Theory Biosci. (2011) 130:45–54
DOI 10.1007/s12064-010-0108-6

ORIGINAL PAPER

Using views of Systems Biology Cloud: application for model building

Oliver Ruebenacker · Michael Blinov

Received: 17 November 2009 / Accepted: 4 July 2010 / Published online: 21 August 2010

Abstract A large and growing network ("cloud") of interlinked terms and records of items of Systems Biology knowledge is available from the web. These items include pathways, reactions, substances, literature references, organisms, and anatomy, all described in different data sets. Here, we discuss how the knowledge from the cloud can be molded into representations (views) useful for data visualization and modeling. We discuss methods to create and use various views relevant for visualization, modeling, and model annotations, while hiding irrelevant details without unacceptable loss or distortion. We show that views are compatible with understanding substances and processes as sets of microscopic compounds and events respectively, which allows the representation of specializations and generalizations as subsets and supersets respectively. We explain how these methods can be implemented based on the bridging ontology Systems Biological Pathway Exchange (SBPAX) in the Systems Biology Linker (SyBiL) we have developed.

Keywords Data integration · Systems Biology knowledge · Modeling · Semantic Web · SBML · BioPAX

Introduction

Turning knowledge into a model elements and annotations involves multiple steps: first, a user, with a research project in mind, may search for available information, such as PubMed or database records for pathways, substances and processes. A typical model creation (such as in Virtual Cell modeling and simulation framework (Moraru et al. 2008)) involves specifying species (represented by one type of node), reactions (represented by another type of node), and specify which species serve as reactants, products or catalysts (represented by different types of arrows connecting species and reactions). If the model is intended to be reused, then the user will use knowledge from the literature to add annotations, such as assigning UniProt (http://uniprot.org/) or ChEBI (Degtyarenko et al. 2009) identifiers to species nodes and PubMed identifiers to reaction nodes. The connection between the reality described in the article or database, the model elements (species, reaction, participation, etc.) and the annotation (UniProt and PubMed identifiers) had to be made manually by the user (or even the curator), which is labor intensive and error prone (and, in the case of annotations, often neglected).

However, much of the necessary knowledge that used to be only available through human language contained in publications is now freely available in machine-processable form from public websites, so that a growing number of tasks can be fully or partially automated. Thanks to Semantic Web technology such as shared links and identifiers, sources of Systems Biology knowledge data have become one interconnected pool of knowledge that can be queried, retrieved, processed, and molded into models and model annotations. Hundreds of pathways and models containing hundreds of thousands of interactions and substances are contained in this rapidly growing pool, and links to genes, organisms, anatomical features, and literature are provided. Terms, records, and their relationships are categorized, standardized, and interlinked across the web. Automation can relieve the user of tedious and error-prone tasks such as copying names, identifiers, and

O. Ruebenacker · M. Blinov (✉)
Center for Cell Analysis and Modeling, University of Connecticut Health Center, Farmington, CT, USA
e-mail: blinov@uchc.edu

references. It can also enable queries of relationships across resources, which are inaccessible to simple keyword searches.

These interlinked sources of terms and records, which we call the Systems Biology Cloud, include databases of a variety of entities, such as pathway databases Reactome (Vastrik et al. 2007), Nature PID (http://pid.nci.nih.gov/) and HumanCyc, compounds database CheBi, proteins database UniProt, etc. Pathway Commons (http://www.pathwaycommons.org/pc/) and Bio2RDF (Belleau et al. 2008) aggregate knowledge data from many sources and make it searchable. BioModels (Le Novère et al. 2006) and the Virtual Cell database store complete models.

The data of the Systems Biology Cloud are encoded in multiple forms. It includes ontology-based RDF and OWL (OWL Web Ontology Language, http://www.w3.org/TR/owl-features/) formats, such as Biological Pathway Exchange (BioPAX) (Luciano 2005). Controlled vocabularies and ontologies such as Systems Biology Ontology (SBO), Gene Ontology (GO), Foundational Model of Anatomy (FMA), Basic Formal Ontology (BFO), and other ontologies from the Open Biomedical Ontologies (OBO) provide shared terms to identify, categorize, and relate pathways, entities, organisms, and anatomical features. Pathways are available in BioPAX, with links to records on the relevant substances, organisms, anatomical features, and publications. The Systems Biology Cloud also includes data encoded in various XML formats, such as the Systems Biology Markup Language (SBML) (Hucka et al. 2003), Virtual Cell Markup Language (VCML), Cell Markup Language (CellML) (Loyd et al. 2004), and others. Annotations of biochemical models are governed by the MIRIAM standard (Minimum Information Requested in the Annotation of biochemical Models, Le Novère et al. 2005).

Based on these standards, entities in the Systems Biology Cloud and their relationships are identified and organized into categories in machine-processable form. For example, unique identifiers are given to publications through references in the PubMed database (http://pubmed.gov), to books through the International Standard Book Number (ISBN), to proteins and genes based on sequence through references to UniProt database (http://www.uniprot.org), and to small molecules through references to the ChEBI database (http://www.ebi.ac.uk/chebi/).

Entities of primary concern to modeling include substances and processes. Relationships of primary concern for modeling include, which processes are part of which pathways, which substances participate at which locations in which processes, which substances are components of which complexes. If data are used to produces some sort of result, then we can identify the portion of the data that influences the result. For example, if we build a reaction network model, then the data that determine the reactions, species, rate laws, and initial conditions is included in that portion, while the references to publications are not (for simplicity, assume this example without annotations). If we draw an image representing a pathway, then the relevant portion is any data that determines how the image will look like. Such a portion of systems biology data relevant for a certain task can be represented by what we call a view. A view is a graph with nodes representing entities of concern and edges representing relationships of concern, with attributes attached to nodes (such as concentrations in a view of a model, or shapes and colors in a data representation in visual editor), attached to edges (such as edge type) or attached to the view as a whole (such as view name). In general, there will be many relationships between elements of the view and elements not part of the view. Such relationships are not considered part of the view. For example, any data describing processes and substances participating in these processes can be described by a view where every process and every substance is a node, and every time a substance participates in a process, this can be represented by an edge connecting the process and the substance. A typical view is a bipartite graph representing reaction network, such as in VCell modeling framework (Moraru et al. 2008; Slepchenko et al. 2003). Each node has a set of attributes attached, such as concentration or population number for species, rate expression for substance node, or stoichiometry for relationship. Sometimes, a somewhat different view can be constructed more closely resembling the structure of the data. For example, RDF data can be seen as a graph where every resource is a node, and every statement is an edge, if the object is a resource; and an attribute, if the object is a literal. With this approach, using the Systems Biology Cloud means organizing it into a collection of views, which can be overlapping and interconnected. The creation of a view itself does not change the data, except for adding the date necessary to describe the view. Rather, the purpose of views is not to perform changes, but to describe them. For example, if three elements are processed to create five new elements, it may not be possible to map the three original elements to the resulting five, but we can map a view that contains the original three elements to a view that contains the resulting five.

Different views of the same reality involve varying levels of detail, such as a substance in a visualization view (e.g., a protein) may correspond to multiple substances in another view (such as several species in a modeling view). The primary challenge is then how to reconcile conflicts between views. A conflict is any case where the same reality is represented by graphs with a different structure, because some thing represented by a single node in one graph is not a single node in another graph.

In this article, we discuss manipulating views to efficiently use the Systems Biology Cloud for building and annotating models. These strategies will also be useful for tasks similar to these, such as visualization. In the section, "Different views of Systems Biology Cloud", we will outline different uses of views and discuss how to build various consistent views based on the knowledge from the Systems Biology Cloud. The various views resolve the same object to varying degrees of details, necessary for different purposes (modeling, pathway visualizing, etc.). In the section, "Substances and processes", we will elaborate how views help with the representation of reaction networks and other types of biological information. In the section, "Manipulating views of substances and processes", we elaborate how to manipulate views of biological networks, involving substances and processes. In the section, "From cloud to model with views and sets", we outline how we approach these issues using the bridging ontology SBPAX and the Systems Biology Linker (SyBiL) (Ruebenacker et al. 2007, 2009; Blinov et al. 2008), a tool we have designed.

Different views of Systems Biology Cloud

For the same set of data, different purposes may require different views with different kinds of attributes. For example, for a pathway retrieved in BioPAX from Pathway Commons, we may have various views:

(1) To analyze the data, we need a view that represents the entire information available, such as a graph of RDF triplets (Resource Description Framework, http://www.w3.org/RDF).
(2) To visualize the data, typically a substantially simplified view is needed because an entire pathway is too complicated to visualize efficiently. For example, BioPAX data can be visualized in Patika (Demir et al. 2002), SyBil, or Cytoscape (Shannon et al. 2003). In this view, elements have additional attributes associated with it, such as shape, label, or color.
(3) To model the data, we also need substantial simplifications to make the model efficient, but we are also constrained to simplifications that do not alter the result in uncontrolled ways. Some elements have additional attributes associated with them, for example, every process has a rate law and every substance has an initial concentration. Note that another model of the same pathway can have different rate laws and different concentrations. Thus, the simplified reaction network without kinetics and initial conditions can be considered a view in itself, while the two models are views on top of it. A project can, therefore, involve a hierarchy of different views, building on top of a collection of views obtained from the cloud.
(4) A model with annotations can be considered a view that extends the model without annotations.

Since Systems Biology knowledge is stored aiming for completeness, it is usually available as a bulky raw view, and the need for simplification to achieve efficiency or clarity is substantial. Without simplification, a view may be too large for efficient simulation and visualization may be too confusing, while significant simplification may yield a view that works well enough for a given purpose but loses a lot of details. For each type of view, the user may perform operations which may use or modify attributes specific for such view. For example, in the visualization view, the user may change label for each element. In the modeling view, the user may change initial concentrations, rates, etc.

The challenge is to keep relationships to the original records valid even if data is modified substantially for various purposes. Model building is often incremental: a user wants to add additional elements to a given model by searching and retrieving additional knowledge based on information already present in the model. Typically, different parts of a view relate to different places in the sources and undergo varying degrees of modifications during the lifetime of a model. Often, modeling requires that the model does not entirely correspond to the data. One such requirement is the need for simplification; another is taking into account artifacts. A common simplification is omitting the explicit representation of some substances or processes while still taking their effect into account. A typical example is including processes that involve ATP and ADP while omitting explicit mentioning of these molecules themselves in the model. The motivation is that, tracking the concentrations of ATP and ADP in a model by adding all ways in which these are produced or consumed is usually not very useful, because they participate in countless processes. Instead, rate laws are often adjusted, for example, by replacing the concentrations of ATP and ADP by constant parameters. In other cases, the experiment causes artifacts, meaning details that relate to the way the experiment is done, which need to be in the model to correctly predict the experiment, but are not part of more generic data (i.e., data not limited to this particular experimental method). A typical case is an experiment where a protein can only be measured if it is tagged: a model would have to distinguish between tagged and untagged, but the data only contain the protein without the distinction of tagged or untagged.

Each of these data alteration creates a new view of the data, which we call a *derived view*. Derived views contain all the data of the original data view; however, they hide some parts shown in the original view (such as deleted

elements), or show some parts absent in the original view (such as added elements). Hidden parts have no influence on the use of a view per se (e.g., simulation of a model), but keep track of relationships among views to help verification and modification of a view based on the original view. This allows connecting model and data during continued model development. In "Manipulating views of substances and processes" section, we will discuss different types of operations on reaction networks, and in "From cloud to model with views and sets" section, we will discuss data structure and software implementation of views.

Substances and processes

The basic elements of systems biology reality we intend to describe are substances and processes. To be compatible with different notions of substances and processes, we need to be as flexible as possible to be expressive without making assumptions that may be violated by the knowledge data we encounter on the web. What follows is an approach that is compatible with all of the most popular systems biology formats. It is the model behind the Systems Biological Pathway Exchange (SBPAX) bridging ontology, which we have discussed elsewhere (Ruebenacker et al. 2007, 2009).

Substances as sets

We assume a substance (e.g., water, ATP, MFA, phosphorylated EGFR) to be a measurable quantity consisting of a set of specimens, typically compounds consisting of a group of bound atoms. The main challenge is to describe all sets and supersets of substances which may be used in different views. This allows us to describe all subsets and supersets of substances which may be used in different views. For example, if we define EGFRp as EGFR phosphorylated at site Y992 and EGFRu as not phosphorylated at site Y992, then EGFR is the union of EGFRp and EGFRu. We can distinguish EGFR molecules according to the phospho-state of two sites, say, Y992 and Y1045, into four disjoint subsets EGFRpp, EGFRpu, EGFRup, and EGFRuu, such that EGFRp is the union of EGFRpp, EGFRpu, and EGFRup, and EGFR is the union of all four.

In general, a view can contain substances which relate to each other via various set relationships. We provided several examples (Ruebenacker et al. 2009), such as

(a) The substance can be replaced by a union of disjoint sub-classes, and all reactions in which it participates can be replaced by reactions of the subsets, as far as these take place. This may require extra knowledge, but in many cases, it can be resolved easily if the difference is an irrelevant phosphorylation site or complex component. For example, a modeler may want to introduce several distinct phosphoforms of a receptor as separate species, or introduce a mutant.

(b) The substance can be replaced by the super-set substance, provided that all reactions, in which it appears as reactant, are equally valid for the super-set substance. This loses some information and simplifies the model, which may or may not be desired. A typical example is specifying superset substance "ligand" in place of multiple ligands with similar functions.

Processes as sets

If substances are sets of specimens, then naturally processes such as reactions are sets of microscopical processes. For example, a reaction R(A + B → C + D) is a set of processes, each of which consumes an a of A and a b of B and produces a c of C and a d of D. If this reaction appears in a view, it is usually assumed that for any a of A and any b of B, the reaction can take place. For this discussion, we assume R to be irreversible—a reversible reaction is understood as two separate reactions, one for each direction. Note that reversibility refers to sets, not individual elements: if a and b react to c and d and then back, then they may turn not into a and b, but instead into specimen a2 and b2, which may be different from a and b (Fig. 1a).

According to our definition of processes as sets it is not required that every a and b can produce every c and d. In

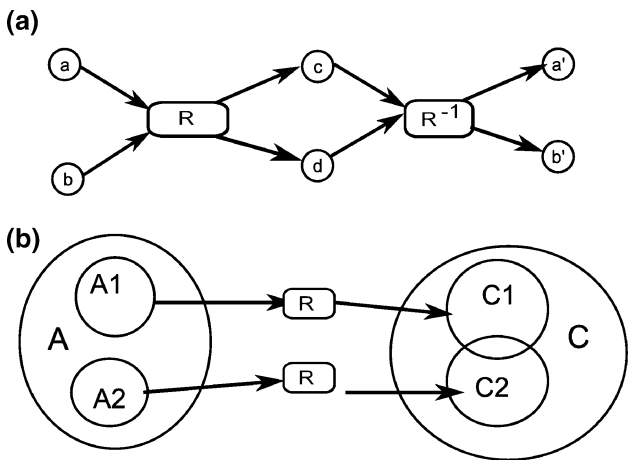

Fig. 1 Processes as sets of microscopical processes: **a** the reverse process R^{-1} of a process R is one where reactants and products as sets are reversed. It is possible that the original reactant (*a* and *b*) are not reverted to their original state, but to different members of the original reactant set (a' and b'); **b** it is also possible that disjoint subsets (*A1* and *A2*) of the reactant set (*A*) may be converted into non-disjoint subsets (*C1* and *C2*) of the product set (*C*)

fact, it is not even assumed that every c of C or every d of D can be produced. While this may seem counter-intuitive at first, it is only consistent with the way reactions are usually understood: for example, if we mark A with a carbon 14 isotope, then either C or D should be marked with a carbon 14, too. We cannot combine marked reactants with unmarked products or vice versa.

Assuming that A is the union of A1 and A2, we can infer that R is the union of R1(A1 + B → C + D) and R2(A2 + B → C + D). This can be done, because the reaction is guaranteed to be possible for any a of A. Note, however, that the same cannot be done on the product side. For example, if C is the union of C1 and C2, then this cannot be used to write R as a union based on C1 and C2 because it is not known under what conditions R will produce elements of C1 or C2. It is possible that the reaction produces only elements of C1 or C2. One way to split products is to obey proper dependence on the reactants: if we define C1 as the set of all c2 of C which may be produced if the reactant a1 of A is also of A1, then R1 becomes R1(A1 + B → C1 + D). If we define likewise C2, then R2 is R2(A2 + B → C2 + D), and R is still the union of R1 and R2. Note that even if A1 and A2 are disjoint, C1 and C2 need not be disjoint; in fact, they may not even be distinct. However, the reverse is true: if C1 and C2 are disjoint, so are A1 and A2 (Fig. 1b).

Manipulating views of substances and processes

This section explains how manipulations of a data set, especially manipulations relevant to modeling, can be understood in terms of views and representing substances and processes as sets. Creating a model from data is understood as deriving from an original view (the data) a new view (the model). Parts of the model that are not changed are considered subviews that correspond to subview in the data. Parts of the model that have been changed are considered subviews derived from subviews of the data.

From a BioPAX conversion to an SBML reaction

If the pathway data are encoded in BioPAX Level 2 and the derived model is encoded in SBML, not every entity in the data corresponds to a single entity in the model. For example, species in SBML has no equivalent in BioPAX Level 2: if EGFR participates as a reactant in three reactions in the cytosol and two reactions in the cell membrane, we would have two SBML species (EGFR in cytosol and EGFR in the membrane). In BioPAX Level 2, we would have instead one physical entity EGFR and five physical entity participants, corresponding to EGFR participating in five reactions. As BioPAX Level 2 is converted to SBML, views can mark the relationships between SBML species and BioPAX physical entities (Fig. 2a). In BioPAX Level 3, on the other hand, this conversion is much easier, since a physical entity corresponds roughly to an SBML species.

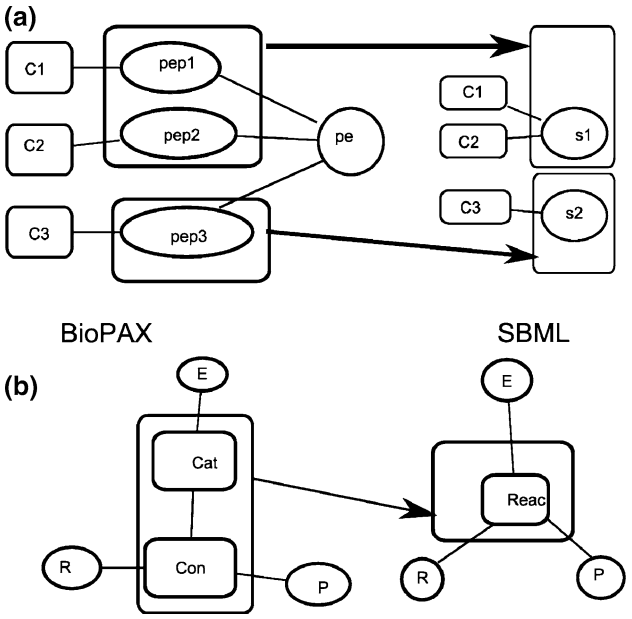

Fig. 2 Views help to clarify the conversion from BioPAX to SBML: **a** BioPAX Level 2 physical entity participants (pep) referring to the same physical entity, location and state (*pep1* and *pep2*) are mapped to the same SBML species *s1*, while another one (*pep3*) is mapped to another species *s2*. BioPAX conversions (*C*) are mapped to SBML reactions with the same names; **b** a BioPAX conversion and its attached catalysis are mapped to an SBML reaction

Another example of relation is a conversion with catalysis. In BioPAX (any level) it consists of two entities, the catalysis and the conversion. In SBML, this corresponds to a single entity, a reaction. A subview of the data that contain the catalysis and the conversion is mapped to a subview of the model that contains the reaction (Fig. 2b).

Deleting substances

The first step in generating a model view is to select data items of interest to the modeler and declare them the data view. However, modelers often include a process, but decide not to use all of its participants. A frequent example is phosphorylation reaction R(A + ATP → Ap + ADP). ATP and ADP are often omitted from models: since they participate in many interactions, most of which are not part of the model, their concentration cannot be predicted by the model. Substance deletion is expressed in terms of views as follows: a subview of the model containing a simpler reaction R′(A → Ap) is declared to be derived from a subview of the data containing R.

Merging and splitting substances and processes

Another common approximation in models is the merging of similar substances, which can imply the merging of processes in which they participate. For example, in yeast, the Pheromone Alpha Factor Receptor (R) has two possible ligands: Mating Hormone Alpha Factor 1 and Mating Hormone Alpha Factor 2. The two ligands have separate entries in UniProt, thus they are usually described by separate entities in pathway databases. Each ligand binds to the receptor to form a ligand–receptor complex, which is described by a two separate processes. Since the processes are similar, modelers often do not distinguish between the two ligands in a model, as is exemplified in model 72 of BioModels database, where there is only one ligand L, one complex RL and one process of binding, $R + L \rightarrow RL$. Note that whether two processes can be merged into a single process is dependent on extra information not present in a pathway database, such as rate laws for these reactions (Fig. 3a).

A typical example is merging a protein A and its precursor pre-A into their union A′. Any process involving A and pre-A would have to be replaced by a process containing A′ instead. A process that turns A into pre-A would be then deleted (Fig. 3b). Splitting a substance while deriving a model view can be necessary in some cases, for example to describe an experiment where some specimens of a protein A are tagged, so that A becomes a union of A-tagged and A-untagged. To declare a new view being derived from the previous (unsplit), we need to add set relationships, such as A′ being the union of A and pre-A (Fig. 3c).

Processes aggregation

Perhaps the most frequent way to simplify reaction networks is collapsing a cluster of constituent processes into a single aggregate process. The typical example is Michaelis–Menten kinetics, which is understood as the kinetics of an aggregate process (a conversion subject to catalysis) that consists of two constituent processes each with mass action kinetics: the first constituent process is the reversible binding of a substrate S to an enzyme E, R1($S + E \rightarrow SE$); the second constituent process is the irreversible release of the product, R2($SE \rightarrow E + P$) (Fig. 4, view V1). The aggregate process is the process from substrate to product, R3($S + E \rightarrow P + E$) (Fig. 4, view V2). Michaelis–Menten kinetics derives from the assumption that the rate of change in the concentration of ES can be neglected compared to the rate of change of S and P. As a result, both constituent processes are assumed to have the same reaction rate, which is at the same time the reaction rate of the aggregate process. This can be expressed by a view containing R being derived from a view containing R1 and R2. Further steps may be the omission of the catalyst (Fig. 4, view V3), a typical simplification. Should there be different

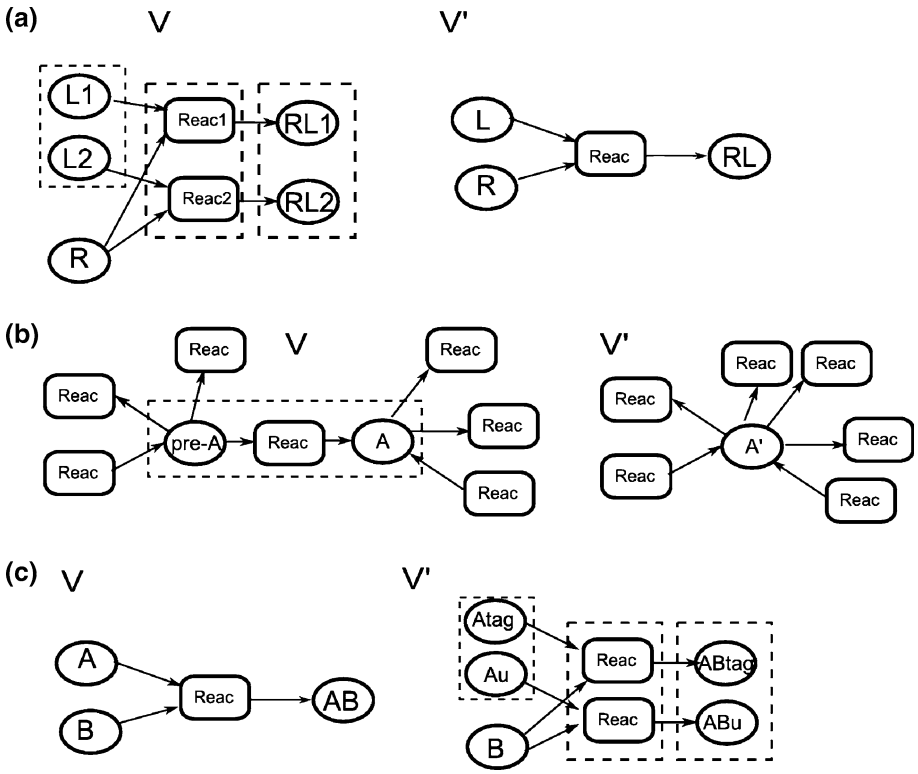

Fig. 3 Tracking sets of substances and processes using views. **a** *V′* is derived from *V* by merging two species *L1* and *L2* into species *L* (the union of *L1* and *L2*), which implies merging their complexes *RL1* and *RL2* into one complex *RL* (union of *RL1* and *RL2*), and merging the two complex formations into one; **b** *V′* is derived from *V* by merging precursor of *A* and *A* into *A′* (union of *A* and its precursor), which eliminates the reaction that turns the precursor of *A* into *A*; **c** *V′* is derived from *V* by splitting *A* into untagged *A* (*Au*) and tagged *A* (*Atag*), which implies splitting complex *AB* into *ABtag* and *ABu*, as well as splitting the complex formation into two reaction. Here, *A* is the union of *Au* and *Atag*, *AB* is the union of *ABu* and *ABtag*, and the original complex formation is the union of the two resulting complex formation

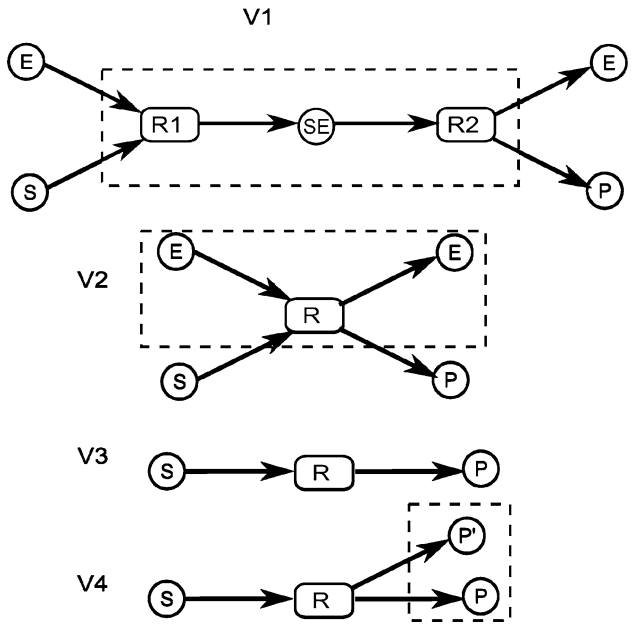

Fig. 4 Derivation of modeling views from data views. View *V1* shows the original data, showing two processes (*R1* and *R2*). View *V2* is derived from *V1* through merging of the processes into one (*R*), eliminating the intermediary (*SE*). A view *V3* is further derived from *V2* using a typical modeling simplification: omission of the catalyst (*E*). For various reasons, a compound (*P*) may be split into two (*P* and *P*′), as shown in view *V4*

versions of the product relevant to the model, the product can be split in two (Fig. 4, view V4).

Note that an aggregate process is different from a process union: if R3 was the union of R1 and R2, it would mean that each element of R3 is element of R1 or R2. In the aggregate process each element is a combination of an element of R1 and an element of R2.

Lockstep aggregates

Before we discuss aggregates in general, we will first discuss a simpler special case, which we call *the lockstep aggregate*. An example for this is Michaelis–Menten kinetics. Consider a substance B which only participates in two reactions: R1(A → B + C) and R2(B → D). Let us assume the net rate of change of B is small compared to the reaction rates of R1 and R2. We say that R1 and R2 are in lockstep. In this case, R1 and R2 can be substituted by their aggregate R3(A → C + D) while simultaneously deleting substance B (Fig. 5a). Further reactions can be added to the lockstep aggregate, if any of the other substances A, C and D appear in only one other reaction, and in matching roles (i.e., A as a product or C or D as a reactant), so that they can be considered intermediaries. Note that lockstep aggregates can have branches, e.g., if C participates in only one other reaction, and is reactant there, this would make C another intermediaries in the same lockstep aggregate (Fig. 5b). A lockstep aggregate is defined by listing its participating reactions and its intermediaries. It is represented by a view derived from a view containing all constituent processes and all intermediaries.

Aggregates

Consider reactions R1(A → B) and R2(2B → C). If B participates in no other reactions and we assume again steady state (that the rate of change of B is small compared to the reaction rate), then the rate of R2 is approximately half the rate of R1. We could substitute this by R4(2A → C) as for the lockstep aggregate, and while the rates are not the same for R1, R2 and R4, their ratios are known (i.e., rate of R4 equals rate of R2, which equals half rate of R1) (Fig. 6a).

The requirement, that each intermediary is reactant and product in exactly one process can be dropped if we know for some other reason, that the reactions are in quasi-steady state and we know the rations of the reaction rates. For example, if we add a third reaction R3(3B → D), and assume quasi-steady state, this means the ratios between the rates of R1, R2 and R3 would be fixed, too. In this case,

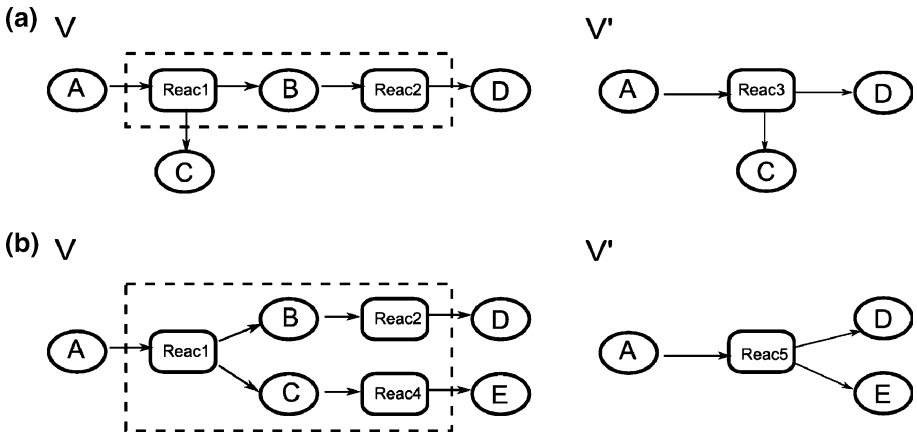

Fig. 5 Using views to track the creation lockstep aggregates: **a** in a linear example, reactions 1 and 2 in view *V* are merged into reaction 3 in view *V*′, eliminating intermediary *B* **b** in a branched example, reactions 1, 2 and 4 in view *V* are merged into reaction 5 in view *V*′, eliminating intermediaries *B* and *C*

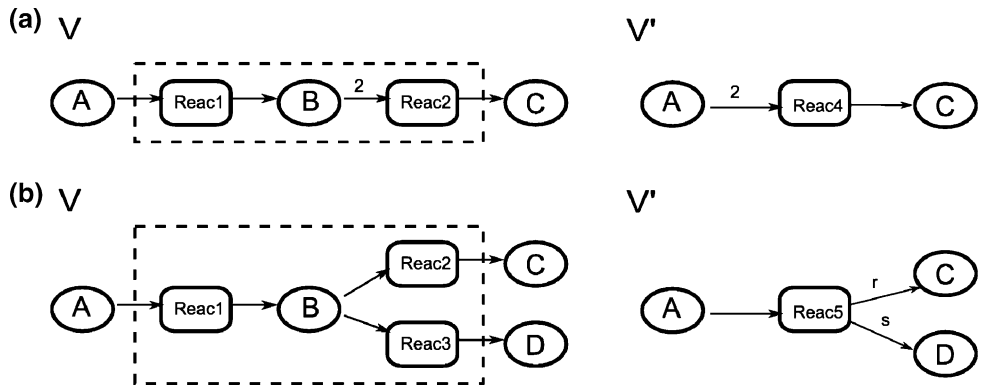

Fig. 6 Using views to track aggregates that are not lockstep: **a** when merging reactions 1 and 2 into reaction 4 and eliminating intermediary *B*, the stoichiometric coefficient of 2 for reaction 2 causes a stoichiometric coefficient of 2 for reaction 4; **b** when merging reactions 1, 2 and 3 into reaction 5, eliminating intermediary *B*, reaction 5 has non-trivial stoichiometric coefficients r and s depending on the relative reaction rates of reactions 2 and 3

we can substitute R1, R2 and R3 by an aggregate R5(A → rC + sD) with non-integer and less obvious stoichiometric coefficients r and s, and likewise less obvious reaction rates. To define such an aggregate, reactions and intermediaries must be listed, as well as for each reaction the ratio between its rate and the rate of the aggregate. Like lockstep aggregate, a general aggregate view is derived from a view containing the constituent processes and intermediaries (Fig. 6b).

From cloud to model with views and sets

Sets and views are complementary building blocks to navigate the cloud and collect the knowledge data used for building and annotating models: set relationships exist between individual entities within the same scenario, while view relationships exist between groups of entities and across scenarios. Among other uses, views are an important component of user interfaces, as users often select a group of entities and apply a series of actions to them. The selection may be based on a query or data set, and may be subsequently filtered or manually modified. Actions include various ways of displaying data, modifying data or building and annotating a model. Such a selection can be considered a view, which can be named, stored and modified and used to create corresponding or derived views. Less visible roles of views are keeping track of relationships across modifications.

Consider the autophosphorylation of EGFR. A search for "human EGFR" and "autophosporylation" can find the process of autophosphorylation of EGFR, for example, from Reactome. Downloading this process (for example in BioPAX format) will create a data view.

By adding a kinetic rate law and performing other modifications (e.g., dropping ADP and ATP), a model view is derived from the data view. As the user adds more processes, a data view and a model view are created for each process. Then the view of the whole model is the union of the model views of the processes included, and it is derived from the union of the data views from which all the process model views were derived. The user can easily create the appropriate model elements and the accompanying annotations. For example, to create an SBML model, the process would become an SBML reaction, and a substance at a location would become an SBML species. The EGFR species is annotated by the UniProt identifier (P00533) and the reaction is annotated by the publication references (e.g., PubMed number, journal, title, authors) found in the BioPAX data.

Subsets are created, among other occasions, when BioPAX data encode modifications. For example, autophosphorylation of EGFR involves several phosphorylation sites on an EGFR monomer. For simplicity, we will first discuss the case of two sites. Each site has two states: phosphorylated (marked as "p") and unphosphorylated (marked as "u"). Based on phosphorylation states, we can distinguish four variants of EGFR: EGFRuu, EGFRup, EGFRpu and EGFRpp, each of which is a subset of EGFR (the first suffixed "p" or "u" refers to the first site and so on). There are two processes that phosphorylate the first site (omitting other participants): P1a(EGFRuu → EGFRpu) and P1b(EGFRup → EGFRpp). If we assume that the phosphorylation of the first site is independent of the state of the second site, then the two processes have the same rate law, and can be merged into one, provided the participants are merged accordingly: We form the union of EGFRuu and EGFRup and call it EGFRux (suffix "x" for "unknown", meaning, it can be "u" or "p") and likewise the union EGRFpx, and then the union process of P1a and P1b is P1(EGFRux → EGFRpx). Similarly, the processes of phosphorylation of the second site can be merged into P2(EGFRxu → EGFRxp).

There is even more reduction in complexity at larger scales. For two sites, subset relationships allow to describe a system using two processes, which would otherwise require four processes. For five sites, we would already have 32 phosphostates and 80 processes, which can be reduced to five processes using subset relationships. Note also that in rule-based modeling (Faeder et al. 2009), often a distinction is made between processes and process rules. However, by understanding processes as sets, there is no need to make such a distinction. This can be quite essential, because it is always possible that a new phospho site is discovered on a protein that participates in a process, which means that the process turns out to be having two previously unknown subsets—which may require a change of category, if there were a distinction between simple process and process rule.

In summary, using view and subset relationships, one can record all changes the data undergo from source to model element or annotation. Every model element and every annotation can be easily traced back, through view and set relationships, to the data from which it was derived, and format conversions and modeling assumptions can be identified, even if the model is changed or extended. The original data can always be consulted for verification, or as a starting point for the retrieval of more data to continue to build and annotate the model.

RDF implementation example: SBPAX and SyBiL

An example of the implementation of views and subsets is the System Biology Linker (SyBiL), which uses the bridging ontology Systems Biology Pathway Exchange (SBPAX) to relate to OWL-based formats such as BioPAX and allows mapping to XML-based modeling formats such as Systems Biology Markup Language (SBML) and Virtual Cell Markup Language (VCML). SyBiL is a Java application which uses the Jena Semantic Web Framework (http://jena.sourceforge.net) to process RDF, perform SPARQL (SPARQL Query Language for RDF, www.w3.org/TR/rdf-sparql-query/) and other queries, and automated reasoning.

Views are implemented in analogy to RDF bags. Subset, subview, view correspondence and view derivation are transitive properties, and correspondence is a sub property of derivation, and both are symmetric properties. Jena features an OWL Microreasoner which evaluates the transitivity and symmetry of properties while avoiding most of the much more expensive OWL inferences (e.g., cardinality restrictions).

SBPAX features classes which are superclasses of BioPAX classes and map to SBML and VCML types, which allows full support for set relationships between BioPAX and SBML or VCML. Set relationships between the model substances and substances identified in standard directories (e.g., UniProt, ChEBI) can be encoded in annotations compliant to MIRIAM. In BioPAX Level 2, every physical entity corresponds to a substance, and every sequence participant implies a substance subset of the former, and potentially a subset of the substances associated with other sequence participants. In BioPAX Level 3, every physical entity reference corresponds to a substance, and every physical entity derived from it is a subset substance, again with possible subset relationships among the latter. We believe that SBPAX flexibility will allow complying with any future BioPAX extension.

Other SBPAX concepts, such as complexes, are designed to be compatible with the notion of sets and subsets. In SBPAX, a complex is defined by its components and by the way these are arranged. If we say that C is a complex with components A and B, then this means that for every a of A and b of B, there is at least one c of C which consists of a and b. In other words, there is at least one way in which a and b can bind to produce an element of c, and there may be other ways that do not.

If A1 is a subset of A, then there is s subset C1 of C such that C1 contains all the complexes of an a1 of A1 and a b of B. Further, if A is the union of A1 and A2, then C is the union of C1 and C2, where C1 is defined like above and C2 contains all complexes of an a2 of A2 and a b of B.

Conclusions

Systems Biology knowledge can be found on the web and processed without tedious and error-prone manual transfer. The prerequisite is a coherent framework that bridges different sources and formats. We presented such a framework geared for building and annotating computational models based on two pillars. First, we represent substances and processes as sets, which can be related through subset relationships. Second, we group entities into views, which can be related by subview, correspondence, and derivation relationships. The framework makes it possible to guide through and record choices necessary to mold the knowledge data into model elements and annotations, and allow to trace back the model elements and annotations to the pieces of data from which the knowledge originated. This allows constant verification, updating, and extending models. At the same time, views allow an organization of the data that are independent of many technical details, which allows an intuitive understanding and helps to let the Systems Biology Cloud appears as a unified, convenient, and efficient resource. A possible implementation has been demonstrated based on the Java application SyBiL, which uses the Jena Semantic Web Framework to handle RDF and RDF-related tasks involving querying and automatic

reasoning, and the SBPAX bridging ontology, which relates to popular formats such as BioPAX, SMBL, and VCML.

We expect the approach in this study to greatly encourage the use of Systems Biology knowledge data on the web for building and annotating models, which in turn would greatly encourage increased availability of Systems Biology knowledge data.

Acknowledgments The authors would like to thank Ion Moraru and Jim Schaff for helpful discussions regarding this project, and Emek Demir and Nadia Anwar for discussions on BioPAX. The project was supported in part under grants from the National Institutes of Health: (NIH) R01 GM076570 grant (MLB); and NIH U54 RR022232 and P41 RR013186 grants (OR, MLB).

References

Belleau F, Nolin MA, Tourigny N, Rigault P, Morissette J (2008) Bio2RDF: towards a mashup to build bioinformatics knowledge systems. J Biomed Inform 41(5):706–716. http://bio2rdf.org/

Blinov ML, Ruebenacker O, Moraru II (2008) Complexity and modularity of intracellular networks: a systematic approach for modeling and simulation. IET Syst Biol 2(5):363–368

Degtyarenko K, Hastings J, de Matos P, Ennis M (2009) ChEBI: an open bioinformatics and cheminformatics resource. Current Protocols in Bioinformatics. Chapter 14:Unit 14.9. http://www.ebi.ac.uk/chebi/

Demir E, Babur O, Dogrusoz U, Gursoy A, Nisanci G, Cetin-Atalay R, Ozturk M (2002) PATIKA: an integrated visual environment for collaborative construction and analysis of cellular pathways. Bioinformatics 18(7):996–1003

Faeder JR, Blinov ML, Hlavacek WS (2009) Rule-based modeling of biochemical systems with BioNetGen. Methods Mol Biol 500:113–167

Hucka M et al (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19:524–531. http://sbml.org

Le Novère N, Finney A, Hucka M et al (2005) Minimum information requested in the annotation of biochemical models (MIRIAM). Nat Biotechnol 12:1509–1515

Le Novère N, Bornstein B, Broicher A et al (2006) BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. Nucleic Acids Res 34:D689–D691 (database issue). http://biomodels.net

Lloyd CM, Halstead MD, Nielsen PF (2004) CellML: its future, present and past. Prog Biophys Mol Biol 85(2–3):433–450. http://cellml.org

Luciano JS (2005) PAX of mind for pathway researchers. Drug Discov Today 10:937–942. http://biopax.org

Moraru II, Schaff JC, Slepchenko BM et al (2008) Virtual cell modelling and simulation software environment. IET Syst Biol 2(5):352–362. http://vcell.org

Ruebenacker O, Moraru II, Schaff JC, Blinov ML (2007) Kinetic modeling using BioPAX ontology. In: Proceedings of the 2007 IEEE international conference on bioinformatics and biomedicine, pp 339–348

Ruebenacker O, Moraru II, Blinov ML (2009) Integrating BioPAX pathway knowledge with SBML models. IET Syst Biol 3(5):317–328

Shannon P, Markiel A, Ozier O et al (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 13(11):2498–2504. http://cytoscape.org

Slepchenko BM et al (2003) Quantitative cell biology with the virtual cell. Trends Cell Biol 13:570–576. http://vcell.org

Vastrik I, D'Eustachio P, Schmidt E et al (2007) Reactome: a knowledge base of biologic pathways and processes. Genome Biol 8(3):R39. http://www.reactome.org/