Published in IET Systems Biology
Received on 19th January 2009
Revised on 17th April 2009
doi: 10.1049/iet-syb.2009.0007

Special Issue – Selected papers from the Second q-bio Conference on Cellular Information Processing

ISSN 1751-8849

Integrating BioPAX pathway knowledge with SBML models

O. Ruebenacker I.I. Moraru J.C. Schaff M.L. Blinov

Center for Cell Analysis and Modeling, University of Connecticut Health Center, CT, USA
E-mail: blinov@uchc.edu

Abstract: Online databases store thousands of molecular interactions and pathways, and numerous modelling software tools provide users with an interface to create and simulate mathematical models of such interactions. However, the two most widespread used standards for storing pathway data (biological pathway exchange; BioPAX) and for exchanging mathematical models of pathways (systems biology markup language; SBML) are structurally and semantically different. Conversion between formats (making data present in one format available in another format) based on simple one-to-one mappings may lead to loss or distortion of data, is difficult to automate, and often impractical and/or erroneous. This seriously limits the integration of knowledge data and models. In this paper we introduce an approach for such integration based on a bridging format that we named systems biology pathway exchange (SBPAX) alluding to SBML and BioPAX. It facilitates conversion between data in different formats by a combination of one-to-one mappings to and from SBPAX and operations within the SBPAX data. The concept of SBPAX is to provide a flexible description expanding around essential pathway data – basically the common subset of all formats describing processes, the substances participating in these processes and their locations. SBPAX can act as a repository for molecular interaction data from a variety of sources in different formats, and the information about their relative relationships, thus providing a platform for converting between formats and documenting assumptions used during conversion, gluing (identifying related elements across different formats) and merging (creating a coherent set of data from multiple sources) data.

1 Introduction

An important method to understand cellular molecular networks is through the use of mathematical modelling. To generate a model, a researcher often needs to gather publicly available data about the relevant biological system. A rapidly growing market of supporting services and tools are available online: databases such as Reactome [1], BioCyc collection of Pathway/Genome databases [2], Pathway Interaction Database (PID) [3], BioModels repository of computable models [4], Integrating Network Objects with Hierarchies (INOH) database [5], Kyoto Encyclopedia of Genes and Genomes [6], storing thousands of molecular interactions and pathways; and software for creation and simulation of mathematical models such as VCell [7], Copasi [8], CellDesigner [9] and so on.

Obviously, it would be of enormous benefit to researchers if these databases and modelling software tools would work together seamlessly. Unfortunately, this is not the case. Would-be users are faced with many obstacles, mainly due to the fact that database-centric formats that are typically used to store molecular pathway information in databases and simulation-centric formats used by modelling software are semantically and structurally different. Presently, there exist a few tools that provide automatic conversion between the database and modelling formats (BiNom [10] plugin for Cytoscape [11], Patika [12]). However, as we will show below, such automatic conversion based on simple one-to-one mappings may not correspond to a modeller's intentions, or may lead to loss or distortion of data.

In this paper we introduce an approach for integration of cellular molecular pathway knowledge and models from

different formats. We have implemented this approach for the two most popular formats: for ontology-based pathway data, the biological pathway exchange (BioPAX [13]), and for kinetic models and simulations, the systems biology markup language (SBML [14]). The core of the implementation is the use of a bridging ontology that we named systems biology pathway exchange (SBPAX, alluding to SBML and BioPAX). It is designed to provide interoperability between data in different formats by a combination of one-to-one mappings to and from SBPAX and operations within the SBPAX data, as schematically depicted in Fig. 1. SBPAX is used for conversion of pathway data into a computational model by a Systems Biology Linker (SyBiL) software designed by the authors [15, 16], as depicted in Fig. 2.

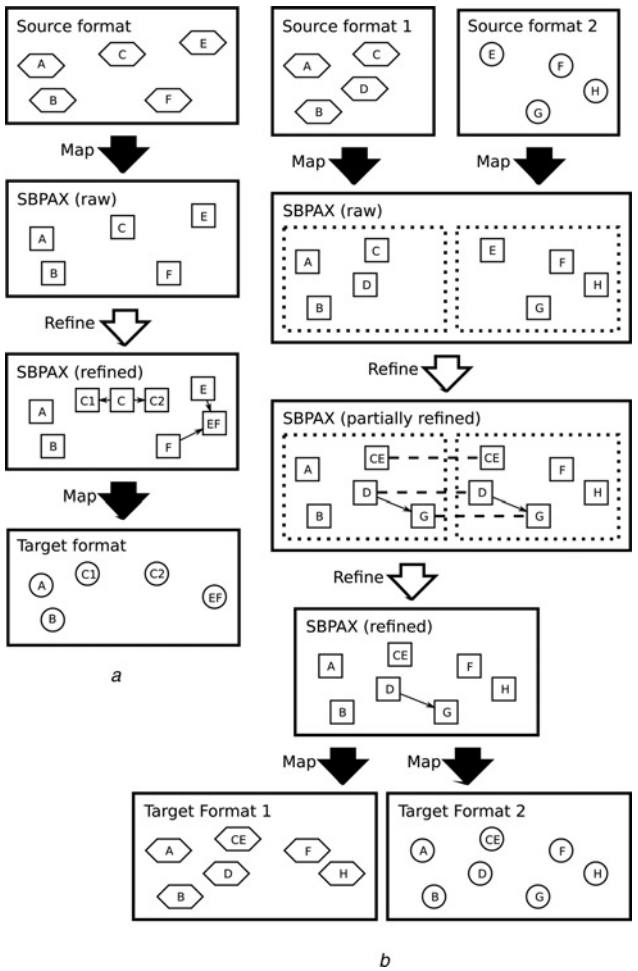

Figure 1 *SBPAX integration tasks consist of mapping and refinement steps*

a Conversion of one format to another. The results of refinement are new subset substances C1, C2 and superset substance EF. Subset relationships are shown by arrow. This procedure is illustrated in Fig. 2 for mapping from BioPAX to SBML

b Merging of two files of same or different formats. Refinement involves establishing identity relationships (dashed lines) and/or subset relationships (arrows) between data. The result is an SBPAX document with the combined network based on the information from the original data. The refined SBPAX data may then be mapped into a target format of choice, for example BioPAX, SBML (*NB*: this process is illustrated in Fig. 3 for two BioPAX files)

The concept of the SBPAX bridging ontology is to provide a flexible description of essential pathway data – basically the common subset of all formats (processes such as reactions and transports, the substances participating in these processes as reactants, products or catalysts, their locations and stoichiometric coefficients). Since SBPAX only defines common terms, it natively covers a much smaller domain than SBML or BioPAX, and it is not designed as a competing format. However, SBPAX is designed to be able to express anything other formats say about these terms. Therefore, the total expressive power of SBPAX is the common superset of these formats.

The most common practical problem for data conversion or integration is that one format does not always provide a simple way to express the same meaning of some entity or relationship in the other format. For example a species type in SBML may or may not correspond to a physical entity in BioPAX (see Section 3). Therefore, one of the design principles for SBPAX was to enable adding and preserving ancillary information that is required for conversion. Thus, SBPAX is designed to document the assumptions that were used for conversion of the original data, such as modelling assumptions used in generating SBML.

To better explain the design and implementation of SBPAX, we will first discuss the formats for modelling and knowledge representation in more detail (Section 2), and the limits of direct mappings between SBML and BioPAX (Section 3). In Sections 4 and 5 we introduce SBPAX as the glue between formats and demonstrate how it can be used for a number of activities that make use of multiple formats or multiple sources, including conversion (making data present in one format available in another, see Fig. 2), bridging (identifying related elements across different formats) and merging (creating a coherent set of data from multiple sources, see Fig. 3).

2 Data formats for knowledge representation and modelling

The main challenge of data integration is accommodating fundamental differences between formats that arise from different requirements and design principles. Our main interest lies in the description of molecular interaction networks, and we focus on two formats, SBML and BioPAX. Although both are often used to describe the same processes, they are semantically quite different: simulation-centric SBML is used for quantitative modelling, while database-centric BioPAX is used for qualitative knowledge representation. The resulting differences represent an important test case for a generic way of integration of different formats describing data

doi: 10.1049/iet-syb.2009.0007

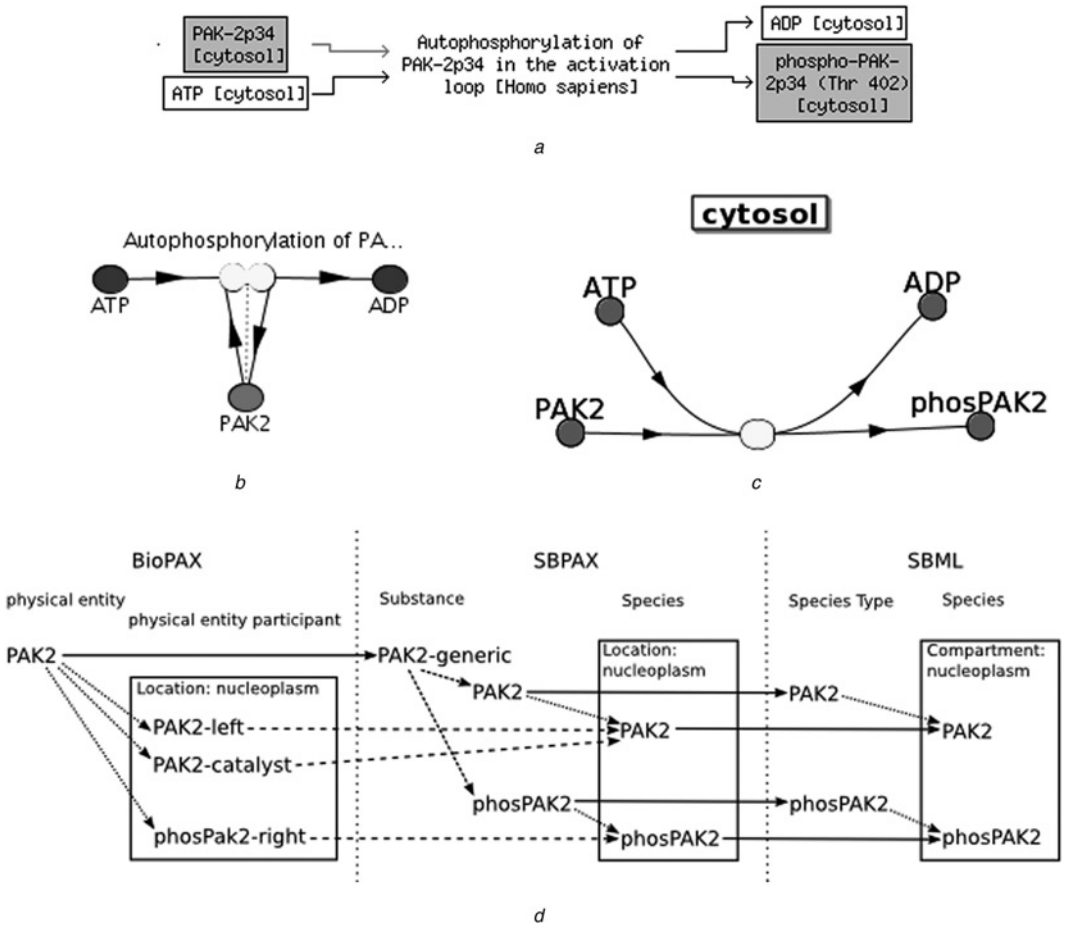

Figure 2 *Autophosphorylation of PAK2p34 (Reactome ID = 211650), where PAK2p34 is both reactant and catalyst, and its phosphorylated form is the product*

a Representation in Reactome. Grey arrow corresponds to a catalytic reaction
b Representation of Reactome's BioPAX export in SyBiL. All forms of PAK2p34 are represented by a single physical entity (note the three distinct lines between PAK2p34 and the reaction node, corresponding to PAK2p34 being a reactant, product and catalyst). Different shades of nodes corresponds to different types of physical entities in BioPAX
c Resulting SBML model imported into VCell. Different PAK2p34 forms are distinct species, but the catalyst is not shown
d Selected elements of conversion from SBML to BioPAX performed using SBPAX. Solid lines represent one-to-one mappings across formats, dashed lines are subset relationships in SBPAX and dotted lines are other relationships within a format. Objects inside boxes are specific for the location or compartment, and are derived from location/compartment-independent objects outside the boxes

related to the same knowledge domain (in this case, molecular pathway data). Moreover, both are established community standards for two research communities: SBML among modellers and BioPAX among database curators.

SBML was designed for models that contain the information necessary for an unambiguous mathematical description and simulation of a scenario, defining the meaning of each part in relation to other parts of the same model. Unambiguous description requires unique identification of all elements within the model; however, the model is being simulated without any larger context and unambiguous element identification is not required outside the model. A model consists of a number of optional lists, usually for species types (type of substance), compartments (space with defined boundaries that can contain substances), species (a certain substance in a certain compartment) and reactions (a process, such as a biochemical reaction, that produces or consumes one or more species). However, these are abstract constructs allowing a wide class of models. A reaction entry in an SBML file lists reactants, modifiers and products, each referring to a species, a stoichiometric coefficient and a compartment. A typical use-case involves giving initial amounts for each species and a rate law for each reaction, to simulate the time course of the concentrations of each species.

In contrast, BioPAX describes molecular pathways independently of any particular scenario. It aims to identify biophysical entities and their relationships in a way meaningful in the largest possible context, explicitly discouraging file boundaries and ordering entities into hierarchies of classes and relationships. BioPAX goes to lengths to enable identification of each physical entity and

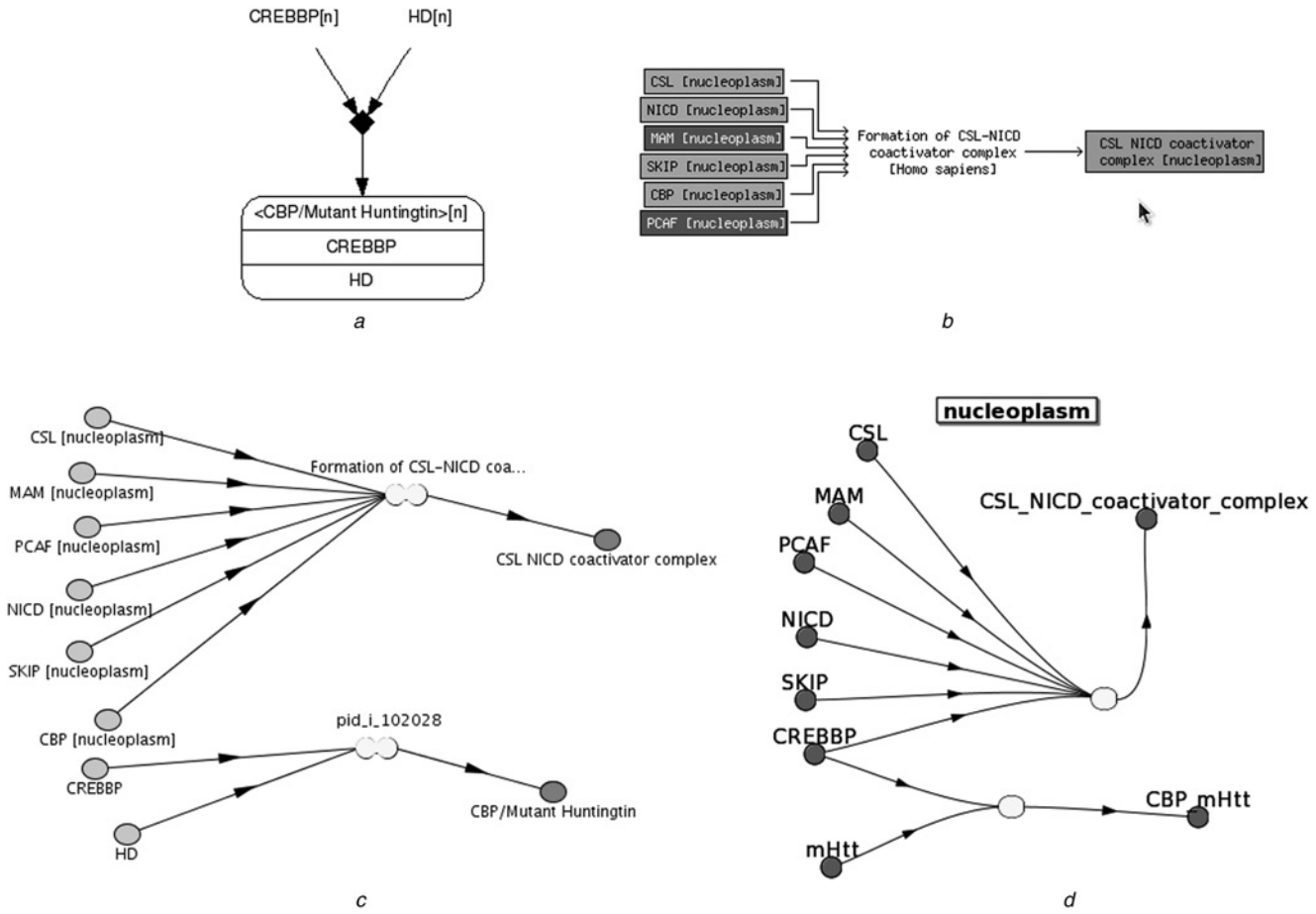

Figure 3 *Example of BioPAX to SBML conversion requiring user intervention*

a A process (ID = 102028) from the NCI Pathway Interaction Database (PID), involving CREB-binding protein CBP

b A process (ID = 212356) from Reactome involving the same substance in the same location, but there referred to as CBP in the nucleoplasm

c Screenshot of SyBil with both interactions imported; SyBiL initially displays two substances, since the information from PID and Reactome is not sufficient to identify them as being the same

d VCell displays the SBML model as the set of connected processes after SyBiL identified identical substances based on user input

its components, providing references to databases, specifying the sequence or the molecular structure. Further, terms assigned by authoritative sources (open controlled vocabularies) are used, for example the gene ontology [17].

Recently, as SBML models have been used by a growing number of applications, an increased interest in reusability has led to the development of community databases of SBML models [4], and as a result, the further development of SBML standards for parts identification in the form of structured annotations (Minimal Information Requested In the Annotation of Models, MIRIAM [18]). One of the promising developments is the systems biology ontology (SBO) [19], which is an ontology tailored specifically for computational modelling that can be used as a controlled vocabulary. MIRIAM sets a standard for annotating computational models in biology (and has been recently used in particular for models encoded in SBML) through controlled annotations of model components and references. Thus, MIRIAM specifies how external formats such as SBO or BioPAX can be used for annotations. However, SBO has not been designed to be used to actually store relationships between entities within an SBML document (or relationships to external data such as BioPAX data).

3 Mapping between SBML and BioPAX

The interest in integrating SBML, BioPAX and other formats has generated various attempts at converting one format into another. A few software tools (simulators and model editors) or plugins exist that have some conversion functionality (such as BiNOM [10]), and databases themselves often perform conversion to a format different from their native format (e.g. Reactome exports SBML and BioModels exports BioPAX). However, these conversion schemes are based on mapping one-to-one each element from the source to the target format, which is possible only for a subset of the data, as discussed below.

doi: 10.1049/iet-syb.2009.0007

3.1 Limitations of one-to-one mapping

One obvious limitation of one-to-one mapping is that format-specific extension information cannot be converted. For example since BioPAX lacks native means of expressing SBML's kinetic laws, these cannot be directly converted from SBML. Unsupported types of information can be included as comments in BioPAX or as MIRIAM-compliant annotations for SBML [18]. It is possible to describe in a MIRIAM-compliant way relationships between SBML data and other data, such as BioPAX. MIRIAM compliance alone is no guarantee that all relevant relationships are present, but if there is a direct correspondence between an SBML element and some external data object, there is a simple way to say that in MIRIAM.

However, the main problem is that creating an SBML model from BioPAX data (and vice versa) is a non-trivial procedure, because information common to SBML and BioPAX cannot be mapped one-to-one. Both formats provide means to describe processes, which substances participate in these processes, where they are located and what the stoichiometric coefficients are. The problem is that terms used to define processes (conversions in BioPAX and reactions in SBML) or substances (physical entities in BioPAX and species types in SBML) rarely refer to a single event or one molecule, but usually refer to a collection of many events or many molecules. For example how much can we change a molecule before it becomes a different substance, or before a reaction in which it participates becomes a different reaction? Does a protein become a different substance when folded differently or phosphorylated? The problem we face is that the answer from a biological knowledge representation perspective (in BioPAX) often differs from the answer from a modelling (SBML) perspective.

Since each SBML dataset describes a particular kinetic model, whether two compounds constitute the same substance or different usually depends on whether they behave differently or not in that particular model scenario. BioPAX, however, aims to make statements about entities independently of any particular scenario. Because of these differences, many elements do not map one-to-one between SBML and BioPAX data. For the case of species and physical entity, for example, common situations are the following.

1. An SBML species type is a subset of BioPAX physical entities. For example a protein is typically a single physical entity in BioPAX, but some SBML models have species types that correspond only to a certain phosphorylated form of the protein. BioPAX would imply these subsets by listing sequence features every time a subset participates in an interaction, but do not identify these subsets explicitly, whereas in a particular SBML model that needs to distinguish between them, they would have to be defined as separate species entries. For example in the partial autophosphorylation of PAK2 (Reactome ID = 211583) the resulting physical entity is a protein phosphorylated at five residues [20]. A modeller may want to consider only some of the residues and introduce a number of species corresponding to various combinations of selected residues being phosphorylated, as it was done in a study of EGF receptor signalling [21].

2. An SBML species type can represent two or more of BioPAX physical entities. For example different physical entities can be defined to be the same species in an SBML model if they behave in the same way in the context of that model. Kinetic models often use many phenomenological or mathematical approximations that take advantage of such situations. Moreover, if such a species participates in a reaction in SBML, the reaction element itself would typically correspond to a superset of an interaction element in BioPAX (two or more interactions, one for each of the corresponding physical entities). For example in fibroblast growth factor (FGF) signaling pathway (Reactome ID = 190236) a modeller may want to declare some of the 22 FGFR forms to be the same species, thereby reducing the number of reactions.

3.2 Required user intervention for conversion from BioPAX to SBML

As we saw, sometimes the question of whether a species will be converted into one or more physical entities, and vice versa, is not trivial. In [15] we discuss how it can be answered automatically by extensive analysis of extensions and annotations in the source file or import from other sources. While most common cases can be automated, a few cases remain where user intervention might be necessary. For example:

1. Determining the topology of the locations (dimensions, nesting) can be automated by storing such information about the most common locations. However, we cannot guarantee that we know all locations ever used or that we can anticipate all possible ways to identify or use a known location, so a fallback to user intervention may be necessary.

2. In most cases, when the same entity is used again, it is identified by the same reference. However, if we merge data from different sources, an equivalent but different reference may be used, and we may not be able to map between such references. Or, a modeller may substitute a substance with another one which is equivalent in a particular model [e.g. using mouse instead of human epidermal growth factor reception (EGFR)]. The example of required human intervention is given in Fig. 3. In this case, two interactions are brought from different databases, both involving the same physical entity. However, automatic identification of this entity as the same species is impossible due to insufficient information (e.g. it has a UniProt ID in Reactome, but not in PID). Note also the different names: Reactome calls it CBP in the nucleoplasm, while PID calls it CREBBP in the nucleus.

3. Automatic conversion can usually be done in cases where BioPAX is used to explicitly spell out the role of each substance participating in each process. Data from most pathway databases does fulfill these criteria, but some other sources do not. One example was communicated to us by Augustin Luna working on molecular interaction maps, MIM [22]. It appears that some MIM constructs cannot be represented in SBML but can be represented in BioPAX, such as modifications of reaction modifications. Such constructs are too ambiguous to be converted to SBML without human intervention.

4. In general, a user may want to introduce certain assumptions as modelling hypotheses. Fig. 4 provides an extreme case when a simple BioPAX interaction from Fig. 2 can be mapped to four different SBML models accounting for different modelling assumptions.

3.3 Refinement and reusability

Since conversion between BioPAX and SBML can thus not be usually done by simple one-to-one mapping, one has to correctly identify which part can be simply transferred to its proper place in the new format by one-to-one mapping, and which part needs to be untangled – and how. Usually, some extra refinement information is required which is not explicitly present in the source data.

This information, once it is obtained, should be stored in some format, so it can be used to document the resolution of mapping ambiguities and modelling assumptions, and used in future data manipulations (reverse conversions, merging with other data etc.). Typically, neither the source nor the target format does store this kind of information. If we could store it in the source format, it may be lost in the conversion; to store it in the target format requires that we perform the conversion first – but to do the conversion it needs to be already present, a chicken and egg problem. In Section 3.1, we saw cases where relationships between SBML and BioPAX data objects are not one-to-one. Thus, a systematic approach is required, which allows storing and reusing these relationships and thus making conversions between formats reproducible and reversible. With the help of SBPAX, a conversion between SBML and BioPAX will become two consecutive one-to-one mappings with an intermediate refinement step that is performed on, and recorded in, the SBPAX data (Fig. 1).

4 SBPAX

We have designed SBPAX as a bridging ontology to support the integration of datasets related to molecular networks and pathways that originate from different sources in different formats and that were created or being used for different purposes. Primary integration tasks are converting data from one format to another, gluing corresponding data sets in different formats and merging multiple datasets into one. SBPAX is a Web Ontology Language (OWL)-based [23] format consisting of classes and properties defined by the SBPAX ontology that defines relationships to BioPAX classes and properties representing core data (Fig. 5). Currently, it is developed to support any core information on molecular pathways (such as processes, the substances participating in these processes and where they are located) expressible in BioPAX and SBML (Fig. 6).

We describe below in more detail some of the SBPAX components that relate to such core information (Sections 4.1–4.3), as well as some of the specific elements related to support of SBML data and use of SBPAX for modelling

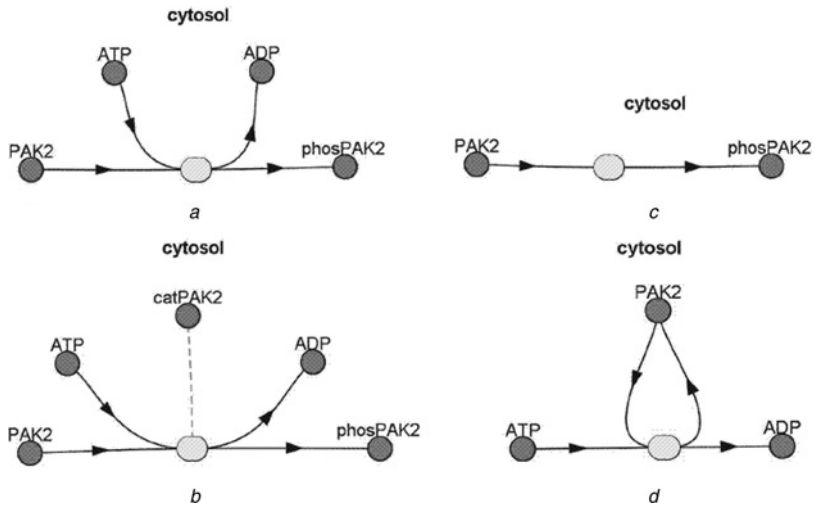

Figure 4 *Different SBML models that can be generated from BioPAX file from Fig. 1 using different modelling assumptions*

a Autophosphorylation of PAK involving ATP- > ADP conversion with ATP and ADP introduced as individual species. This model is in one-to-one correspondence with the original BioPAX file

b Phosphorylation with a catalytic co-factor catPAK2 introduced as an individual species

c Interaction with implicit ATP and ADP – a simplification that is often used in biochemistry outside energy metabolism

d Modelling of ATP–ADP conversion, where all forms of PAK are mapped to same species

doi: 10.1049/iet-syb.2009.0007

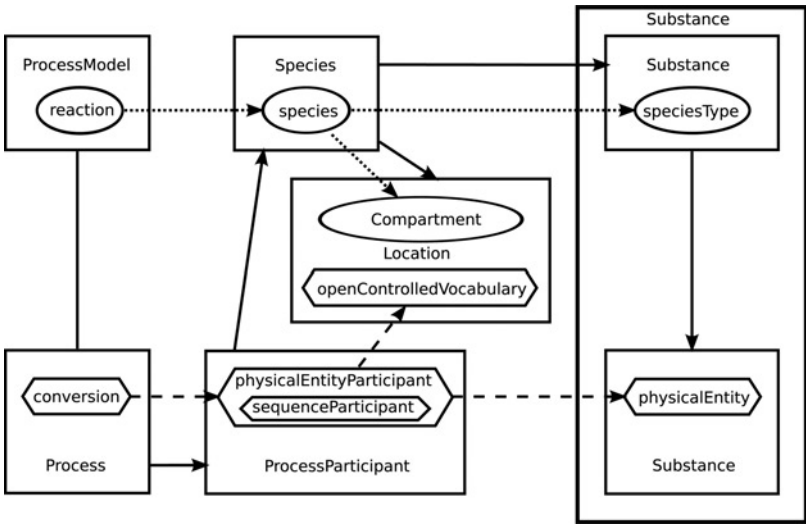

Figure 5 *Mapping between SBML and BioPAX using SBPAX. This figure shows the central SBML elements (ovals) and BioPAX classes (hexagons) used to refer to parts of a molecular pathway. Each is placed inside a rectangle representing the SBPAX class that can be used to bridge them. Relationships between elements and classes are expressed by arrows – solid for SBPAX, dashed for BioPAX and dotted for SBML*

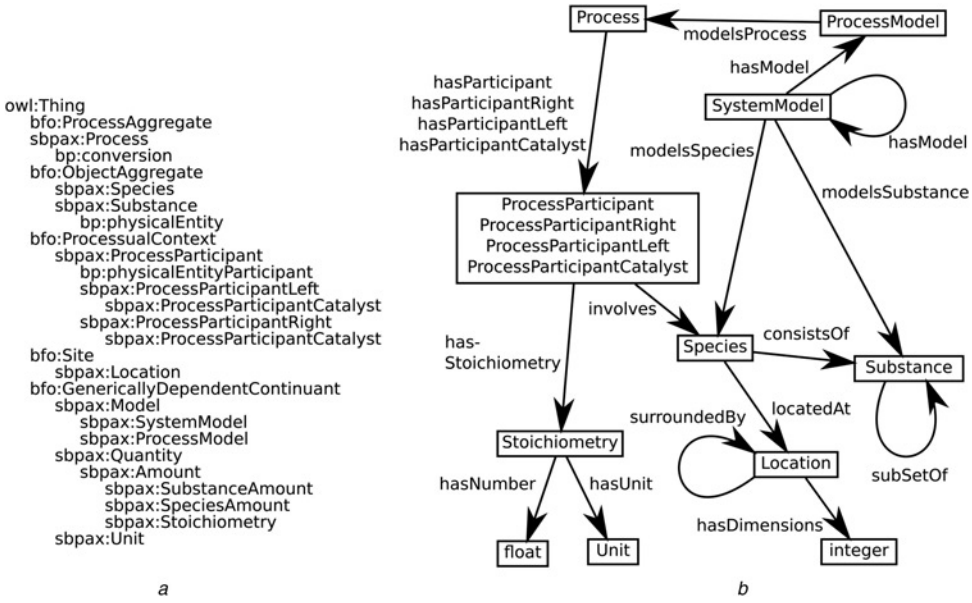

Figure 6 *SBPAX classes and properties*

a A class hierarchy of the central SBPAX classes and related classes from other ontologies. Prefixes indicate ontologies, for example bfo stands for Basic Formal Ontology and bp stands for BioPAX

b Central SBPAX properties are shown as arrows between the classes as they relate to each other

purposes (Section 4.4). The description of the full SBPAX specification is beyond the scope of this paper, and can be found in [24].

4.1 SBPAX substances and set relationships

SBPAX is designed to express all substances that can be a species type in SBML or a physical entity in BioPAX. An SBPAX substance is defined as any group of molecules or other compounds. SBPAX provides properties to define a substance as a superset or subset of another substance (Fig. 7*a*), or as the union or intersection of two other substances (useful for substances defined by constraints, e.g. on their phosphorylation state). This way, we can create a substance hierarchy and identify substances that can, just like a species type, cover more or less than one physical entity in BioPAX.

For example an SBPAX substance that is identical to a physical entity such as a protein can include subset substances that represent that protein in a phosphorylated form, as in (1)

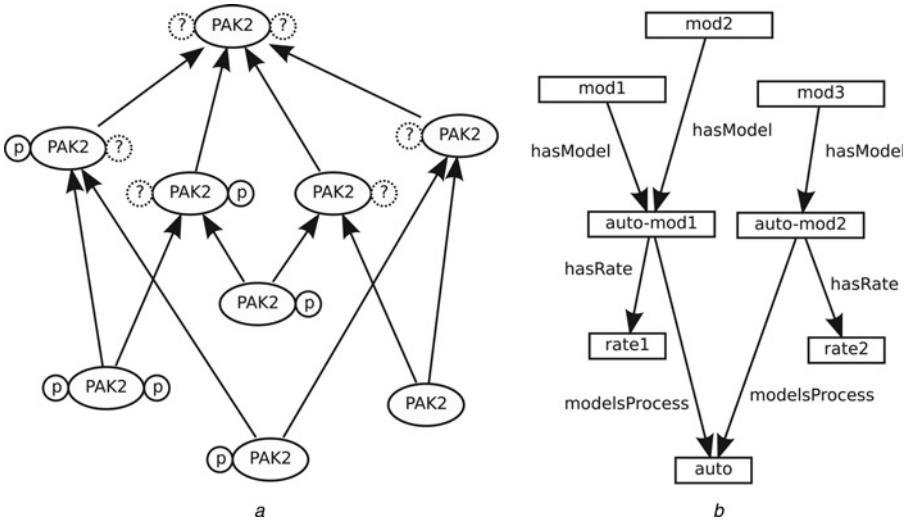

Figure 7 *Elements of SBPAX*

a Subsets of substances. PAK2 (adapted from Reactome ID = 211583, with just two phosphosites shown) with the state of both sites unspecified (circle with question mark and dotted line) is a superset of both phosphorylated form (circle with p) and unphosphorylated form (no circle). Arrows point from subsets to supersets. Partially specified forms (e.g. first site is phosphorylated, second site unspecified) are supersets of more explicitly specified forms (e.g. first site phosphorylated, second site not)
b System and process models. System models *mod1, mod2* and *mod3* include the same process *auto* as process models *auto-mod1, auto-mod1* and *auto-mod2* with kinetic laws *rate1, rate1* and *rate2* respectively. A process model is introduced as an intermediate to be able to use the same process with different kinetic laws in different models

of Section 3.1. Fig. 2 demonstrates the conversion of the autophosphorylation of PAK2p34 from BioPAX via SBPAX to SBML. Here PAK2 is a BioPAX physical entity, representing all forms of the protein PAK2p34. It maps to an SBPAX substance PAK2-generic, which has two subset substances PAK2 (the unphosphorylated form) and phosPAK2 (the phosphorylated form). These two subset substances map to SBML species types of the same names.

Although substances represent sets in reality, SBPAX represents them as instances rather than classes, to make SBPAX simpler and more flexible. Therefore, subset relationships are subSetOf in SBPAX, rather than subClassOf in OWL.

The amount of information stored in SBPAX for a given substance can vary widely according to how much information is available, ranging from a mere URI to a lot more information, such as subset or superset substances, or links to other OWL data taken from BioPAX data or from MIRIAM compliant SBML annotations.

4.2 SBPAX processes and process participants

An SBPAX process is a set of microscopic interactions between SBPAX substances. This allows an SBPAX process to represent any SBML reaction or BioPAX interaction. SBPAX describes participants in a process by a processParticipant (Fig. 6), which is the closest equivalent to a physical entity participant in BioPAX and a species reference in SBML.

4.3 SBPAX species and locations

An SBPAX species is defined as a substance with an assigned location, like an SBML species. It is equivalent to a set of BioPAX physical entity participants referring to the same physical entity and location. For example in Fig. 2*d* the SBPAX species PAK2 corresponds to the BioPAX physical entity participants PAK2-left and PAK2-catalyst, while the SBPAX species phosPAK2 corresponds to the BioPAX physical entity participant phosPAK2-right. Each SBPAX species maps to an SBML species of the same name. An SBPAX species has no attributes other than a substance and a location; in particular, it does not have context-related attributes like initial amount, which are delegated to a model (see Section 4.4). An SBPAX location is any identifiable space where some amount of a species can exist, corresponding to an SBML compartment, or to open controlled vocabulary terms used in BioPAX as cellular locations.

4.4 Models

To enable the use of core pathway information in specific contexts, such as required for kinetic models, SBPAX makes a critical distinction between the pathway and the context. The pathway is a common subset of SBML and BioPAX and is defined by terms such as processes, participants, locations and stoichiometric coefficients. The context includes anything that has to do with the setup of a scenario, or with mathematical and numerical elements used to describe it, such as initial conditions, rate laws and so on. Each model is represented by an SBPAX system model, which links to objects describing the context. For

doi: 10.1049/iet-syb.2009.0007

every process involved, an SBPAX system model links to an SBPAX process model, which links to an SBPAX process, which links to objects defining the pathway. This allows two models to link to the same pathway objects while linking to different context objects, as illustrated in Fig. 7*b*.

5 Integration

We describe integration of BioPAX and SBML data using SBPAX in Fig. 5. A prototype of a software application designed to automate parts of this process and facilitate the user-required activities (SyBiL) is currently under development [15, 16]. The integration can be divided into the following two types of tasks, as depicted in Fig. 1 and described in detail in the following three subsections.

1. Converting BioPAX data into SBML, and vice versa (Fig. 2);

2. Bridging data in BioPAX and SBML (by identifying and storing relationships between BioPAX and SBML data) and merging different datasets into one (Fig. 3).

Specific steps that are involved in these tasks are (i) converting BioPAX and SBML data to SBPAX, which is a simple one-to-one mapping, resulting in raw SBPAX data; (ii) refining SBPAX data, by adding information to the raw SBPAX data (the additional information coming from analysis of the original data, from other data sources or supplied by the user); and (iii) one-to-one mapping of the refined SBPAX data to the desired BioPAX or SBML data, either to create this data or to link to existing data. The details are explained in the following subsections.

5.1 Conversion from BioPAX to SBML

The conversion from BioPAX to SBML as implemented in SyBiL typically goes as follows:

1. Automatically map BioPAX data one-to-one to SBPAX data based on the fact that core BioPAX classes (interaction, physical entity participant and physical entity) are subclasses of SBPAX classes (process, process part and substance respectively (Fig. 7*a*)). Multiple versions of BioPAX can be used at the same time, as long as the required bridging relations (maintained in SBPAX) are provided.

2. Establish which BioPAX physical entity participants should be the same SBPAX substance, by evaluating database references and scanning sequence features, whether they represent post-translational modifications. For example in Fig. 2 we establish that BioPAX physical entity participants PAK2-left and PAK2-catalyst refer to the same SBPAX substance (PAK2), while phosPAK2-right refers to a distinct SBPAX substance (phosPAK2). This step can be often done automatically, but some user intervention may be required in some cases (e.g. when sequence features are not consistently listed).

3. Determine the topology of the cellular locations (dimensions, nesting). SyBiL is designed to store the topology of commonly used locations, but user intervention is necessary for specific less common layouts.

4. Generate an SBPAX system model for the context, and decide what parts of the pathway should be included. Note that an SBPAX system model can contain another SBPAX system model (or parts thereof) as a sub-model, facilitating hierarchical model building, which is the goal of one of the forthcoming SBML Level 3 Hierarchical Model Composition extension.

5. Generate SBPAX species by scanning relationships between SBPAX substances and creating subset substances where necessary: for example when only some phospho forms of an entity participate in a reaction. Assign locations to SBPAX species. The role of subset substances and locations is illustrated in Fig. 2*d*, where PAK2 and phosPAK2 are subset substances of PAK2 generic, and from each, an SBPAX species is derived in the location nucleoplasm. This step can be done automatically, but also can be adjusted or fine tuned based on users' modelling assumptions.

6. Automatically map one-to-one from SBPAX to SBML, turning the relevant substances, species, locations and processes (or process models) in SBPAX into species types, species, compartments and reactions in SBML.

The resulting SBML may not contain rate laws, initial conditions or other quantitative information, since such data do not exist in the BioPAX source data, but will otherwise be a complete SBML model that can be further processed with any tool supporting SBML.

Any information added during steps (2) to (5) is stored in SBPAX. Thus, next time the conversion can be reproduced fully automatically.

Finally, another critical aspect is the mechanisms by which the relationships between the generated SBML model and the SBPAX data, and by extension the BioPAX data, can be recorded and maintained in the resulting SBML file. The SBPAX or BioPAX data will be included as MIRIAM-compliant annotations, for example as the URI of the corresponding SBPAX element. This way, the original BioPAX document can be recovered from either SBPAX or SBML

5.2 Conversion from SBML to BioPAX

The conversion from SBML to SBPAX as implemented in SyBiL typically goes as follows:

1. Automatically generate raw SBPAX from SBML. Since SBML is not an RDF/OWL-based format, mappings between SBML and SBPAX are not part of the SBPAX ontology, but are simply one-to-one: an SBML model becomes an SBPAX system model, an SBML species type becomes an SBPAX substance, an SBML species becomes an SBPAX species, an SBML compartment becomes an SBPAX location, and an SBML reaction becomes an SBPAX process model with its associated SBPAX process.

2. Check for each SBPAX substance whether it is a proper BioPAX physical entity. Proper MIRIAM-compliant annotations eliminate the need for user input in most cases.

3. Express all SBPAX substances that are not proper BioPAX physical entities as subset of superset SBPAX substances of proper BioPAX physical entities. For example, PAK2 and phosPAK2 in Fig. 2 are not BioPAX proper physical entities because they cover only some phospho forms, while a protein in BioPAX always refers to all phospho forms. Their common superset substance PAK2 generic refers to all phospho forms and therefore is a proper BioPAX physical entity. Since, as we discussed in Section 4.1, SBML species type can be subsets or supersets of proper BioPAX physical entities, mapping one-to-one from raw SBPAX to BioPAX would generate physical entities that violate BioPAX conventions, making the data unreliable (e.g. for querying) and defeating the purpose of BioPAX. Thus, refinement must be performed. As we describe in [15], it can be automated based on text processing. Refinement data are stored in SBPAX, by mapping every SBML species type or BioPAX physical entity to an SBPAX substance and storing a set of relationships among substances.

4. If an SBPAX substance is a subset of a BioPAX physical entity, determine sequence features. For every SBML reaction in which the species type participates, there will be a sequence participant listing these sequence features and location. This step can be done automatically if sufficient annotations are present.

5. If an SBML species type is a union of BioPAX physical entities, split each reaction in which it participates into one interaction per physical entity. This step is automatic to the extent that existing annotations allow.

6. Automatically map SBPAX objects one-to-one to corresponding BioPAX objects, as specified in Fig. 5. During mapping, each SBPAX substance has to be classified as a protein, DNA, RNA, a complex or a small molecule, and provided with a database reference to identify it. If no annotations are given, then user input is required.

After the entire core information is mapped to SBPAX and then to BioPAX, the entire SBML data are included in SBPAX. This procedure is non-trivial because SBML is defined by an XML (Extensible Markup Language [25]) schema, while SBPAX is defined by an OWL – that is, SPPAX is not an XML-based format, although it can, like anything in OWL, be serialised using XML (but also in other ways, such as N3 [26]). However, SBPAX can also include data from a non-OWL format such as SBML. Such data are divided into fragments, and each fragment is added as a string via a special SBPAX property verbatim attached to its most closely related SBPAX object. For example the complete definition of an SBML species (including attributes such as boundary condition) is attached to the corresponding SBPAX species. Other SBML objects (e.g. events, assignment rules etc.) will be represented by SBML fragments that may be attached to the SBPAX system model. The entire SBML model can then be recovered from such XML fragments when needed.

Thus, the refined SBPAX file is used for (i) conversion to BioPAX, (ii) storage of refinement information documenting relationships between SBPAX objects and (iii) recovery of original SBML file.

5.3 Merging data from different formats

Merging (creating a coherent set of data from multiple sources) is important for many problems, such as (i) assembling a large pathway or model from smaller sets of data, possibly both in SBML and BioPAX, (ii) building a repository from which multiple pathways or models can be extracted, (iii) assembling new pathways or models from the merged data. BioPAX supports nesting pathways, whereas SBPAX supports nesting system models, as illustrated in Fig. 7*b*. Thus, SBPAX provides a capability for merging pathways or models into larger ones, creating a hierarchy of nested models or pathways. SBML does not currently support nesting models, but is expected to support it in the near future (via the SBML Level 3 Hierarchical Model Composition extension). Meanwhile, nested system models will be flattened out when being converted to SBML Level 2.

Once the data are mapped from SBML and BioPAX to SBPAX, all the data can simply be dumped together, since OWL data are essentially a collection of statements in no particular order. Next, we need to bridge multiple datasets within SBPAX, that is establishing relationships between identical objects. This refinement procedure will be performed by software tools like Sybil [15, 16], and it relies mostly on BioPAX and MIRIAM-compliant references. In [15] we have discussed how we can identify elements based on text analysis. Again, the user is the final decision maker. Since we want to use data in different formats together, it may be desirable, after bridging it, to store it in the same place. For this purpose, SBPAX supports inclusion of the entire SBML and BioPAX data. The included refinement information records the relationships between SBML and BioPAX entities.

doi: 10.1049/iet-syb.2009.0007

6 Conclusion

Multiple formats are being used in systems biology to describe molecular interaction networks, either tailored to assisting modellers (e.g. SBML) or to assembling and organising biological knowledge (e.g. BioPAX). They differ in many aspects, but they also do have many things in common when they describe core data related to pathways, such as processes and the substances participating in these processes. We have developed a new bridging ontology, SBPAX, with the goal of being flexible and descriptive enough to express anything these formats say about the common terms, to facilitate data integration and conversion between formats. We designed SBPAX to provide a unified way of describing integration tasks (converting, bridging and merging the data in multiple formats) and storing in SBPAX the data and all relationships required to reproduce the entire process. We can split most tasks into smaller steps, vary the order in which they are performed and store intermediary results in SBPAX. Thus, SBPAX can act as a repository for molecular interaction data from a variety of sources in different formats, and for documenting relationships between data and assumptions made for conversion between formats.

We focused initially on support for BioPAX and SBML data. These are widely popular standards for exchange and storage of pathway data and models, and thus, bridging these formats can be of great benefit to the systems biology community. Besides, multiple tools are being developed to support data aggregation within each of these formats (e.g. Ontology-Based Aggregator of Biological Pathway Datasets [27], SBMLMerge for Combining Biochemical Network Models [28]). Nevertheless, additional popular native formats exist in both the data and simulation worlds, such as Cell Systems Ontology (CSO) [29] and CellML [30]. We do expect our SBPAX-based approach to be generic and flexible enough to support elements related to pathways or pathway models from these and other formats as well, since SBPAX can be easily extended with additional domain-specific elements. With regard to implementation, SBPAX is currently being used by the Systems Biology Linker software [15, 16] that is developed for the purpose of bringing pathway data from multiple sources into the VCell simulation framework [7, 16]. Finally, it is worth noting that our approach is bringing a standard OWL technology into the world of XML-encoded models. OWL technologies greatly facilitate interfacing with external source data and applications [31, 32]. SBPAX provides capabilities for identification and classification of data, the organisation of complex relationships between originally unrelated data, as well as storing, transferring and querying data, and performing various modes of automatic reasoning [15].

7 Acknowledgments

Authors would like to thank Alan Ruttenberg, Elgar Pichler, Andrea Splendiani and Augustin Luna for many ideas and helpful discussions regarding this project. The project was supported in part by NIH R01 GM076570 grant (MLB), NIH U54 RR022232 grant (OR, IIM, MLB) and NIH P41 RR13186 (IIM, JCS).

8 References

[1] VASTRIK I., D'EUSTACHIO P., SCHMIDT E., *ET AL.*: 'Reactome: a knowledge base of biologic pathways and processes', *Genome Biol.*, 2007, **8**, (3), R39. Database is accessible at: http://www.reactome.org/

[2] KARP P.D., OUZOUNIS C.A., MOORE-KOCHLACS C., *ET AL.*: 'Expansion of the BioCyc collection of pathway/genome databases to 160 genomes', *Nucleic Acids Res*, 2005, **33**, (19), pp. 6083–6089. Database is accessible at: http://biocyc.org

[3] http://pid.nci.nih.gov/, accessed April 2009

[4] LE NOVÈRE N., BORNSTEIN B., BROICHER A., *ET AL.*: 'BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems', *Nucleic Acids Res*, 2006, **34** (Database issue), D689–D691. Database is accessible at: http://biomodels.net

[5] http://www.inoh.org/, accessed April 2009

[6] OGATA H., GOTO S., SATO K., FUJIBUCHI W., BONO H., KANEHISA M.: 'KEGG: Kyoto Encyclopedia of genes and genomes', *Nucleic Acids Res.*, 1999, **27**, (1), pp. 29–34. Database is accessible at: http://www.genome.jp/kegg/

[7] MORARU I.I., SCHAFF J.C., SLEPCHENKO B.M., *ET AL.*: 'Virtual Cell modelling and simulation software environment', *IET Syst. Biol.*, 2008, **2**, (5), pp. 352–362. Software is accessible at: http://vcell.org

[8] HOOPS S., SAHLE S., GAUGES R., *ET AL.*: 'COPASI – a COmplex PAthway Simulator', *Bioinformatics*, 2006, **22**, (24), pp. 3067–3074. Available at: http://copasi.org

[9] FUNAHASHI A., TANIMURA N., MOROHASHI M., KITANO H.: 'CellDesigner: a process diagram editor for gene-regulatory and biochemical networks', *BIOSILICO*, 2003, **1**, pp. 159–162. Software is accessible at: http://celldesigner.org

[10] ZINOVYEV A., VIARA E., CALZONE L., BARILLOT E.: 'BiNoM: a Cytoscape plugin for manipulating and analyzing biological networks', *Bioinformatics*, 2008, **24**, pp. 876–877

[11] SHANNON P., MARKIEL A., OZIER O., *ET AL.*: 'Cytoscape: a software environment for integrated models of biomolecular interaction networks', *Genome Res.*, 2003, **13**, (11), pp. 2498–2504

[12] DEMIR E., BABUR O., DOGRUSOZ U., *ET AL.*: 'PATIKA: an integrated visual environment for collaborative construction and analysis of cellular pathways', *Bioinformatics*, 2002, **18**, (7), pp. 996–1003

[13] LUCIANO J.S.: 'PAX of mind for pathway researchers', *Drug Discov. Today*, 2005, **10**, pp. 937–942. Documentation is available at: http://biopax.org

[14] HUCKA M., FINNEY A., SAURO H.M., *ET AL.*: 'The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models', *Bioinformatics*, 2003, **19**, pp. 524–531. Documentation is available at: http://sbml.org

[15] RUEBENACKER O., MORARU I.I., SCHAFF J.C., BLINOV M.L.: 'Kinetic modeling using BioPAX ontology'. Proc. 2007 IEEE Int. Conf. Bioinformatics and Biomedicine, 2007, pp. 339–348

[16] BLINOV M.L., RUEBENACKER O., MORARU I.I.: 'Complexity and modularity of intracellular networks: a systematic approach for modelling and simulation', *IET Syst. Biol.*, 2008, **2**, (5), pp. 363–368

[17] ASHBURNER M., BALL C.A., BLAKE J.A., *ET AL.*: 'Gene ontology: tool for the unification of biology. The Gene Ontology Consortium', *Nat. Genet.*, 2000, **25**, pp. 25–29

[18] LE NOVERE N., FINNEY A., HUCKA M., *ET AL.*: 'Minimum information requested in the annotation of biochemical models (MIRIAM)', *Nat. Biotechnol.*, 2005, **23**, (12), pp. 1509–1515

[19] LE NOVERE N.: 'Model storage, exchange and integration', *BMC Neurosci.*, 2006, **7**, (Suppl 1), p. S11

[20] GATTI A., HUANG Z., TUAZON P.T., TRAUGH J.A.: 'Multisite autophosphorylation of p21-activated protein kinase gamma-PAK as a function of activation', *J. Biol. Chem.*, 1999, **274**, (12), pp. 8022–8028

[21] BLINOV M.L., FAEDER J.R., GOLDSTEIN B., HLAVACEK W.S.: 'A network model of early events in epidermal growth factor receptor signaling that accounts for combinatorial complexity', *Biosystems*, 2006, **83**, (2–3), pp. 136–151

[22] KOHN K.W., ALADJEM M.I., WEINSTEIN J.N., POMMIER Y.: 'Molecular interaction maps of bioregulatory networks: a general rubric for systems biology', *Mol. Biol. Cell*, 2006, **17**, (1), pp. 1–13

[23] http://www.w3.org/TR/owl-features/. Accessed April 2009

[24] http://vcell.org/biopax/sbpax.html. Accessed April 2009

[25] http://www.w3.org/XML/. Accessed April 2009

[26] http://www.w3.org/DesignIssues/Notation3.html. Accessed April 2009

[27] JIANG K., NASH C.: 'Ontology-based aggregation of biological pathway datasets'. *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, 2005, **7**, pp. 7742–7745

[28] SCHULZ M., UHLENDORF J., KLIPP E., LIEBERMEISTER W.: 'SBMLmerge, a system for combining biochemical network models', *Genome Inform.*, 2006, **17**, (1), pp. 62–71

[29] JEONG E., NAGASAKI M., SAITO A., MIYANO S.: 'Cell system ontology: representation for modeling, visualizing, and simulating biological pathways', *In Silico Biol.*, 2007, **7**, (6), pp. 623–638

[30] LLOYD C.M., HALSTEAD M.D., NIELSEN P.F.: 'CellML: its future, present and past', *Prog. Biophys. Mol. Biol.*, 2004, **85**, (2–3), pp. 433–450

[31] LISTER A.L., POCOCK M., A WIPAT A.: 'Integration of constraints documented in SBML, SBO, and the SBML – Manual facilitates validation of biological models', *J. Integr. Bioinf.*, 2007, **4**, (3), p. 80

[32] KÖHN D., LENA STRÖMBÄCK L.: 'A method for semi-automatic integration of standards in systems biology'. 19th Int. Conf. Database and Expert Systems Applications (DEXA) 2008, 2008, (*Lecture Notes in Computer Science* **5181**, (0745)), pp. 745–752

doi: 10.1049/iet-syb.2009.0007