B. M., Blinov, M. L., Morgan, F., Lakshminarayana, A., et al. (2008). Virtual cell modelling and simulation software environment. *IET Syst. Biol.* 2, 352–362. doi:10.1049/iet-syb:20080102

Müller, K. M., and Arndt, K. M. (2012). "Standardization in synthetic biology," in *Synthetic Gene Networks*, eds W. Weber and M. Fussenegger (New York: Springer), 23–43.

Myers, C. J., Barker, N., Jones, K., Kuwahara, H., Madsen, C., and Nguyen, N.-P. D. (2009). ibiosim: A tool for the analysis and design of genetic circuits. *Bioinformatics* 25, 2848–2849. doi:10.1093/bioinformatics/btp457

Nickerson, D. P., Garny, A., Nielsen, P. M. F., and Hunter, P. J. (2013). "Standards and tools supporting collaborative development of the virtual physiological human," in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE* (Osaka: IEEE), 5541–5544.

O'Brien, E. J., Lerman, J. A., Chang, R. L., Hyduke, D. R., and Palsson, B. Ø. (2013). Genome-scale models of metabolism and gene expression extend and refine growth phenotype prediction. *Mol. Syst. Biol.* 9, 693. doi:10.1038/msb.2013.52

Olivier, B. G., Rohwer, J. M., and Hofmeyr, J.-H. S. (2005). Modelling cellular systems with PySCeS. *Bioinformatics* 21, 560–561. doi:10.1093/bioinformatics/bti046

Olivier, B. G., and Snoep, J. L. (2004). Web-based kinetic modelling using JWS Online. *Bioinformatics* 20, 2143–2144. doi:10.1093/bioinformatics/bth200

Orth, J. D., Thiele, I., and Palsson, B. Ø. (2010). What is flux balance analysis? *Nat. Biotechnol.* 28, 245–248. doi:10.1038/nbt.1614

Palsson, B. Ø. (2011). *Systems biology: Simulation of Dynamic Network States.* Cambridge: Cambridge University Press.

Resasco, D. C., Gao, F., Morgan, F., Novak, I. L., Schaff, J. C., and Slepchenko, B. M. (2012). Virtual cell: computational tools for modeling in cell biology. *Wiley Interdiscip. Rev. Syst. Biol. Med.* 4, 129–140. doi:10.1002/wsbm.165

Rolfsson, O., Palsson, B. Ø., and Thiele, I. (2011). The human metabolic reconstruction recon 1 directs hypotheses of novel human metabolic functions. *BMC Syst. Biol.* 5:155. doi:10.1186/1752-0509-5-155

Sauro, H. M., Hucka, M., Finney, A., Wellock, C., Bolouri, H., Doyle, J., et al. (2003). Next generation simulation tools: the systems biology workbench and BioSPICE integration. *OMICS* 7, 355–372. doi:10.1089/153623103322637670

Sauro, H. M., Karlsson, T. T., Swat, M., Galdzicki, M., and Somogyi, A. (2013). libRoadRunner: a high performance SBML compliant simulator. *Cold Spring Harbor Laboratory*. doi:10.1101/001230

Schaefer, C. F., Anthony, K., Krupa, S., Buchoff, J., Day, M., Hannay, T., et al. (2009). PID: the pathway interaction database. *Nucleic Acids Res.* 37, D674–D679. doi:10.1093/nar/gkn653

Schaff, J. C., Lakshminarayana, A., and Smith, L. P. (2013). "Spatial processes," in *Technical Report*. Pasadena, CA: Caltech. Available from: http://sbml.org/Documents/Specifications/SBML_Level_3/Packages/spatial

Schellenberger, J., Park, J. O., Conrad, T. M., and Palsson, B. Ø. (2010). Bigg: a biochemical genetic and genomic knowledgebase of large scale metabolic reconstructions. *BMC Bioinformatics* 11:213. doi:10.1186/1471-2105-11-213

Schellenberger, J., Que, R., Fleming, R. M. T., Thiele, I., Orth, J. D., Feist, A. M., et al. (2011). Quantitative prediction of cellular metabolism with constraint-based models: the cobra toolbox v2.0. *Nat. Protoc.* 6, 1290–1307. doi:10.1038/nprot.2011.308

Schilstra, M. J., Li, L., Matthews, J., Finney, A., Hucka, M., and Le Novère, N. (2006). CellML2SBML: conversion of CellML into SBML. *Bioinformatics* 22, 1018–1020. doi:10.1093/bioinformatics/btl047

Schmidt, H. (2007). SBaddon: high performance simulation for the systems biology toolbox for MATLAB. *Bioinformatics* 23, 646–647. doi:10.1093/bioinformatics/btl668

Schmidt, H., and Jirstrand, M. (2006). Systems biology toolbox for MATLAB: a computational platform for research in systems biology. *Bioinformatics* 22, 514–515. doi:10.1093/bioinformatics/bti799

Schulz, M., Krause, F., Le Novère, N., Klipp, E., and Liebermeister, W. (2011). Retrieval, alignment, and clustering of computational models based on semantic annotations. *Mol. Syst. Biol.* 7, 512. doi:10.1038/msb.2011.41

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., et al. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* 13, 2498–2504. doi:10.1101/gr.1239303

Shapiro, B. E., Finney, A., Hucka, M., Bornstein, B. J., Funahashi, A., Jouraku, A., et al. (2007). *Introduction to Systems Biology*, Chapter SBML Models and MathSBML. Totowa, NJ: Humana Press, 395–421.

Shapiro, B. E., Hucka, M., Finney, A., and Doyle, J. (2004). MathSBML: a package for manipulating SBML-based biological models. *Bioinformatics* 20, 2829–2831. doi:10.1093/bioinformatics/bth271

Shen, S. Y., Bergmann, F. T., and Sauro, H. M. (2010). SBML2Ti*k*Z: supporting the SBML render extension in $\LaTeX$. *Bioinformatics* 26, 2794–2795. doi:10.1093/bioinformatics/btq512

Shepherd, G. M., Mirsky, J. S., Healy, M. D., Singer, M. S., Skoufos, E., Hines, M. S., et al. (1998). The human brain project: neuroinformatics tools for integrating, searching and modeling multidisciplinary neuroscience data. *Trends Neurosci.* 21, 460–468. doi:10.1016/S0166-2236(98)01300-9

Smith, L. P., Bergmann, F. T., Chandran, D., and Sauro, H. M. (2009). Antimony: a modular model definition language. *Bioinformatics* 25, 2452–2454. doi:10.1093/bioinformatics/btp401

Smith, L. P., Butterworth, E., Bassingthwaighte, J. B., and Sauro, H. M. (2013a). SBML and CellML translation in antimony and JSim. *Bioinformatics* 30, 903–907. doi:10.1093/bioinformatics/btt641

Smith, L. P., Hucka, M., Hoops, S., Finney, A., Ginkel, M., Myers, C. J., et al. (2013b). "Hierarchical model composition," in *Technical Report*. Available from: http://sbml.org/Documents/Specifications/SBML_Level_3/Packages/comp

Snoep, J. L., and Olivier, B. G. (2003). JWS online cellular systems modelling and microbiology. *Microbiology* 149, 3045–3047. doi:10.1099/mic.0.C0124-0

Sonnenschein, N., and Palsson, B. Ø. (2013). *MASS Toolbox*. Available at: http://opencobra.github.io/MASS-Toolbox/

Swainston, N., Smallbone, K., Mendes, P., Kell, D., and Paton, N. (2011). The SuBliMinaL toolbox: automating steps in the reconstruction of metabolic networks. *J. Integr. Bioinform.* 8, 186. doi:10.2390/biecoll-jib-2011-186

Takizawa, H., Nakamura, K., Tabira, A., Chikahara, Y., Matsui, T., Hiroi, N., et al. (2013). LibSBMLSim: a reference implementation of fully functional SBML simulator. *Bioinformatics* 29, 1474–1476. doi:10.1093/bioinformatics/btt157

Taylor, C. F., Field, D., Sansone, S.-A., Aerts, J., Apweiler, R., Ashburner, M., et al. (2008). Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project. *Nat. Biotechnol.* 26, 889–896. doi:10.1038/nbt.1411

Terfve, C., Cokelaer, T., Henriques, D., MacNamara, A., Goncalves, E., Morris, M. K., et al. (2012). CellNOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC Syst. Biol.* 6:133. doi:10.1186/1752-0509-6-133

Thiele, I., Swainston, N., Fleming, R. M. T., Hoppe, A., Sahoo, S., Aurich, M. K., et al. (2013). A community-driven global reconstruction of human metabolism. *Nat. Biotechnol.* 31, 419–425. doi:10.1038/nbt.2488

Töpfer, S., Guthke, R., Driesch, D., Wötzel, D., and Pfaff, M. (2007). "The Net*Gen*erator algorithm: reconstruction of gene regulatory networks," in *Knowledge Discovery and Emergent Complexity in Bioinformatics*, Volume 4366 of *Lecture Notes in Computer Science*, eds K. Tuyls, R. Westra, Y. Saeys, and A. Nowé (Berlin, Heidelberg: Springer).

van Iersel, M. P., Villéger, A. C., Czauderna, T., Boyd, S. E., Bergmann, F. T., Luna, A., et al. (2012). Software support for SBGN maps: SBGN-ML and LibSBGN. *Bioinformatics* 28, 2016–2021. doi:10.1093/bioinformatics/bts270

Vella, M., Cannon, R. C., Crook, S., Davison, A. P., Ganapathy, G., Robinson, H. P. C., et al. (2014). libNeuroML and PyLEMS: using Python to combine procedural and declarative modeling approaches in computational neuroscience. *Front. Neuroinform.* 8:38. doi:10.3389/fninf.2014.00038

Vlaic, S., Hoffmann, B., Kupfer, P., Weber, M., and Dräger, A. (2013). GRN2SBML: automated encoding and annotation of inferred gene regulatory networks complying with SBML. *Bioinformatics* 29, 2216–2217. doi:10.1093/bioinformatics/btt370

Waltemath, D., Adams, R., Beard, D. A., Bergmann, F. T., Bhalla, U. S., Britten, R., et al. (2011a). Minimum information about a simulation experiment (MIASE). *PLoS Comput. Biol.* 7:e1001122. doi:10.1371/journal.pcbi.1001122

Waltemath, D., Adams, R., Bergmann, F. T., Hucka, M., Kolpakov, F., Miller, A. K., et al. (2011b). Reproducible computational biology experiments with SED-ML-the simulation experiment description markup language. *BMC Syst. Biol.* 5:198. doi:10.1186/1752-0509-5-198

Waltemath, D., Bergmann, F. T., Chaouiya, C., Czauderna, T., Gleeson, P., Goble, C., et al. (2014). Meeting report from the fourth meeting of the Computational Modeling in Biology Network (COMBINE). *Stand. Genomic Sci.* 9, 1285–1301, doi:10.4056/sigs.5279417

Waltemath, D., Henkel, R., Winter, F., and Wolkenhauer, O. (2013). "Reproducibility of model-based results in systems biology," in *Systems Biology*, eds A. Prokop and B. Csukás (New York: Springer), 301–320.

Wimalaratne, S. M., Halstead, M. D. B., Lloyd, C. M., Cooling, M. T., Crampin, E. J., and Nielsen, P. F. (2009). A method for visualizing CellML models. *Bioinformatics* 25, 3012–3019. doi:10.1093/bioinformatics/btp495

Wittig, U., Rey, M., Kania, R., Bittkowski, M., Shi, L., Golebiewski, M., et al. (2014). Challenges for an enzymatic reaction kinetics database. *FEBS Journal* 281, 572–582. doi:10.1111/febs.12562

Wolstencroft, K., Owen, S., du Preez, F., Krebs, O., Mueller, W., Goble, C., et al. (2011). *The SEEK: A Platform for Sharing Data and Models in Systems Biology*, Vol. 500. San Diego, CA: Elsevier.

Wrzodek, C., Büchel, F., Ruff, M., Dräger, A., and Zell, A. (2013). Precise generation of systems biology models from KEGG pathways. *BMC Syst. Biol.* 7:15. doi:10.1186/1752-0509-7-15

Wrzodek, C., Dräger, A., and Zell, A. (2011). KEGG translator: visualizing and converting the KEGG pathway database to various formats. *Bioinformatics* 27, 2314–2315. doi:10.1093/bioinformatics/btr377

Wrzodek, C., Schröder, A., Dräger, A., Wanke, D., Berendzen, K. W., Kronfeld, M., et al. (2010). Module master: a new tool to decipher transcriptional regulatory networks. *BioSystems* 99, 71–81. doi:10.1016/j.biosystems.2009.09.005

Yu, T., Lloyd, C. M., Nickerson, D. P., Cooling, M. T., Miller, A. K., Garny, A., et al. (2011). The physiome model repository 2. *Bioinformatics* 27, 743–744. doi:10.1093/bioinformatics/btq723

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 01 September 2014; accepted: 14 November 2014; published online: 08 December 2014.

*Citation: Dräger A and Palsson BØ (2014) Improving collaboration by standardization efforts in systems biology. Front. Bioeng. Biotechnol. **2**:61. doi: 10.3389/fbioe.2014.00061*

This article was submitted to Systems Biology, a section of the journal Frontiers in Bioengineering and Biotechnology.

Copyright © 2014 Dräger and Palsson. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.