Chapter 7
Automating Mathematical Modeling of Biochemical Reaction Networks

Andreas Dräger, Adrian Schröder, and Andreas Zell

Abstract In this chapter we introduce a five-step modeling pipeline that ultimately leads to a mathematical description of a biochemical reaction system. We discuss how to automate each individual step and how to put these steps together. First, we create a topology of interconversion processes and mutual influences between reactive species. The Systems Biology Markup Language (SBML) encodes the model in a computer-readable form and allows us to add semantic information to each component of the model. Second, from such an annotated network, the procedure known as SBMLsqueezer generates kinetic equations in a context-sensitive manner. The resulting model can then be combined with already existing models. Third, we estimate the values of all newly introduced parameters in each created rate law. This procedure requires that a time series of quantitative measurements of the reactive species within this system be available, because we calibrate the parameters with the aim that the model will fit these experimental data. Fourth, an experimental validation of the resulting model is advisable. Fifth, a model report is generated automatically to document the model with all of its components. For a better understanding, we will begin with an introduction to current standardization attempts in systems biology and generalized approaches for common rate equations before discussing computer-aided modeling, parameter estimation, and automatic report generation. We complete this chapter with a discussion of possible further improvements to our modeling pipeline.

Keywords Computer aided modeling · Automatic rate law generation · Model documentation · Model annotation · Model semantics · Model merging · Modeling tools · Software in systems biology

A. Zell (✉)
Center for Bioinformatics Tübingen (ZBIT), Sand 1, 72076 Tübingen, University of Tübingen, Germany
e-mail: andreas.zell@uni-tuebingen.de

S. Choi (ed.), *Systems Biology for Signaling Networks*, Systems Biology 1,
DOI 10.1007/978-1-4419-5797-9_7,

7.1 A Straightforward Modeling Pipeline

The mathematical modeling of biochemical reaction networks plays a central role in understanding the behavior of complex biological systems (Kitano 2002a; Lloyd et al. 2004) (Fig. 7.1). All these networks bear many resemblances in their structure: a set of reactions interconverts substances, often referred to as reacting species, that are located within some cellular compartment. The type of species and therefore also the type of reaction can strongly vary due to the variety of substances a cell is composed of. Some species interfere with a reaction but are neither consumed nor produced. These species are often called modulators of the reaction. This does not mean, however, that the amount of modulators cannot change, because, just like other species, modulators of one reaction can act as reactants or products in other reactions. In cellular environments, a special type of modulator, enzymes, catalyze most reactions. Other modulators speed up the reaction rate and are therefore called potentiators. If a modulator lowers the velocity of a reaction, it is referred to as an inhibitor.

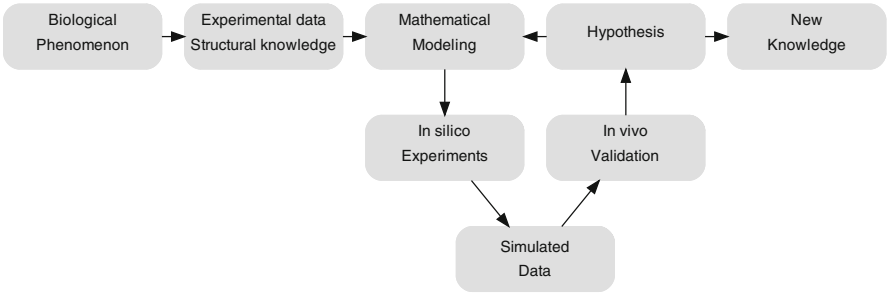


Fig. 7.1 The process of knowledge discovery in systems biology. This figure shows how research in systems biology proceeds, starting with a biological phenomenon to be investigated. In close collaboration between experimenters and modelers, new insights into the phenomenon can be discovered by iteratively performing in silico and in vivo experiments. This procedure refines the mathematical model, leading to new hypotheses and finally new biological knowledge. In this chapter we focus on the question of how to create such a model aided by a combination of dedicated software tools

In order to create a mathematical model to quantitatively describe these reaction dynamics, specialized ordinary differential equations are needed. These equations are given by kinetic equations derived from the experimental analysis of the reaction mechanism and may contain several uncertain parameters (Liebermeister and Klipp 2005). To set up a model of a biochemical system, the following five steps have to be undertaken:

1. Determination of the structure of the reaction system, the network topology.
2. Assignment of appropriate rate laws to each reaction, thereby considering all modifiers (e.g., catalysts, inhibitors, and potentiators), and the type and stoichiometry of all reactants and products.

3. Model calibration, i.e., determination of values for all the parameters within the rate laws with the aim that the dynamic behavior of the model will mirror given measurement data of the reacting species.
4. Experimental validation of the model.
5. Model documentation.

This model building process, however, continues to be a highly complicated and labor-intensive task that requires human expertise in numerous details of biochemistry and mathematics. In this chapter we explore possibilities to let automatic procedures do the work, but in some steps human intervention remains indispensable.

To tackle the first task of our modeling pipeline, we make use of databases like KEGG and MetaCyc (Caspi et al. 2008; Kanehisa et al. 2006), which provide a large set of known reaction pathways in a multitude of organisms. There we can easily look up the processes we are interested in. But before we can extract any information from these pathway databases, we have to think about the level of detail of the biological process we intend to investigate (Wilkinson 2006 p. 1). Instead of considering all reactions within the pathway map of interest, it often makes sense to lump several reactions together. To give an example of this, a chain of coupled reactions within a protein complex may proceed much faster than all other reactions and we can therefore ignore intermediate steps. Sometimes it may be desirable to abstract from the real process because, for technical reasons, the amount of certain species cannot be measured with sufficient accuracy. In other cases we may want to disregard changes in the amount of small molecules such as carbon dioxide in order to keep the system simple. Changes in the amount of water molecules can usually also be neglected because normally water is found abundantly in cellular systems. As soon as we have decided on the level of detail our model should encompass, we can use an appropriate graphical pathway modeling tool like CellDesigner (Funahashi et al. 2003) to build a reaction pathway based on the information obtained from pathway databases. CellDesigner itself provides instant access to several online databases. Additionally, tools such as KEGG2SBML[1] can assist us in this step. KEGG2SBML converts KEGG's pathway files into a format we can open in CellDesigner.

In the second step, all the reactions within the network topology have to be described by one kinetic equation. Manifold equation types have been suggested for this purpose. Some rate laws are based upon the exact mechanism of the reaction, the so-called mechanistic equations. Often these equations have been individually derived for a particular reaction. The reaction kinetics database SABIO-RK provides a large collection of curated and annotated rate equations from multiple sources (Krebs et al. 2007; Wittig et al. 2006). For many reactions there is, however, still a lack of detailed rate laws (Bulik et al. 2009). Therefore, in these cases more general equations have been suggested to provide an approximative description

[1] http://sbml.org/Software/KEGG2SBML

of the real process. Purely phenomenon-oriented equations reproduce the dynamics of the reaction without taking the reaction process itself into account, whereas semi-mechanistic equations simplify the underlying process. We call both types of simplified equations *approximative* equations to distinguish them from detailed kinetic equations. Many errors can potentially be introduced when assembling complicated kinetic equations manually. It may also be desirable to apply several types of approximative kinetic equations to the reactions and compare the differences in the dynamic behavior of the whole system (Dräger et al. 2009a). For these reasons, the second step of our pipeline requires the modeler to have a considerable level of biochemical and mathematical knowledge about the underlying reaction mechanisms and also to invest a significant amount of time and effort into this process (Ziller 2009). It is impossible to entirely automate this second step, even though this would be desirable (Dräger et al. 2008). To be beneficial for the modeler, a semi-automatic procedure that suggests rate equations for each reaction must be able to take the types of reacting species on both sides of the reaction, as well as all the types of modifiers, into account. Due to the wide variety of cellular components and different reaction types, a plethora of special cases needs to be covered by such a program. After the model has been appropriately annotated it can be integrated into already existing models.

In the third step of the modeling pipeline, the parameters within the kinetic equations are equipped with meaningful values leading to a biologically plausible dynamic behavior of the model as a whole. Often optimization procedures try to estimate the parameter values with regard to the error between model output and measured data. Many optimization methods exist and it is often difficult to choose the best method for each case. Most optimization procedures themselves provide several parameters that highly influence their performance. A systematic comparison of several procedures, together with their specific settings, improves the quality of the solution (Dräger et al. 2009a).

In the next step, an experimental validation of the result is desirable. To this end, several in silico experiments should be performed with the model that has already been determined to fit the experimental data. We can, for instance, introduce variations in the initial concentration of some substances or modify the temperature to affect the temperature-dependent parameters. Analyzing the long-term behavior of the model usually suggests another in silico experiment. All these model variants or simulation outputs should be verified by an experimenter. In additional wet-lab experiments, the predictive power of the model can be further checked by setting the experimental conditions to those from our in silico experiments. A subsequent quantification of all model components and a comparison of these new values with the predicted values is performed. Basically, two contrary results are possible, but in practice several intermediate levels can occur. First, the model simulation may provide an exact prediction of the experimental values. In this case, the model is considered valid. Second, the model predictions may deviate strongly from the new experimental data. This means that either the model or the system under study is incorrect. We have to go back and critically rethink both our model and the system. In the best case, this leads to new insights about the pathways in the system (Fig. 7.1). Furthermore, a model that reproduces experimental data can also serve

as a source of inspiration for the experimenter to perform completely different investigations on the system.

Finally, a model should be annotated and fully documented to make it reusable by other researchers even after a long time. As with the previous steps, an automatic procedure that assists the modeler in this time-consuming step is desirable (Dräger et al. 2009b; Laible and Le Novère 2007; Liebermeister et al. 2008, 2009). Such an automatic documentation procedure can also assist in the construction of the model structure and help the modeler to gain an overview of the equations within the model.

An important prerequisite for automating the modeling process is a standard data format that ensures reusability for exchange of quantitative models. A computer program can only deal with formal model representations like the Systems Biology Markup Language (SBML) (Hucka et al. 2003, 2008), which should not be written by hand but be created with the help of computer programs (Lloyd et al. 2004). In this way, modelers do not have to be concerned about particular file formats that are required to represent both mathematical and biological concepts in computer-interpretable constructs. Such software tools should also check the syntactical correctness of the model during creation to ensure consistent mathematical frameworks. To communicate the semantics of a model unambiguously to both computers and humans, controlled vocabularies like the Systems Biology Ontology (SBO) are important (Le Novère et al. 2006b). Otherwise, an automatic procedure will not be able to create rate equations in a context-dependent manner. An alternative approach would be to apply the same type of approximative rate equations to every reaction within a network (Borger et al. 2007a). However, a minimal amount of computer-interpretable knowledge is still required, for instance, in order to distinguish between different types of modification.

In the following sections, we give an overview of all the requirements of the modeling pipeline described above. We assume that the topology of the system is known and for the sake of simplicity, we also assume that the sizes of compartments and the values of all parameters stay constant during the period of simulation.

7.2 Standards in Systems Biology

Any automatic procedure that assists modelers during the model building process greatly depends on the availability of specific information about the model components and reaction types. In particular, our modeling pipeline requires that the results of each modeling step can be used directly as the input for the next procedure. For this purpose, clear and standardized semantics for all model components are important because it does matter whether, for instance, a modifier acts as an enzyme or as an inhibitor. The role of each reaction participant should be clearly and unambiguously highlighted to be interpretable for both humans and machines.

Models of biological processes are often published as a set of mathematical equations or in a descriptive form (Nickerson and Buist 2009). Both formats hamper the reproducibility and reusability of the models. In even worse cases, the publication

of a model in the form of program code shifts the focus from the model itself to the algorithmic description of how to compute and simulate it. This form of representing the model can therefore potentially hide its mathematical meaning (Lloyd et al. 2004). Furthermore, the exchange of such a model between different simulation environments may not be easy (Finney and Hucka 2003).

Often reaction pathways are encoded in figures with certain graphical shapes (glyphs) representing reactants, products, and modifiers. Several types of arcs interconnect these glyphs representing processes such as the reaction, catalysis, and inhibition. Such figures can only be interpreted by humans with the help of a figure legend and are not easily machine-readable.

Model description standards are thus required to ensure both software interoperability and model reusability. Otherwise one day the plethora of naming and drawing conventions in biology, and file formats in software development, will resemble the confusion of tongues at the Tower of Babel (Fig. 7.2) leading to the need to reinvent the wheel again and again.

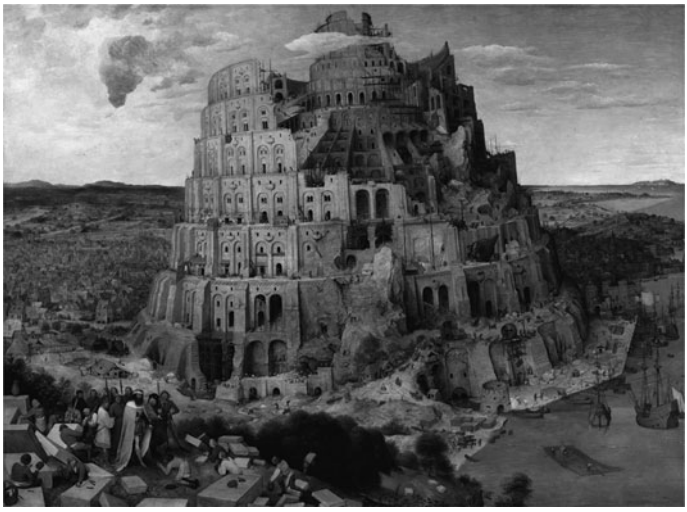


Fig. 7.2 The construction of the Tower of Babel. According to biblical legend (Genesis 11:1–9), before the Tower of Babel was built, mankind had one language and one culture. Upon seeing man's conceit and this arrogant monument to himself, God became angry and disrupted the work on it by creating such a diversity of tongues or languages that workers could no longer communicate with each other. The Tower of Babel could hence never be completed and the people were dispersed over the face of the earth. Reproduced from the oil painting "The Tower of Babel" by Pieter Bruegel the Elder, 1563 AD, with permission from the *Kunsthistorisches Museum* Vienna, Austria

Since even syntactically correct models can still be semantically incorrect, or barely understandable, a standardized form of model annotation, together with its visual representation, will allow automated validation of model consistency (Le Novère 2006; Le Novère et al. 2006b).

7.2.1 The Systems Biology Markup Language

A well-known problem in the lifetime of software is that as soon as the program is no longer being further enhanced or administrated, its internal storage standard may disappear. In the field of research such cases are especially likely to occur, because often only functional software prototypes are developed. As a consequence, models that have been created with this software and stored in their own format may no longer be readable and therefore not usable anymore. If one wants to reuse such a model in further investigations, it often has to be recreated from scratch or translated into another language or format, which is often an error-prone process. In even worse cases, models are hard-coded in a specific programming language and have therefore never been exchangeable to other software tools (Rodriguez et al. 2007).

To avoid such cases a standard format for model creation and storage is required to provide the scientific community with a highly reproducible and reusable data structure. To this end, the Systems Biology Markup Language (SBML) has been developed to store and represent various kinds of biochemical models, i.e., gene-regulatory or signal transduction pathways, and metabolic networks. (Hucka et al. 2003, 2004). As an XML-based language (eXtensible Markup Language), SBML is designed as a platform-independent, computer-readable, and tool-neutral data format (Shapiro et al. 2007). A complete documentation can be found at http://sbml.org. With more than 180 software tools that now support SBML (December 2009), it has become a widely accepted standard in the systems biology community and now defines a special mime type of the Internet Engineering Task Force (IETF) (Le Novère 2006).

The development of SBML is driven by the needs of software developers and the scientific community (Le Novère 2006). SBML thus mirrors a consensus as it does not cover all requested features but the currently most important aspects of modeling (Hucka et al. 2008). One main concept of the language is that it is organized on coexisting levels. Minor changes in the language definitions are called versions. Currently, SBML Level 1 Version 2 and SBML Level 2 Version 4 define the two language specifications and SBML Level 3 Version 1 Core Release 1 Candidate has just been proposed (December 2009) (Finney and Hucka 2003; Hucka et al. 2008).

Besides the precise XML schema for SBML, the powerful and easily usable library, libSBML (Bornstein et al. 2008), constitutes one of the main reasons for the widespread acceptance of SBML: libSBML assures software interoperability because it provides not only parsers and writers for SBML but also methods for consistency checks and interconversion methods between SBML levels and versions.

With the help of libSBML sophisticated user interfaces can be created that assist modelers when encoding their models in SBML.

CellML constitutes another freely available XML-based modeling language with an objective similar to that of SBML (Lloyd et al. 2004). Since there is a broad overlap in the scope of the two modeling languages, CellML models can be converted to SBML (Schilstra et al. 2006). However, the slightly different language structures may cause a loss of information during this conversion. A full documentation on CellML can be found at http://www.cellml.org. An advantage of CellML is that it

allows both modular and multiscale models, a feature that is also intended to be introduced into SBML with the release of Level 3. In contrast to CellML, SBML is a language tailored specifically for biochemical models and therefore serves as a *lingua franca* in this field of research (Hucka et al. 2004). Therefore, we will only focus on SBML, but similar approaches could be taken for CellML as well.

In the remainder of this section, we introduce and explain some main components of SBML Level 2 Version 4. Every SBML file starts with the declaration of the XML type used, followed by the definition of one model object, which may contain several other components each collected and defined within dedicated list objects (Listing 1). The name of each such component list starts with the prefix `listOf` followed by the plural form of the name of the component, for instance, `UnitDefinitions` or `Compartments`. In our description of SBML, we focus only on the following subset of all possible components:

Listing 1 Minimal framework of an SBML model

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <sbml xmlns="http://www.sbml.org/sbml/level2/version4"
      level="2" version="4">
3  <model id="Example">
4    <!-- optional lists of model components -->
5  </model>
6 </sbml>
```

`listOfUnitDefinitions` If undefined, SBML predefines the following units: `substance` (in moles), `volume` (in liters), `area` (in square meters), `length` (in meters), and `time` (in seconds). User-defined units can be useful in order to ensure unit balance and consistency.

`listOfCompartments` In SBML a compartment constitutes a finite reaction space, which does not have to be a cellular compartment. At least one compartment should be defined.

`listOfSpecies` The reacting substances within the model that can be referenced in reactions as reactants, modifiers, or products.

`listOfParameters` A list of model-wide valid constants or variables. A `KineticLaw` object defined within one reaction may contain a list of local parameters whose values are always constant and only valid within the particular rate equation.

`listOfReactions` Each reaction should reference species acting as reactants, modifiers, or products and may contain one `KineticLaw` object.

For a more comprehensive description, we refer the reader to the SBML specifications (Hucka et al. 2008). Note that SBML requires a designated order of all lists in the file, but all lists are optional.

The reactive species play a central role within the model. Each species has to be located inside of exactly one compartment. Therefore, a model that contains species

must also contain at least one compartment. Listing 2 declares the species S, P, and E, each within the compartment cytosol with a volume of 1 nl. To let a species take part in a reaction, the species must act as a reactant, product, or modifier of the reaction. For this purpose, a reaction contains references to the species. With the help of its unambiguous identifier, such a `speciesReference` refers to the original species in the model. Listing 3 encodes the irreversible reaction of substrate S being converted into product P catalyzed by enzyme E:

$$\mathrm{S} \xrightarrow{\mathrm{E,\,P}} \mathrm{P}, \tag{7.1}$$

with feedback inhibition of the product. To be able to simulate the model dynamically, the reaction needs a `kineticLaw` object to be assigned to it. This law is written in terms of a subset of the XML-based mathematics language MathML and it may call the values of several parameters. These parameters can be defined either globally within the model or locally within the reaction. Local parameters are only valid within the `kineticLaw` that defines them. Global parameters are not necessarily constants because they can also act as variables in the models if their `constant` attribute is set to `false`. Listing 4 encodes the following equation, the enzymatic rate law describing the *competitive inhibition of an irreversible unireactant enzyme by the product*:

$$v = \frac{\mathrm{V} \times \mathrm{S}}{K_\mathrm{S} + \frac{K_\mathrm{S}}{K_\mathrm{P}}\mathrm{P} + \mathrm{S}}\,. \tag{7.2}$$

Listing 2 Definition of compartments and species in SBML

```
<listOfCompartments>
  <compartment id="cytosol" size="1e-09" units="volume"/>
</listOfCompartments>
<listOfSpecies>
  <species id="S" compartment="cytosol"
     initialAmount="9e-15" hasOnlySubstanceUnits="true"/>
  <species id="P" compartment="cytosol"
     initialAmount="1e-15" hasOnlySubstanceUnits="true"/>
  <species id="E" compartment="cytosol"
     initialAmount="5e-15" hasOnlySubstanceUnits="true"/>
</listOfSpecies>
```

Listing 3 Definition of a reaction in SBML

```
    <listOfReactions>
      <reaction id="StoP" reversible="false">
       <listOfReactants>
         <speciesReference species="S"/>
       </listOfReactants>
```

```
      <listOfProducts>
        <speciesReference species="P"/>
      </listOfProducts>
      <listOfModifiers>
        <modifierSpeciesReference species="E"/>
        <modifierSpeciesReference species="P"/>
      </listOfModifiers>
      <!-- A KineticLaw object can be placed here. -->
      </reaction>
    </listOfReactions>
```

Listing 4 Definition of a rate equation in SBML

```
    <kineticLaw>
     <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
       <divide/>
       <apply>
        <times/>
        <ci> V </ci>
        <ci> S </ci>
       </apply>
       <apply>
        <plus/>
        <ci> Ks </ci>
        <apply>
         <times/>
         <ci> Ks </ci>
         <apply>
          <divide/>
          <ci> P </ci>
          <ci> Kp </ci>
         </apply>
        </apply>
        <ci> S </ci>
       </apply>
      </apply>
     </math>
     <listOfParameters>
      <!-- These could also be defined globally. -->
      <parameter id="V" value="1" units="mol_per_s"/>
      <parameter id="Ks" value="1" units="substance"/>
      <parameter id="Kp" value="1" units="substance"/>
     </listOfParameters>
    </kineticLaw>
```

Listing 5 Definition of a unit in SBML

```
<listOfUnitDefinitions>
  <unitDefinition id="mol_per_s">
   <listOfUnits>
     <unit kind="mole"/>
     <unit kind="second" exponent="-1"/>
   </listOfUnits>
  </unitDefinition>
</listOfUnitDefinitions>
```

For the parameter V we declare the unit `mol_per_s` as follows (Listing 5):

$$1\,\mathrm{mol\,s^{-1}} \tag{7.3}$$

to ensure the validity of the model, because every `kineticLaw` should evaluate to units of `substance` per `time`. The parameters K_S and K_P have the predefined unit `substance`, which is mol by default. The species S, P, and E also use the default substance unit because for these species initial amounts are declared and their `hasOnlySubstanceUnits` attribute is set to `true`. This flag indicates that the species always has to be interpreted in terms of molecule counts and not in terms of concentration, e.g., mol per litre. Therefore, the unit of the reaction rate is exactly the unit of parameter V.

7.2.2 The Systems Biology Ontology

From the SBML example model in the previous section we learn that there is no way to distinguish between different kinds of modifiers within a reaction (Listing 3). Looking at the SBML model alone, it remains decidedly unclear as to whether a modifier acts as an inhibitor, a catalyst, or an activator. However, as the kinetic equation must reflect the role of the modifier, implicit knowledge about this function should also be contained within the SBML model. With the aim of automatically deriving rate equations from a given network topology, this difference even becomes a prerequisite. But why does SBML not contain special objects for each kind of modification? The reason is that this semantic information goes far beyond the scope of SBML, which has been developed to store and exchange a pure description of the mathematical model, in particular, complete models that already contain kinetic equations. Fortunately, SBML offers several ways to overcome this limitation. The Systems Biology Ontology (SBO)[2] (Le Novère et al. 2006) provides the simplest way to satisfy our requirement.

[2] http://www.ebi.ac.uk/sbo

In the context of information science, an ontology is an explicit specification of a conceptualization. A conceptualization is a simplified, abstract view of the world that we wish to represent for a specific purpose. Ontologies are represented in a collection of controlled vocabulary terms and formal axioms that constrain their interpretation and guarantee the well-formed use of these terms. Each term has a unique identifier and a verbal definition. Often ontologies can be viewed in a hierarchically structured, directed acyclic graph, a taxonomic hierarchy of classes (Gruber 1993).

Figure 7.3 illustrates a part of the SBO graph up to level five. Several other ontologies have also been defined to help clarify and structure the usage of concepts in science, of which the Gene Ontology (GO) (Ashburner et al. 2000) is one of the best-known examples in biology. The SBO is similar to the GO, but only contains terms that are clearly related to systems biology. The SBO mainly aims to define relations between semantic descriptions of model components and model structure.

The SBO is organized in six vocabularies with an *is a* hierarchy of sub-classes. The *entity* class allows us to specify the material entity for species in SBML, e.g., whether a species represents a ribonucleic acid or an ion. In the *interaction* branch the SBO provides terms to distinguish between various kinds of reactions such as transport processes or degradation. Furthermore, the SBO already contains several mathematical expressions with predefined rate laws to describe specific types of reactions, each including links to other SBO terms. One of these is the branch *quantitative parameters*, which defines various kinds of kinetic parameters, and the second one is *participant role*, which characterizes products, reactants, biological activity, compartments, and modifiers like inhibitors, potentiators, or catalysts. The *modelling framework* branch is useful to ascertain if a model is to be interpreted in a continuous, discrete, or logical manner. Therefore, all kinetic equations in the SBO point to sub-categories of this branch. The model itself, however, should be annotated with a term from the *interaction* branch. The program called semanticSBML provides convenient methods to create and edit SBO annotations within an SBML file (Liebermeister et al. 2008, 2009).

With this powerful annotation at hand, an automatic procedure can, from the stoichiometry of a reaction and the knowledge of the roles and material entities of the participating species, automatically derive a list of the most suitable rate equations. As an example of what such an annotated model looks like, let us consider the definition of Reaction 1 in Listing 3. An annotated version of this definition is shown in Listing 6. This version allows an automatic procedure to clearly distinguish between the different roles of the two modifiers E and P. The SBO itself already contains many special cases of such equations. Generated rate laws and their parameters can, on the other hand, also be automatically annotated using the corresponding terms from the SBO. In this way, even automatically generated models are human- and machine-interpretable. For even more comprehensive model annotation parameters, compartments and species should also be annotated accordingly.

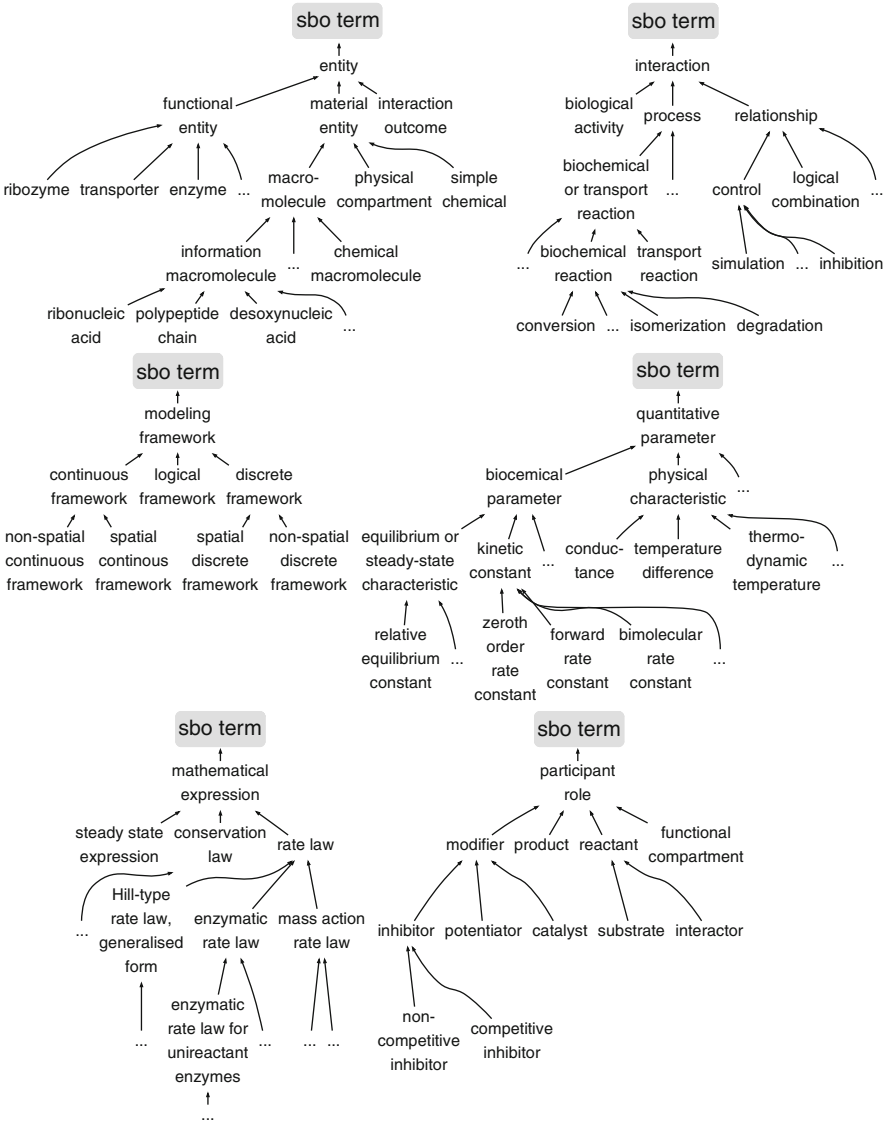


Fig. 7.3 Extract from the Systems Biology Ontology graph (June 2009). This figure shows the SBO terms up to level five in the directed acyclic SBO graph. All arcs mean *is a* and are directed toward the parent node. The nodes labeled with "..." stand for omitted terms. The graph is split into six distinct graphs, each rooted at the same node *sbo term* (highlighted in *gray*)

Listing 6 Definition of a reaction in SBML including SBO annotations

```
<listOfReactions>
  <reaction id="StoP" reversible="false"
    sboTerm="SBO:0000393"> <!-- production -->
  <listOfReactants>
   <speciesReference sboTerm="SBO:0000015"
     species="S"/> <!-- substrate -->
  </listOfReactants>
  <listOfProducts>
   <speciesReference sboTerm="SBO:0000011"
     species="P"/> <!-- product -->
  </listOfProducts>
  <listOfModifiers>
   <modifierSpeciesReference sboTerm="SBO:0000013"
     species="E"/> <!-- catalyst -->
   <modifierSpeciesReference sboTerm="SBO:0000020"
     species="P"/> <!-- inhibitor -->
  </listOfModifiers>
  <!-- A KineticLaw object can be placed here. -->
 </reaction>
</listOfReactions>
```

7.2.3 The Systems Biology Graphical Notation

Up to now, various glyphs have been used in a multitude of publications to display reaction pathways under the tacit assumption that a specific set of known symbols is widely accepted within the community. However, in a cross-disciplinary field such as systems biology, these diagrams should also be intuitively interpretable for scientists from other fields of research. To this end, the Systems Biology Graphical Notation (SBGN) explicitly defines these formerly implicitly accepted symbols in a comprehensive system and brings them together into one context with precise and visually unambiguous semantics. The SBGN consistently defines syntactic rules for the usage of all specified shapes, arcs, and arrows. In this way, interconversion between graphical models for visualization and formal models for analysis or simulation becomes possible, because all glyphs and edges are intended to correspond to specific SBO terms. Based on SBGN graph drawing software, the publication, education, or simply visualization of biological models can be easily implemented. Due to the unambiguous interpretation of the graphical information, it will also be possible to automatically extract models from printed diagrams. The SBGN homepage[3] provides the documentation of the standard for process diagrams (Kitano et al. 2005; Le Novère et al. 2008). Figure 7.4 gives an overview of several single

[3] http://sbgn.org

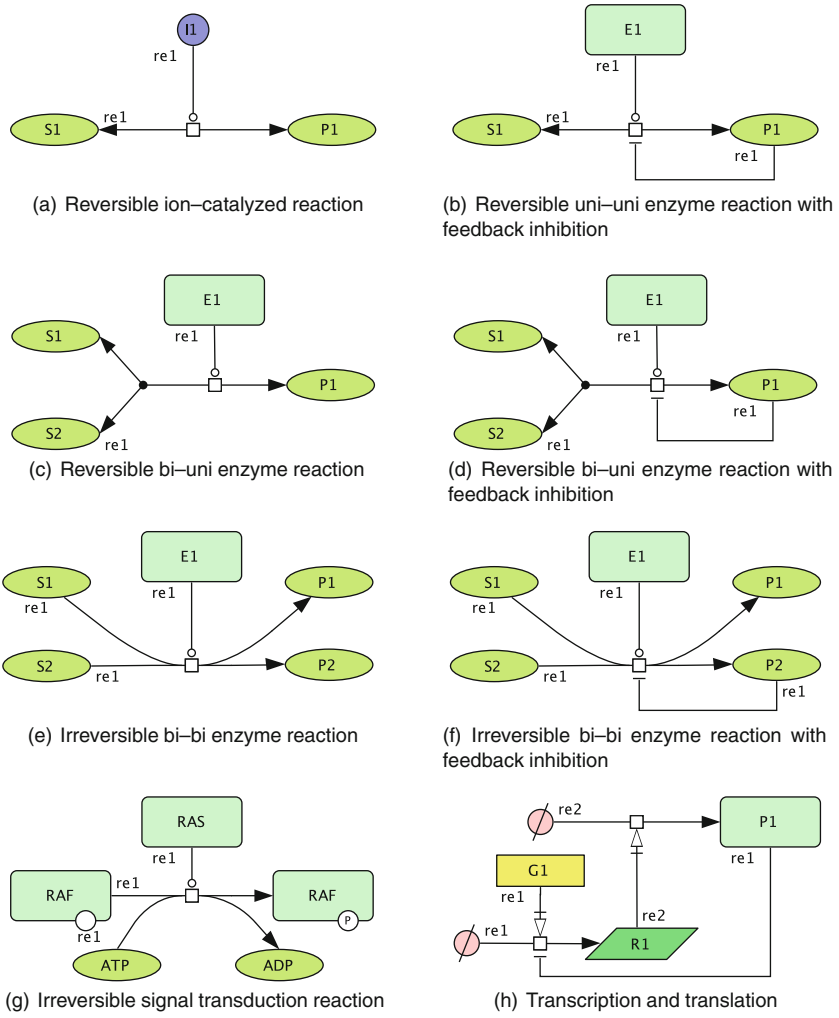


Fig. 7.4 Example reactions in systems biology graphical notation. *Ovals* represent simple molecules, whereas ions are displayed as *circles*. *Rectangles with rounded edges* denote macromolecules. Residues of these macromolecules are highlighted using *circles* at the *edges* of these glyphs (Fig. 7.4g). According to CellDesigner, genetic elements can be drawn as *rectangles*. The empty set symbol Ø illustrates sources and sinks of reactions. In CellDesigner, *parallelograms* denote RNA. Table 7.1 explains the various kinds of reaction modifications, such as catalysis, and inhibition. CellDesigner places the identifier of each reaction next to each arc, e.g., `re1`

reactions in SBGN. In our modeling pipeline, we apply the SBGN-based modeling tool, CellDesigner, which allows us to easily translate the information from pathway databases into an SBML-formatted file without having to care about SBO terms or exact SBML syntax.

7.3 Toward Generalized Rate Laws

The topology of a given reaction system corresponds to a *stoichiometric matrix* **N**, which SBML encodes implicitly. Each row in this matrix stands for one species and each column is assigned to one reaction. A positive value n_{ij} means that reaction R_j produces n_{ij} molecules of species S_i whereas negative values represent the consumption of the species. An entry $n_{ij} = 0$ means that species S_i does not take part as a reactant or product in reaction R_j. In a similar way, we define the ternary *modulation matrix* W that contains the values $w_{jm} = -1$ if species S_m inhibits reaction R_j, $w_{jm} = 1$ if S_m acts as an activator, or $w_{jm} = 0$ if no interaction takes place between R_j and S_m. In our framework we can then compute the rates of change of each species within the system as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{S} = \mathbf{N}\mathbf{v}(\mathbf{S}(t), t, \mathbf{W}, \mathbf{p}), \tag{7.4}$$

with the vector of reaction velocities or rate laws **v** that depends on time t, the vector of reacting species **S**, modulation matrix **W**, and the parameter vector **p**. In the rate laws we will consider here, **v** does only implicitly depend on time.

Every rate law v_j corresponds to one dynamic modeling framework, i.e., a description of how to interpret the equation. We classify these modeling frameworks according to a two-dimensional scheme. On the first axis we distinguish discrete and continuous equations. On the second axis we classify each rate equation to be either probabilistic or deterministic. Many kinetic equations have been proposed for every one of the four possible combinations along this scheme of modeling frameworks (Albert 2007). The purpose of our model and its desired level of detail play a central role when selecting a modeling framework together with appropriate kinetic equations.

In the remainder of this section we give a short overview of some important classes of continuous deterministic rate equations, which we can insert into Eq. (7.4). More information about specific rate laws and special cases can be found in relevant textbooks on enzyme kinetics (Bisswanger 2000; Cornish-Bowden 2004; Segel 1993) and in the SBO specifications. Here we mainly consider the reversible forms of selected rate equations because in multi-enzyme systems the rate equation of an irreversible reaction implies that the product concentrations are completely zero and do not interfere with the progress of the reaction. The assumption of having irreversible reactions originates from in vitro enzyme studies on single reactions and is, in most cases, not satisfied for multi-enzyme systems (Cornish-Bowden 2004, pp. 312–314). In several studies it has been found that the effects of the products in multi-enzyme systems play an important role (Dräger et al. 2007a,b, 2009a). Irreversible rate equations should therefore only be applied in very specific cases like the transport of a substance out of the system.

In many cases actual reaction mechanisms remain unknown and therefore reliable, specific rate equations cannot be derived (Bulik et al. 2009). With an increasing number of pathways and reactions stored in databases like KEGG or MetaCyc

(Caspi et al. 2008; Ogata et al. 2000) the discovery of the actual underlying reaction mechanisms becomes more and more important. In vitro experiments with isolated reactions of purified enzymes are often applied to uncover the individual reaction steps. The same enzyme can, however, behave differently in its cellular environment (Cornish-Bowden 2004, p. 277), leading to discrepancies between measured rate equations and its actual behavior in vivo. In the models encoded in SBML it is usually not evident how the reaction mechanism proceeds, because, for the sake of simplicity, intermediate steps are typically omitted. Therefore, the mechanism can often not be reconstructed from such a topology. Any method that assigns rate laws more or less automatically needs some level of abstraction from the reaction mechanism. Hence, what is needed are generalized rate equations that abstract from the underlying processes to a certain degree and are therefore applicable to a wider range of reactions than the equations that are derived individually for very specific reaction mechanisms.

7.3.1 Generalized Mass Action Kinetics

For cases in which the enzyme mechanism does not play the main role, or in non-catalyzed or in non-enzyme reactions, the mass action rate law can be applied. This very simple equation is derived from the assumption that the reaction probability grows proportionally to the collision probability of the reactants and is therefore proportional to the concentrations of all substrates raised to the power of their stoichiometric molecularity, i.e., the number of molecules that need to collide to initiate the reaction:

$$v_j(\mathbf{S}, \mathbf{p}) = k_{+j} \prod_i [\mathrm{S}_i]^{n_{ij}^-} - k_{-j} \prod_i [\mathrm{S}_i]^{n_{ij}^+}, \tag{7.5}$$

where $n_{ij}^{\pm}$ are the absolute values of the negative and the positive stoichiometric coefficients and the parameters $k_{\pm j}$ denote the *forward and reverse rate constants* that depend on temperature and pressure. Square brackets around a species denote the concentration of the species. In 1983 Schauer and Heinrich propose a generalized form of the mass action equation (Heinrich and Schuster 1996; Schauer and Heinrich 1983):

$$v_j(\mathbf{S}, \mathbf{W}, \mathbf{p}) = F_j(\mathbf{S}, \mathbf{W}, \mathbf{p}) \times \left(k_{+j} \prod_i [\mathrm{S}_i]^{n_{ij}^-} - k_{-j} \prod_i [\mathrm{S}_i]^{n_{ij}^+} \right), \tag{7.6}$$

in which the F_j terms are arbitrary positive functions introducing saturation and inhibition effects to the mass action equation. With this generalization it becomes possible to include the effects of activators and inhibitors even in the mass action equation. This can, for instance, be achieved by setting

$$F_j(\mathbf{S}, \mathbf{W}, \mathbf{p}) = \prod_m h_{\mathrm{A}}([\mathrm{S}_m], k_{\mathrm{A}jm})^{w_{jm}^+} h_{\mathrm{I}}([\mathrm{S}_m], k_{\mathrm{I}jm})^{w_{jm}^-}. \tag{7.7}$$

Here, the modulation matrix $\mathbf{W}$ comes into play. Accordingly, the $w_{jm}^{\pm}$ denote the absolute values of the positive and negative elements in $\mathbf{W}$. The product runs over all modulators. Both the activation function h_{A} and inhibition function h_{I} depend on the concentration of the modulating species S_m and one parameter, $k_{\mathrm{A}jm}$ or $k_{\mathrm{I}jm}$, and are defined as follows (Liebermeister and Klipp 2006):

$$h_{\mathrm{A}}([\mathrm{S}_m], k_{\mathrm{A}jm}) = \frac{[\mathrm{S}_m]}{k_{\mathrm{A}jm} + [\mathrm{S}_m]}\ , \tag{7.8}$$

$$h_{\mathrm{I}}([\mathrm{S}_m], k_{\mathrm{I}jm}) = \frac{k_{\mathrm{I}jm}}{k_{\mathrm{I}jm} + [\mathrm{S}_m]}. \tag{7.9}$$

Including both functions allows the generalized mass action kinetics to be applied to various reactions. In several publications this rate law is successfully used as an approximative equation in cases where detailed knowledge about the reaction kinetics remains unknown (Bulik et al. 2009; Dräger et al. 2007b, 2009a). It should also be noted that reactions with multiple (more than two) reactants are unlikely to take place if all molecules are required to collide by chance within the medium. In many cases these reactions contain several reaction steps each involving only two reactants at a time (Cornish-Bowden 2004, P. 6).

7.3.2 Generalizing Enzyme Kinetics

The well-known Michaelis–Menten equation from 1913 constitutes the basis of modern enzyme kinetics and has been extended several times (Cornish-Bowden 2004; Segel 1993). The full reversible form of this equation assumes the reaction mechanism pictured in Fig. 7.5, which is sometimes referred to as a uni–uni reaction because one substrate molecule is converted to one product molecule. An outstanding feature of this rate law is that it already contains terms describing the influence of inhibitors. Activation is, however, not considered in this equation but can be included by combining this equation with a function similar to $F_j(\mathbf{S}, \mathbf{W}, \mathbf{p})$ from the generalized mass action equation:

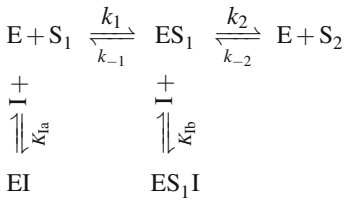


Fig. 7.5 Uni–uni reaction scheme. This reaction scheme demonstrates the underlying mechanism of an enzyme-catalyzed reaction, including the states in which an inhibitor interferes with the reaction

$$v_j(\mathbf{S},\mathbf{W},\mathbf{p}) = [\mathrm{E}]_0 \underbrace{\prod_m h_\mathrm{A}([\mathrm{S}_m], k_{\mathrm{A}jm})^{w^+_{jm}}}_{\text{activation}} \frac{\frac{k_{\mathrm{cat}+j}}{K_{\mathrm{M}j,\mathrm{S}_1}}[\mathrm{S}_1] - \frac{k_{\mathrm{cat}-j}}{K_{\mathrm{M}j,\mathrm{S}_2}}[\mathrm{S}_2]}{1 + \frac{[\mathrm{I}]}{K_{\mathrm{Ia}j}} + \left(\frac{[\mathrm{S}_1]}{K_{\mathrm{M}j,\mathrm{S}_1}} + \frac{[\mathrm{S}_2]}{K_{\mathrm{M}j,\mathrm{S}_2}}\right)\left(1 + \frac{[\mathrm{I}]}{K_{\mathrm{Ib}j}}\right)}. \tag{7.10}$$

Besides the *activation constant* $k_{\mathrm{A}jm}$, this equation contains several other parameters: the *turnover number* or *catalytic constant* $k_{\mathrm{cat}\pm j}$, the *Michaelis constants* $K_{\mathrm{M}j,\mathrm{S}_1}$ and $K_{\mathrm{M}j,\mathrm{S}_2}$ of the substrate S_1 and the product S_2, and the *inhibition constants* $K_{\mathrm{Ia}j}$ and $K_{\mathrm{Ib}j}$ that belong to the inhibitory species I. Often the initial enzyme concentration $[\mathrm{E}]_0$ is unknown and therefore avoided in the formula by introducing the *limiting rates* $V_{\pm j} = [\mathrm{E}]_0 \times k_{\mathrm{cat}\pm j}$ as new parameters. It should be noted that the vector $\mathbf{S}$ includes the amounts of all reacting species and modifiers taking part in the reaction, i.e., S_1, S_2, E, and I.

The SBO tree contains several special cases of this formula and also other enzyme rate laws that go beyond the scope of this chapter. We would like to draw the reader's attention to some special cases of enzyme kinetics with more than one substrate molecule. For those reactions, the order in which the substrate molecules bind to the enzyme strongly influences the velocity of the reaction. We distinguish basically three types of reaction mechanisms for cases of two substrate and two product molecules (bi–bi reactions): random order, compulsory order, and ping-pong mechanism. In the first case, both substrate molecules can bind to the enzyme in a random order. The products are also released randomly. In contrast, in the ordered mechanism, the order in which substrate molecules bind to the enzyme and the products are released is strictly fixed. The ping-pong mechanism works differently: the binding of the first substrate molecule induces a modification of the enzyme that enables the second substrate to bind right after the first product is released. Then the second substrate reacts to the second product, thereby recovering the enzyme for the next reaction. A rate law for these types of reactions must reflect these different reaction steps. Many special cases of rate laws are described in dedicated text books like references (Bisswanger 2000; Cornish-Bowden 2004; Segel 1993), all of which also cover the method of King and Altman (1956) to derive additional rate laws from the reaction mechanism. Many rate equations are already included in the SBO together with their formula and a short description.

In 2006, Liebermeister and Klipp derive a generalization of the Michaelis–Menten equation, known as convenience kinetics:

$$v_j(\mathbf{S},\mathbf{W},\mathbf{p}) = F_j(\mathbf{S},\mathbf{W},\mathbf{p}) \times [\mathrm{E}_j] \frac{k_{\mathrm{cat}+j}\prod_i \left(\frac{[\mathrm{S}_i]}{K_{\mathrm{M}ji}}\right)^{n^-_{ij}} - k_{\mathrm{cat}-j}\prod_i \left(\frac{[\mathrm{S}_i]}{K_{\mathrm{M}ji}}\right)^{n^+_{ij}}}{\prod_i \sum_{m=0}^{n^-_{ij}} \left(\frac{[\mathrm{S}_i]}{K_{\mathrm{M}ji}}\right)^m + \prod_i \sum_{m=0}^{n^+_{ij}} \left(\frac{[\mathrm{S}_i]}{K_{\mathrm{M}ji}}\right)^m - 1}. \tag{7.11}$$

The convenience rate law is based on the random order ternary-complex mechanism, in which two substrate molecules bind in random order to the enzyme. One important achievement of the authors in the derivation of convenience kinetics is that they

ensure the thermodynamic correctness of the system's parameters by deriving a second form of this equation for cases in which the stoichiometric matrix of the system contains linearly dependent columns, i.e., $\mathbf{N}$ does not have a full column rank:

$$v_j(\mathbf{S},\mathbf{W},\mathbf{p}) = F_j(\mathbf{S},\mathbf{W},\mathbf{p}) \times k_{\mathrm{V}j} \times [\mathrm{E}_j] \times \frac{\prod_i \left(\frac{[\mathrm{S}_i]}{K_{\mathrm{M}ji}}\right)^{n_{ij}^-} (k_{\mathrm{G}i}K_{\mathrm{M}ji})^{-\frac{n_{ij}}{2}} - \prod_i \left(\frac{[\mathrm{S}_i]}{K_{\mathrm{M}ji}}\right)^{n_{ij}^+} (k_{\mathrm{G}i}K_{\mathrm{M}ji})^{\frac{n_{ij}}{2}}}{\prod_i \sum_{m=0}^{n_{ij}^-} \left(\frac{[\mathrm{S}_i]}{K_{\mathrm{M}ji}}\right)^m + \prod_i \sum_{m=0}^{n_{ij}^+} \left(\frac{[\mathrm{S}_i]}{K_{\mathrm{M}ji}}\right)^m - 1}. \tag{7.12}$$

This condition is necessary because if the stoichiometric matrix does not have a full column rank, at least one reaction thermodynamically depends on another reaction. Note that the thermodynamically independent form shown in Eq. (7.12) replaces the parameters $k_{\mathrm{cat}\pm j}$ of the simple form in Eq. (7.11) with $\left(k_{\mathrm{G}i}K_{\mathrm{M}ji}\right)^{\mp\frac{n_{ij}}{2}}$, where the *energy constant* $k_{\mathrm{G}i}$ belongs to species S_i regardless of reaction R_j. Additionally, the whole fraction is multiplied by the *velocity constant* $k_{\mathrm{V}j}$, which is only assigned to reaction R_j. The *Michaelis constant* $K_{\mathrm{M}ji}$ contains both indices i and j and therefore belongs to one species in one reaction.

7.3.3 The Hill Equation

In 1910, Archibald V. Hill derives a purely empirical equation that describes the cooperative effects of the binding of oxygen to hemoglobin (Heinrich and Schuster 1996, p. 23; Cornish-Bowden 2004 pp. 243–245). Mathematically, Hill's equation takes the following form:

$$v_j(\mathrm{S}_i,\mathbf{p}) = \frac{V_j[\mathrm{S}_i]^{h_j}}{K_j + [\mathrm{S}_i]^{h_j}}, \tag{7.13}$$

with the *limiting rate* v_j, a constant K_j (sometimes referred to as the *half saturation constant*, e.g., by Heinrich and Schuster (1996, p. 23)), and the *Hill coefficient* h_j. For $h_j = 1$ this equation simplifies to the irreversible non-modulated Michaelis–Menten equation. Note that h_j is not necessarily an integer. It is a measure of the cooperativity effects of the enzymes, which does not depend on the number of substrate binding sites. Positive cooperative effects can be obtained with $h_j > 1$, whereas values of $h_j < 1$ lead to the much less frequently observable effect of negative cooperativity. The Hill equation is successfully used to model the effects of gene-regulatory reactions (Hinze et al. 2007).

Equation (7.13) does not take reverse reactions into account, which are, however, of great importance in the case of systems with multiple enzyme-catalyzed reactions. Athel Cornish-Bowden therefore generalizes this equation to produce a reversible form that can be applied to multi-enzyme systems and also includes the effects of modifier species M (Cornish-Bowden 2004, p. 314):

$$v_j(\mathrm{S},\mathrm{P},\mathrm{M},\mathbf{p}) = \frac{\frac{V_j[\mathrm{S}]}{k_{\mathrm{S}j}}\left(1-\frac{[\mathrm{P}]}{K_j[\mathrm{S}]}\right)\left(\frac{[\mathrm{S}]}{k_{\mathrm{S}j}}+\frac{[\mathrm{P}]}{k_{\mathrm{P}j}}\right)^{h_j-1}}{\frac{k_{\mathrm{M}j}^{h_j}+[\mathrm{M}]^{h_j}}{k_{\mathrm{M}j}^{h_j}+\beta_j[\mathrm{M}]^{h_j}}+\left(\frac{[\mathrm{S}]}{k_{\mathrm{S}j}}+\frac{[\mathrm{P}]}{k_{\mathrm{P}j}}\right)^{h_j}}, \tag{7.14}$$

with the *limiting rate* of the forward reaction v_j and the constants $k_{\mathrm{S}j}$, $k_{\mathrm{P}j}$, and $k_{\mathrm{M}j}$ describing the substrate S, product P, and modifier concentrations giving $\frac{1}{2}\ V_j$. K_j denotes the *equilibrium constant* and h_j is the *Hill coefficient*. The modifier M acts either as an activator for $\beta_j > 1$ or as an inhibitor if $\beta_j < 1$. For $\beta_j = 1$ the modification term vanishes. However, one problem with this equation is the dependence between some of its parameters.

7.4 Computer-Aided Mathematical Modeling of Biological Systems

After this introduction to two fundamental modeling requirements, standards in systems biology and generalized rate equations, we now direct the reader's attention to our modeling pipeline. We start with the graphical design of a network topology and then discuss how to automatically equip all reactions with appropriate rate equations in a context-sensitive manner using the information from the annotated process diagrams.

7.4.1 The Graphical Modeling Tool CellDesigner

The SBGN and the model storage format SBML could only become accepted standards amongst the scientific community because of the existence of user-friendly tools that provide Graphical User Interfaces (GUIs) for model creation in SBML format according to SBGN. The program known as CellDesigner was developed as a graphical process diagram editor to fulfill exactly this task as a modeling tool for gene-regulatory, signaling, and biochemical networks (Funahashi et al. 2003, 2007b). Although a growing number of other modeling tools has since become available, and many of them even provide a diagram-based GUI (Alves et al. 2006), CellDesigner remains one of the most frequently used tools amongst members of the systems biology community (Klipp et al. 2007) and it has been used in several studies for model creation (Calzone et al. 2008; Dampier and Tozeren 2007; Oda et al. 2005). The reasons for this success are probably the user-friendly nature of the tool (Fig. 7.6), the intuitive program layout, and its free availability[4] for Windows, Linux, and Mac OS.

The internal data format of all diagrams created with CellDesigner is SBML. CellDesigner does not convert the data to any other data structure. Therefore,

[4]Since August 2008, CellDesigner has been available in version 4.0.1 at http://celldesigner.org.

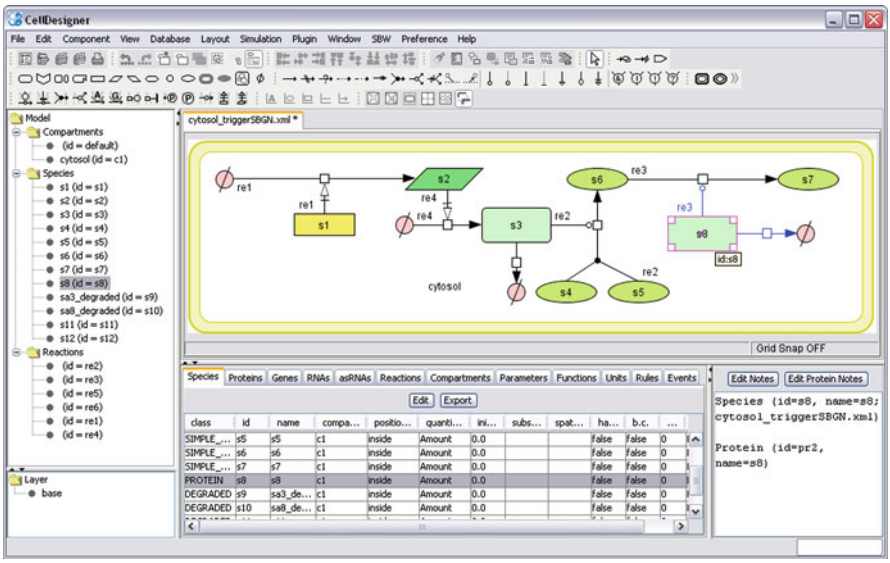


Fig. 7.6 Screenshot of the SBGN-based graphical modeling tool named CellDesigner version 4.0.1. CellDesigner's tool bar offers a large variety of glyphs representing certain types of species, compartments, state transitions, or types of modification. The user can insert any glyph by selecting it from the menu bar and just clicking on the desired position within the process diagram. The menu bar provides several options, for instance, the plug-in menu or access to online databases. On the *left-hand side*, the hierarchy of model components is shown. On the *bottom* all details about the components within the model can be viewed in well-arranged tables

CellDesigner is fully SBML compliant. The graphical layout of CellDesigner allows the user to create process diagrams (2005) that can be exported to figures in many different formats, including PDF, SVG, EPS, PNG, and JPEG. As one of the very early tools that supported graphical SBML editing, it uses the software-specific annotation tags in SBML to store its layout information in a way that can only be interpreted by CellDesigner. More recently, the SBML layout extension has been suggested and it will become an integral part of SBML Level 3 (Deckard et al. 2006; Gauges et al. 2006), but it is currently not supported by CellDesigner's SBML code.

CellDesigner is an SBW-enabled[5] (Hucka et al. 2002) program. This allows other programs to access CellDesigner's functions through the SBW broker. CellDesigner can also call methods from other SBW-enabled programs. To simulate networks created with, or imported into, CellDesigner, it contains the SBML ODE Solver (Machná et al. 2006) and COPASI (Hoops et al. 2006) can also be easily integrated. All simulation results can be exported to JPEG, PNG, and various bitmap file formats.

[5] Systems Biology Workbench (SBW)

Furthermore, specialized online databases, such as SGD[6] (*Saccharomyces* genome database) (Cherry et al. 1998), DBGET[7] (database retrieval system for a diverse range of molecular biology databases) (Fujibuchi et al. 1998), iHOP[8] (information hyperlinked over proteins) (Hoffmann and Valencia 2004), PubMed,[9] and BioModels.net[10] (Le Novère et al. 2006a), all of which can be accessed via menu items (Funahashi et al. 2007a), assist the user when setting up the topology of a reaction system. A web service integration of CellDesigner and SABIO-RK allows us to equip many reactions with enzyme kinetic equations from this powerful database (Funahashi et al. 2007a).

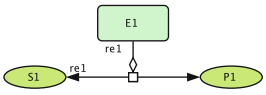
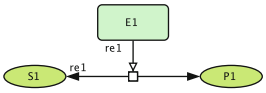
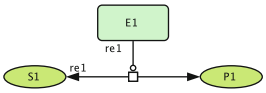
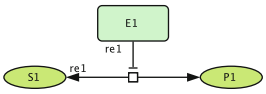
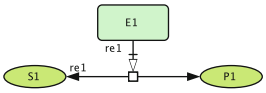
7.4.2 Context-Sensitive Assignment of Rate Equations

CellDesigner aims to provide easy methods for model creation, simulation, as well as analysis and to allow users to convert a graphically represented model into mathematical formulas for analysis and simulation (Funahashi et al. 2007b). However, its internal equation generator does not allow for context-sensitive rate law creation. Since kinetic equations can only be selected from a rather limited set of predefined formulas, the user has to manually assemble most equations, which remains a highly error-prone and cumbersome task. Furthermore, such a manual procedure runs the risk of rate equations conflicting with the SBGN representation of the process diagram. A fast comparison of several modeling approaches only becomes possible with the availability of a quick procedure that creates kinetic equations for given topologies based on the SBGN representation. The current version of CellDesigner cannot benefit from the SBO effort (Le Novère et al. 2006b) although CellDesigner contains its own annotation system to provide a lot of features to distinguish different kinds of species, reactions, and, most importantly for our purposes, modifications. At this point a specialized rate law generator is required, one which performs such a semi-automatic rate law selection based on the context of each reaction and creates the user-selected equations. Since the release of CellDesigner 4.0α in December 2006, a plug-in interface has been made available that allows developers to create a customized tool to fill this gap.

The great advantage of CellDesigner over other model creation tools is that it has been offering methods to distinguish between several different glyphs for reactions, species, and regulatory modes since before the actual SBGN and SBO were specified. Figure. 7.4 shows examples for many special cases of reversible and irreversible reactions in CellDesigner's SBGN representation, including various

[6] http://www.yeastgenome.org
[7] http://www.genome.jp/dbget
[8] http://www.ihop-net.org/UniPub/iHOP
[9] http://www.ncbi.nlm.nih.gov/pubmed
[10] http://www.ebi.ac.uk/biomodels-main

Table 7.1 Types of modulation. The exact meaning of the symbol *modulation* remains unclear. *Enzymatic catalysis* is a special case of *catalysis*. Currently, both forms of catalysis cannot be distinguished from each other based on the process diagram. Since most enzymes are proteins or RNA, we may consider a catalyst an enzyme if it belongs to these classes of molecules

SBGN connecting arc	Definition	Interpretation
	Modulation	Activation, inhibition, or catalysis
	(Physical) stimulation	Activation
	Catalysis	Catalysis or enzymatic catalysis if E_1 is, for instance, a protein or RNA
	Inhibition	Inhibition
	Trigger	Necessary activation

species types (ions, simple molecules, genes, RNA, proteins, and phosphorylation sites; empty sets as unknown sources) and the modulation symbols for feedback inhibition, catalysis, and necessary stimulation. Table 7.1 gives an overview of all the kinds of modulation arcs in CellDesigner and SBGN together with their interpretation. CellDesigner encodes this additional information within the SBML `annotation` tags, which contain software-specific information on reactions, modifiers, and species. Therefore, there exists no conflict between the SBML standard and CellDesigner's annotations. Since SBML Level 2 Version 3 has been released, the same information can also be encoded using the SBO attributes of each element.

In any case, an automatic procedure can interpret this information to suggest possible rate laws for each reaction. In particular, the annotations for different kinds of modulation are a prerequisite for such a rate law generation, because interpreting a modifier, e.g., as an inhibitor, makes a big difference to the interpretation of the same species to act as a potentiator. The special reaction types *transcription*, *translation*, and regular *state transition* may also deserve special attention. It should also be considered that each species type may act differently. A reaction that takes place on the surface of a metal plate, e.g., with an ion species as the catalyst, Fig. 7.4a can hardly fulfill the properties of an enzyme-catalyzed reaction (Fig. 7.4b–g), because the metal cannot, for instance, pass the substrate from one catalytic center to another. In contrast, several other species, especially proteins, complexes, and RNA, may show enzyme-like behavior under certain circumstances. A rate law generator must thus take the annotation of the species into account. The stoichiometric structure

of the reaction also plays an important role. For enzyme-catalyzed reactions with one reactant and one product (uni–uni), we may apply a Michaelis–Menten-based kinetic equation. The convenience rate law constitutes a generally applicable choice for all enzyme-catalyzed reactions with any number of reactants and products. In other cases, the mass action rate law can be selected. Note that the catalyst may be omitted from the process diagram for the sake of simplicity, but the reaction may still be considered to be enzyme-catalyzed.

What still cannot be distinguished from a process diagram, even with SBO annotation, are both the order in which the reactants bind to the catalyst and the order in which products are released into the surrounding medium. Therefore, for bi–uni and bi–bi reactions (Fig. 7.4c–f) or even more complicated stoichiometries, any automatic procedure can only suggest the kinetic equation for compulsory or random order reactions or the substituted enzyme mechanism (also known as ping-pong mechanism) in the case of two reactants and two products. The knowledge of the user is still required to select one of these equations.

Another special case of reactions are those that involve genes or mRNA, the so-called gene-regulatory reactions. Often these reactions show cooperativity or saturation effects because several transcription factors determine the expression level (the mRNA concentration) of genes, a behavior that can be modeled with Hill-like equations (Hinze et al. 2007) or a zeroth order mass action rate law in the case of the absence of any modifiers (basal transcription rate).

The fact is often neglected that gene expression, i.e., the formation of RNA molecules, is a complex reaction process. Actually, this process is a reaction of nucleic acids to produce RNA molecules and requires genes as template. In many cases transcription factors, i.e., specialized proteins, inhibit or enhance the transcription process. The SBGN suggests that one should illustrate this overall process in order to be consistent with the example in Fig. 7.4h. Without SBO annotation, the SBML code corresponding to this cascade only describes two distinct uni–uni reactions with several modifiers. This information is insufficient for an automatic rate law generator, because such a procedure relies on annotations or SBO attributes to point out that the reactant is actually an empty set, the product of the first reaction is an RNA molecule, and that the modifiers are a gene and some stimulating or inhibiting proteins. Therefore, a controlled vocabulary together with a standardized diagrammatic network representation are required for all these attempts on automation.

7.4.3 SBMLsqueezer

SBMLsqueezer is an application that was designed to perform context-sensitive rate law generation as described above (Dräger et al. 2008). SBMLsqueezer is a freely available plug-in for CellDesigner and entirely written in Java™. Up-to-date versions of the source code and binaries for this application can be found at the project homepage.[11]

[11] http://www.ra.cs.uni-tuebingen.de/software/SBMLsqueezer

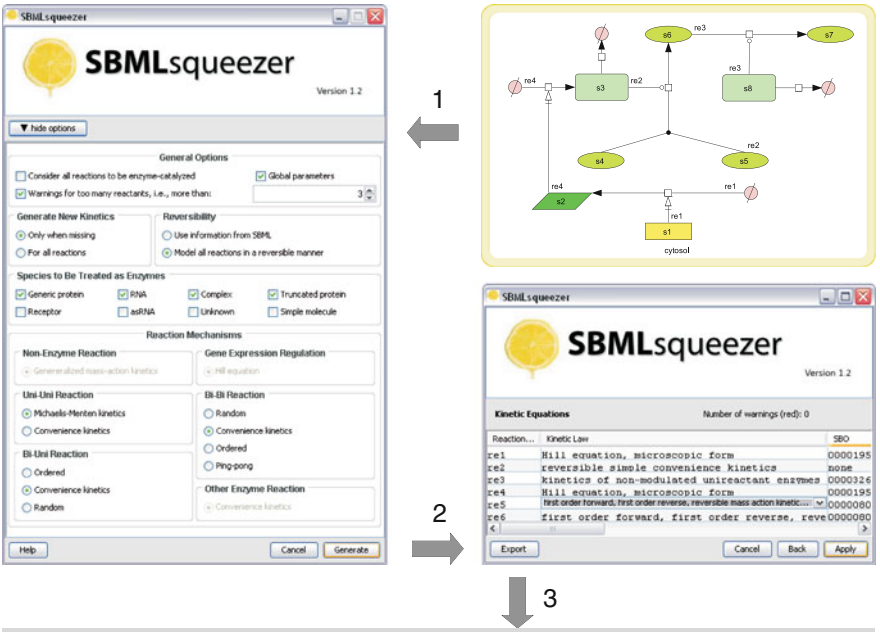
SBMLsqueezer interprets the context of the reactions in accordance with the SBGN representation to assemble lists of applicable kinetic equations for each reaction, thereby supporting most of the continuous rate laws defined in the SBO. Moreover, it suggests several rate laws that are not yet specified by the SBO consortium, especially those that were introduced in Section 7.3. One of these is the convenience rate law (Liebermeister and Klipp 2006). Other examples include the kinetic equations for the detailed ternary-complex mechanisms with random and compulsory order as well as the substituted enzyme mechanism. Each generated equation reflects the influence of modifiers that inhibit or activate the reaction. SBMLsqueezer always sets the `boundaryCondition` flag of genes to `true` to avert genes being consumed or produced within reactions. It highlights improper usage of transcription or translation arcs. Kinetic equations can even be generated in the case of non-integer stoichiometries of the reaction participants. All equations are written to SBML in MathML format, an XML-based and machine-readable code, hardly understandable or writable for humans.

SBMLsqueezer can be used in two different ways. First, it is possible to assign rate equations to all reactions within the model in a single effort (Fig. 7. 7). Second, the reaction context menu allows the user to apply SBMLsqueezer to single reactions (Fig. 7.8). In either case, the user can change all suggestions made by the program. In the current version, SBMLsqueezer presents the SBO term of its created equations but is unable to assign these to the model because the feature is lacking in CellDesigner.

One of the problems one may encounter when using SBMLsqueezer is the following: for SBML models not created in CellDesigner, a default annotation is added upon import into CellDesigner. Species become proteins, and reactions are always state transitions, and knowledge about the type of modulation can get lost. In these cases SBMLsqueezer may provide unexpected results. With the increasing use of, and support for, SBO, even in CellDesigner, this problem will vanish. In the future, SBMLsqueezer will be extended with additional rate laws like power law approximations (Savageau 1969a,b, 1970), loglin (Hatzimanikatis and Bailey 1996; Hatzimanikatis et al. 1996), or linlog (Visser and Heijnen 2002, 2003) kinetics and a stand-alone version will be released that interprets SBO attributes instead of the CellDesigner-specific annotation tags. This way, SBMLsqueezer can then be integrated as an equation-generating core into several applications. This could be used, for instance, to generate and evaluate the dynamic behavior of several versions of one model – each based on a different type of rate equation.

7.4.4 Model-Merging Using MIRIAM Annotations

Various SBML models are publicly available in curated repositories like the BioModels.net database (Le Novère et al. 2006; Le Novère et al. 2006a) or JWS online (Olivier and Snoep 2004). By modifying and combining existing pathway models, more comprehensive models of the cell can be built (Snoep et al. 2006).



1 Rate Laws

1.1 Reaction: re1, Hill equation, microscopic form

$$v_5 = V_5 \cdot \frac{[s_1]^{h_{+5:s_1}}}{[s_1]^{h_{+5:s_1}} + k_{S+5:s_1}} \tag{1}$$

1.2 Reaction: re2, reversible simple convenience kinetics

$$v_1 = [s_3] \cdot \frac{k_{cat+1} \cdot \frac{[s_4]}{k_{M1:s_4}} \cdot \frac{[s_5]}{k_{M1:s_5}} - k_{cat-1} \cdot \frac{[s_6]}{k_{M1:s_6}}}{\left(1 + \frac{[s_4]}{k_{M1:s_4}}\right)\left(1 + \frac{[s_5]}{k_{M1:s_5}}\right) + \frac{[s_6]}{k_{M1:s_6}}} \tag{2}$$

1.3 Reaction: re3 , kinetics of non-modulated unireactant enzymes

$$v_2 = [s_8] \cdot \frac{\frac{k_{cat+2}}{k_{M2:s_6}}[s_6] - \frac{k_{cat-2}}{k_{M2:s_7}}[s_7]}{1 + \frac{[s_6]}{k_{M2:s_6}} + \frac{[s_7]}{k_{M2:s_7}}} \tag{3}$$

$$\vdots$$

Fig. 7.7 Creating rate laws for the whole network in a single step, adapted from Dräger et al. (2008). If started from the *plugin* menu, SBMLsqueezer can create all rate equations for the whole model at once. By default, all reactions are set to be reversible and modeled accordingly. Already existing equations are not overwritten and parameters are stored globally. All these defaults can be changed by clicking on *show options*. Details on all suggested equations, including the SBO number if available, are then presented in a table. Double-clicking on an equation's name allows the user to select a different formula for the reaction. A single click elsewhere in any row shows an equation preview

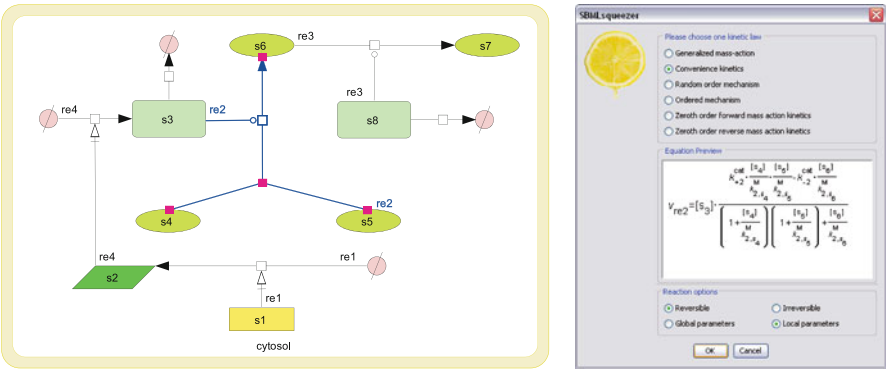


Fig. 7.8 SBMLsqueezer's reaction context menu, adapted from Dräger et al. (2008). The selection of suggested kinetic equations can change if the user sets the reaction from reversible to irreversible or the other way around. SBMLsqueezer shows an equation preview using HotEqn (available under the terms of GPL Version 3 at http://www.esr.ruhr-uni-bochum.de/VCLab/software/HotEqn/HotEqn.html) for all generated or existing kinetic equations. With the radio buttons at the bottom, the user can select whether to store parameters globally or locally for the particular reaction. Note that some kinetic equations, like convenience kinetics, contain parameters that belong to the species and have therefore to be stored globally. SBMLsqueezer treats these parameters correctly even if *local parameters* is selected. Already existing kinetic equations are also displayed to the user. To avoid inconsistencies for these equations, neither reversibility nor the way parameters are stored can be changed

However, when models have processes or substances in common and do not provide special interfaces for combining them, the merged output model will contain redundant elements. If the original models contain different mathematical statements (values, formulas, kinetics) for these elements, conflicts will arise and for each such element, one of the statements will have to be selected. Thus, key tasks in model-merging are (i) matching all redundant elements and (ii) recognizing and resolving all conflicts between them, in order to obtain a single consistent element in the output model (Liebermeister 2008). In severe cases, models may disagree so strongly in their way of describing biochemical processes and substances that merging is impossible. Serious conflicts arise when a model contains lumped reactions or substances (e.g., variables representing several modifications of one substance), especially if models describe the same processes at different levels of detail.

Software tools can assist the user by detecting redundant elements, highlighting the conflicts, suggesting possible solutions, or resolving the conflicts automatically according to rules (e.g., based on a priority ranking of the input models). A basic requirement for matching the elements is that their biological meaning be recognizable automatically; comparing elements by their names would not suffice because the input models may adopt different naming conventions.

The biochemical meaning of model elements (for instance, the fact that a certain `species` element represents water) can be declared in SBML in terms of annotations compliant with the standard MIRIAM (Minimal Information Required In the

Annotation of Models). MIRIAM is a set of guidelines for the annotation and curation processes of computational models, which facilitates their exchange and reuse (Laible and Le Novère 2007; Le Novère et al. 2005). Such an annotation points to an entry in web resources such as public databases or database ontologies. Water, for instance, can be represented by the entry `CHEBI:15377` in the ChEBI database[12] (Degtyarenko et al. 2008) and by the compound identifier `C00001` in the KEGG database. A number of official data types for annotation are listed in the MIRIAM resources[13] (Laible and Le Novère 2007; Le Novère et al. 2005). To declare that the SBML `species` A represents water, an `annotation` tag as shown in Listing 7 should be inserted into the declaration tag of `species` A.

Listing 7 MIRIAM annotation for species A

```
<species id="A" name="H2O" compartment="cytosol"
    metaid="substanceA">
 <annotation>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22
       -rdf-syntax-ns#" xmlns:dc="http://purl.org/dc/
       elements/1.1/" xmlns:dcterms="http://purl.org/
       dc/terms/" xmlns:vCard="http://www.w3.org/2001/
       vcard-rdf/3.0#" xmlns:bqbiol="http://biomodels.
       net/biology-qualifiers/" xmlns:bqmodel="http://
       biomodels.net/model-qualifiers/">
   <rdf:Description rdf:about="\#substanceA">
    <bqbiol:is>
     <rdf:Bag>
      <rdf:li rdf:resource="urn:miriam:obo.chebi:CHEBI%
        3A15377"/>
      <rdf:li rdf:resource="urn:miriam:kegg.compound:
        C00001"/>
     </rdf:Bag>
    </bqbiol:is>
   </rdf:Description>
  </rdf:RDF>
 </annotation>
</species>
```

The RDF (Resource Description Framework) syntax allows the modeler to specify a relation between the model element A and the database entry B in more detail: various possible relations (like A *is exactly* B, and A *is an instance of* B, and A *contains* B as a physical part) can be declared by BioModels.net qualifiers (Le Novère et al. 2006b).

[12] Database of chemical entities of biological interest, http://www.ebi.ac.uk/chebi

[13] http://www.ebi.ac.uk/miriam

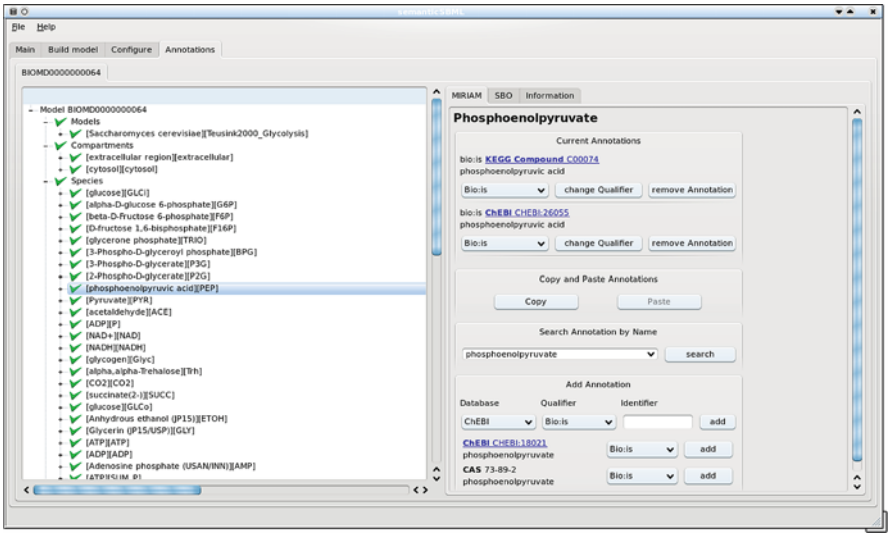


Fig. 7.9 Annotating an SBML model with semanticSBML (screenshot). All SBML elements that can be annotated are shown in a tree on the *left*. *Tickmark icons* indicate that an element has MIRIAM annotations or SBO terms, and *flag icons* mark elements without annotation. Annotations and SBO terms for a selected element can be edited on the *right*

The web application Saint automatically annotates uploaded models (Lister et al. 2009). A program known as semanticSBML[14] (Fig. 7.9) allows modelers and model curators to edit MIRIAM-compliant annotations and SBO terms and to merge several SBML models (Liebermeister et al. 2009). In the annotation section, the user can browse through the elements of a model and edit the annotations. The tool checks existing annotations automatically for syntactic correctness, based on the list of database resources specified by MIRIAM (Le Novère et al. 2005). New entries for MIRIAM-compliant annotations and possible SBO terms can be found easily by a keyword search.

In the merging section, the program generates a preliminary version of the merged model based on the MIRIAM annotations of all elements in the input models (Fig. 7.10). The merged model structure can be manually refined by the user to create a biologically meaningful structure. Each element of the merged model can be selected in an overview. The properties of a selected element are listed in a detailed view. This view groups the properties and highlights conflicts that arise through the merging of the elements. For instance, matching `species` elements can have different initial values. The user can resolve each conflict manually or, alternatively, let the program automatically resolve all conflicts. SemanticSBML makes its decisions based on a list of model priorities supplied by the user: In cases of conflicts, the program chooses the property value of the model with the highest priority. After

[14] http://www.semanticsbml.org

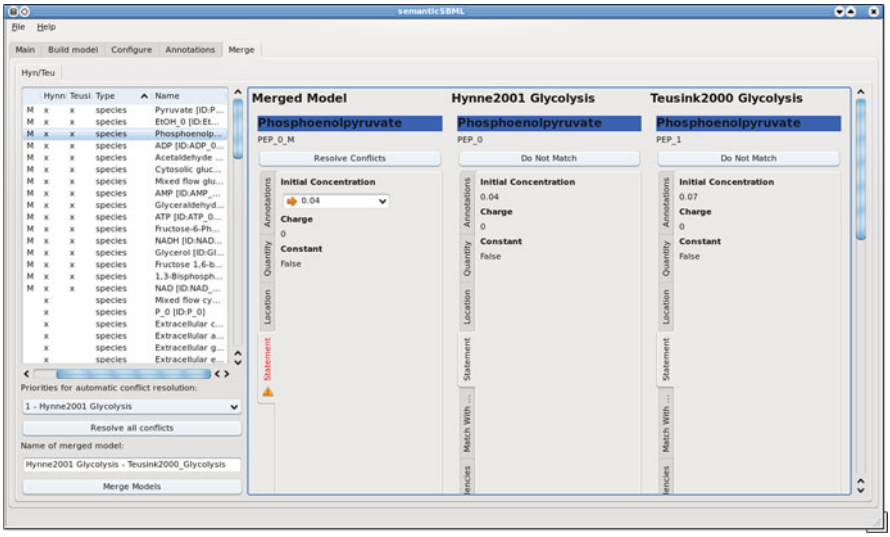


Fig. 7.10 Model-merging with semanticSBML (screenshot). On the *left*, matching elements from all input models are aligned with each other. Details about a selected element (as described by the input models and by the output model in its preliminary form) are shown on the *right*. The user can edit their properties and change the matching between elements until the models are ready for merging

all conflicts have been resolved, the merged model can be exported into the SBML format. An online version of semanticSBML with limited functionality is accessible at the project homepage.[15]

7.5 Obtaining Model Parameters

When the topology of the system under study is known, the model is annotated, and kinetic equations are defined for all reactions within the system, but one question remains: How does one determine all the parameters belonging to the equations in the model, e.g., Michaelis constants, limiting rates, or the constants describing the influence of inhibitors and activators? By default, SBMLsqueezer sets all newly created parameters to one. In contrast to CellDesigner's default of zero, this allows a direct simulation of the model. The argument to justify this decision is that in many rate equations, parameters that are set to zero lead to trivial terms within the mathematical expression or to even more questionable results if division by these parameters becomes necessary. Let us consider Eq. (7.2) on p. 9 to illustrate both problems. The whole expression equates to zero when setting parameter $V = 0$. For

[15] http://www.semanticsml.org

$K_P = 0$, the rate equation becomes undefined. However, both cases are avoided if the parameter value is set to one.

Nevertheless, default values do occasionally constitute a good choice for a parameter value since it is usually desired that the model reproduces experimental data and allows predictions of unmeasured values. Many parameters such as Michaelis constants or first-order rate constants can be determined experimentally. This requires expensive and time-consuming experiments and is not possible for all parameters. Two alternative ways exist to obtain meaningful parameter values for newly created rate laws: one way is to select known parameter values from freely available online databases such as the Brunswick Enzyme Database BRENDA (Barthelmes et al. 2007; Chang et al. 2009; Schomburg et al. 2002). The second way is to estimate the parameter values with respect to given measurement data. Like Borger et al. (2007b), we suggest a combination of both strategies.

The values within enzyme databases often correspond to in vitro experiments of isolated reactions. Hence, the values of these parameters are obtained with respect to the dynamics of the particular reaction. Often experimental conditions vary from reaction to reaction, because, e.g., different pH buffers are needed to stabilize the enzyme. When combining many such reactions in a multi-enzyme model it can therefore not be ensured that the in vitro parameter values lead to realistic in vivo dynamics of the overall system (Teusink et al. 2000). Subsequent adjustment of the parameter values is therefore advisable. An advantage of enzyme databases is that these often contain parameter values of the same reaction from multiple organisms. Hence we can gain – besides meaningful initial parameter values – additional statistical properties of many parameters like lower and upper bounds, averages, or medians. For some parameters we can even guess a distribution. These key features of the parameters should be considered in subsequent optimizations.

In this section we describe how naturally inspired heuristic optimization procedures can be used to tackle this task. The basic idea of the parameter estimation approach is that the parameters should be adapted in such a way that the difference between the simulation result of the model and the given measurement data becomes minimal. This distance serves as a quality measure for a given parameter set. Heuristic optimization procedures try to minimize this distance by iteratively sampling from the parameter space and simulating the model. The parameter set leading to the smallest distance between experimental data and model output is called *optimal* with respect to the data. This procedure is often referred to as model calibration.

In the first step, one needs to choose an appropriate distance function as a measure of quality for the parameters: the target function $f(\mathbf{p})$ that depends on the parameter vector $\mathbf{p}$. Minimization of the target function is defined as the search for a vector $\mathbf{p_0} \in P$ in the set of possible solutions P satisfying $\forall \mathbf{p} \in U \subseteq P : \; f(\mathbf{p}) \geq f(\mathbf{p_0})$, with U being a connected set. We call $\mathbf{p_0}$ a *global optimum* of f if $U = P$ and a *local optimum* of f otherwise. Several studies mention the relative squared error (RSE)

$$f_{\mathrm{RSE}}(\hat{\mathbf{x}}(\mathbf{p}), \mathbf{X}) = \sum_{i=1}^{\dim \hat{\mathbf{x}}(\mathbf{p})} \sum_{t=1}^{\dim \tau} \left(\frac{\hat{x}_i(\tau_t, \mathbf{p}) - x_{ti}}{x_{ti}} \right)^2 \tag{7.15}$$

as a useful distance function (Spieth et al. 2004, 2005a,b, 2006b). Here, $\hat{\mathbf{x}}$ is the model output vector and the matrix $\mathbf{X} = (x_{ti})$ contains all experimental data. The vector τ contains the time points from the experiment. The first sum runs over all measured state variables in the model (reacting species, variable parameters, and compartments) and the second sum runs over all time points. Missing values are ignored. The advantage of f_{RSE} over other distance measures like the Euclidean distance consists in the weighted difference between model output and experimental data for each state variable. In biological systems, certain substances occur in much higher concentration than others. An appropriate quality measure has to take this into account, or otherwise more highly concentrated species would dominate the parameter optimization procedure. Furthermore, f_{RSE} is a dimensionless quantity, which becomes especially important if state variables are of mixed types like compartment sizes and amounts or concentrations of substances. To avoid division by zero a sufficiently high default value is returned instead of the fraction in cases in which $x_{ti} = 0$.

Many optimization approaches have been proposed and applied to the parameter estimation problem in biochemical pathways (Balsa-Canto et al. 2008; Rodriguez-Fernandez et al. 2006a,b). The simplest randomized method, Monte Carlo, virtually rolls the dice multiple times and memorizes the best solution found, whereas deterministic approaches try to perform a clever and almost exhaustive search in the space of possible solutions using, for instance, a grid search that can be combined with a branch-and-bound strategy. Regarding the cost-performance trade-off, one class of optimization procedures has been found to be especially promising: naturally inspired heuristic optimization procedures, which traverse the search space using a heuristic strategy to "look" at a larger set of potential solutions in parallel. Many evolutionary algorithms that are inspired by Darwinian evolution maintain several *individuals*, each representing a possible solution, in such a *population* and use mechanisms like crossover and mutation to exchange parameter values between their individuals to increase the chance of escaping local optima. Examples are the genetic algorithm (Holland 1975), the evolution strategy (Rechenberg 1973), and differential evolution (Storn 1996). Other approaches mimic the way a mountaineer ascends a mountain (hill climbing, suggested by Tovey (1985)) or simulate the formation of crystal structures in metallurgy (simulated annealing, introduced by (Kirkpatrick et al. 1983)). More recent approaches try to emulate the swarm behavior of fishes or birds (particle swarm optimization (Clerc and Kennedy 2002; Clerc 2005)). Hybrid strategies have been suggested to combine the advantages of global and local strategies (Balsa-Canto et al. 2008; Rodriguez-Fernandez et al. 2006b). The great advantage of naturally inspired algorithms is that these procedures are capable of optimizing even highly multimodal[16] and non-convex target functions. Model calibration often belongs to this class of problems due to the high nonlinearities of the rate laws.

[16] An optimization problem is called *multimodal* if it contains a large number of local optima.

The performance of optimization procedures can be influenced by several settings of the particular algorithm. Examples are temperature in simulated annealing or population size in population- or swarm-based approaches. The large number of available optimization procedures, together with their various specific settings, leads to the question: Which method actually estimates the parameters most successfully (Banga 2008; Dräger et al. 2007a,b; Moles et al. 2003)? Taking this one step further, an additional question arises: Which approximative rate law leads to the best results when the model is calibrated to experimental data?

A study (Dräger et al. 2009a) investigates both questions, with seven variants of one exemplary network, based on four different rate laws: generalized mass action, Michaelis–Menten, convenience kinetics, and the stochastic Langevin equation (Gillespie 2000). For each model, except the stochastic one, there is one counterpart in which all reactions are considered reversible. Note that each model contains a different number of parameters.

According to William Occam's famous dictum from the 14th century, "shave away all that is unnecessary," also known as *Occam's razor*, the model with the smallest number of parameters for an adequate representation of data constitutes the preferred choice. This principle is called the *law of parsimony* in science. The reason for this strategy is that the fit of any model can be increased by introducing a higher number of free parameters. However, a model that is too simple may miss important treatment effects in experimental settings. Therefore, there is a tradeoff between model complexity and parsimony (Burnham and Anderson 2002, pp. 29–37).

To evaluate both the dynamic properties of all seven model variants and the performance of various parameter optimization methods, several heuristic optimization procedures (Monte Carlo optimization, hill climber, simulated annealing, genetic algorithm, evolution strategy, differential evolution, particle swarm optimization, and tribes) are systematically evaluated and benchmarked with alternative settings for each dynamic model. More details about each of these optimization methods can be found in the additional material of the study.[17] As an example network, the authors pick the well-investigated biosynthesis of the amino acids valine and leucine in *Corynebacterium glutamicum*. All seven models are calibrated based on in vivo measurements of metabolite concentrations along the pathway (Magnus et al. 2006).

The authors conclude that the settings-free and, therefore, easily usable tribes algorithm yields good results for first optimization attempts. Table 7.2 summarizes the most successful optimization procedures together with their specific settings. Furthermore, the reversible forms of the generalized mass action and convenience kinetics are found to be the most promising approximative approaches for the kinetics of each reaction when compared to modeling using kinetic equations for irreversible reactions (Dräger et al. 2009a). As explained in Section 7.3, in multi-enzyme systems, all reactions should be considered reversible.

[17] A comprehensive introduction to naturally inspired heuristic optimization procedures can be found at http://www.biomedcentral.com/imedia/8946429342473639/supp1.pdf

Table 7.2 Most successful settings for heuristic optimization procedures. Shown are five promising optimization procedures together with their best settings according to a benchmark study (Dräger et al. 2009a)

Algorithm	Settings	Population size
Particle swarm optimization	$\phi_1 = \phi_2 = 2.05$ on linear3, star, or grid3 topology	25
Differential evolution	The triplet (f, λ, CR) should be set to (0.8, 0.5, 0.3), (0.8, 0.5, 0.5), or (0.8, 0.8, 0.3)	100
Evolution strategy	Covariance matrix adaptation as mutation operator without crossover	(5+25) or (10, 50)
Binary genetic algorithm	Adaptive mutation and one-point or no crossover at all	250 or 100
Tribes	–	–

The freely available workbench EvA2[18] (Kronfeld 2008; Streichert and Ulmer 2005; Ulmer 2005) contains several naturally inspired heuristic optimization procedures, including those presented in Table 7.2 and has therefore been integrated into software tools such as JCell for the inference of gene-regulatory networks (Spieth et al. 2006a) or the Systems Biology Toolbox (SBToolbox2) (Schmidt and Jirstrand 2006; Schmidt 2007; Schmidt et al. 2007). For our purposes, the combination of EvA2 with the MATLAB™-based SBToolbox2 provides several useful features. We can import our SBML file into the SBToolbox2 and apply its very fast integration routine as well as several analytical functions. Access to the optimization procedures of EvA2 is well described on the project homepage.[19] EvA2 provides a well-documented Application Programming Interface (API) and can therefore be easily included as the parameter estimator into an automatic modeling pipeline.

7.6 Generation of Model Reports with SBML2LaTeX

Once a mathematical description of the biochemical network has been created, it is desirable to create a human-readable report about the model as a whole. Such documentation can enhance the model's re-usability. The best way to create a model report would be to obtain the necessary information directly from the SBML file without writing additional text. Similar approaches exist for modern programming languages like Java™, for which a complete API documentation can be created using the specialized tool JavaDoc. The idea of JavaDoc is basically to extract the existing comments from the source code and to put them in context with the methods the objects provide in a human-readable text format, e.g., HTML. In this way, the

[18] http://www.ra.cs.uni-tuebingen.de/software/EvA2

[19] The homepage http://www.sbtoolbox2.org freely provides SBToolbox2 for MATLAB™ including a comprehensive documentation.

double work of writing comments and documentation, which often leads to inconsistencies between the documentation and source code after modification of one or both, can be avoided.

In the object-oriented language SBML, all components inherit from the abstract class `SBase`. This class passes the two optional tags `notes` and `annotation` to every SBML component. The `notes` tag allows modelers to insert comments, descriptions, and other information into each SBML element in XHTML format, intended to be displayed to human readers. Additionally, in the `annotation` field, modelers can place MIRIAM descriptions pointing to external resources (Le Novère et al. 2005). With the help of annotations and notes, knowledge comparable to that of comments in source code can be assigned to every SBML component in two different ways.

SBML models contain some information only implicitly, e.g., the ordinary differential equation system of the time-dependent change of the concentrations or the amounts of all species. These equation systems are defined by the reactions each species participates in together with its `constant` and `boundaryCondition` fields. In this system, the rate of change of some species is not determined by reactions at all but by other SBML constructs such as `events` or all kinds of `rules` that affect this species. The size of compartments or the value of variable parameters can also change due to rules or events, and `initialAssignments` can override the initial values of all elements. In many cases, these complex coherences are not directly apparent from the SBML file. Furthermore, SBML itself was never intended to be a language that is written or read by humans directly. It is designed as a computer-readable and writable modeling language.

Furthermore, it may happen that, during the modeling process, a user does not carefully annotate some of the components in the model or that the units of the components turn out to be inconsistent.

For these reasons, the SBML community provides two helpful applications. First, the SBML validator[20] checks the consistency and validity of uploaded models. Second, SBML2LaTeX (Dräger et al. 2009b) contains this SBML validator and weaves its results into its comprehensive model report, in which all components of the model are clearly arranged in tables or written text. SBML2LaTeX extracts all information including notes, MIRIAM annotations, and SBO terms from the SBML file and puts it all together. If the model contains any SBO terms, a glossary with term name and definition appears at the end of the report. Its convenient online version directly creates various human-readable files, including PDF, PS, DVI, or LaTeX (Fig. 7.11). Writing these often complex mathematical expressions from SBML files into external documentation tends to be especially error-prone. However, these can be directly adopted from the generated LaTeX file. The report file also helps modelers to identify potential errors in the model or insufficient annotation that cannot be detected from a pure validity check alone.

[20] http://sbml.org/Facilities/Validator

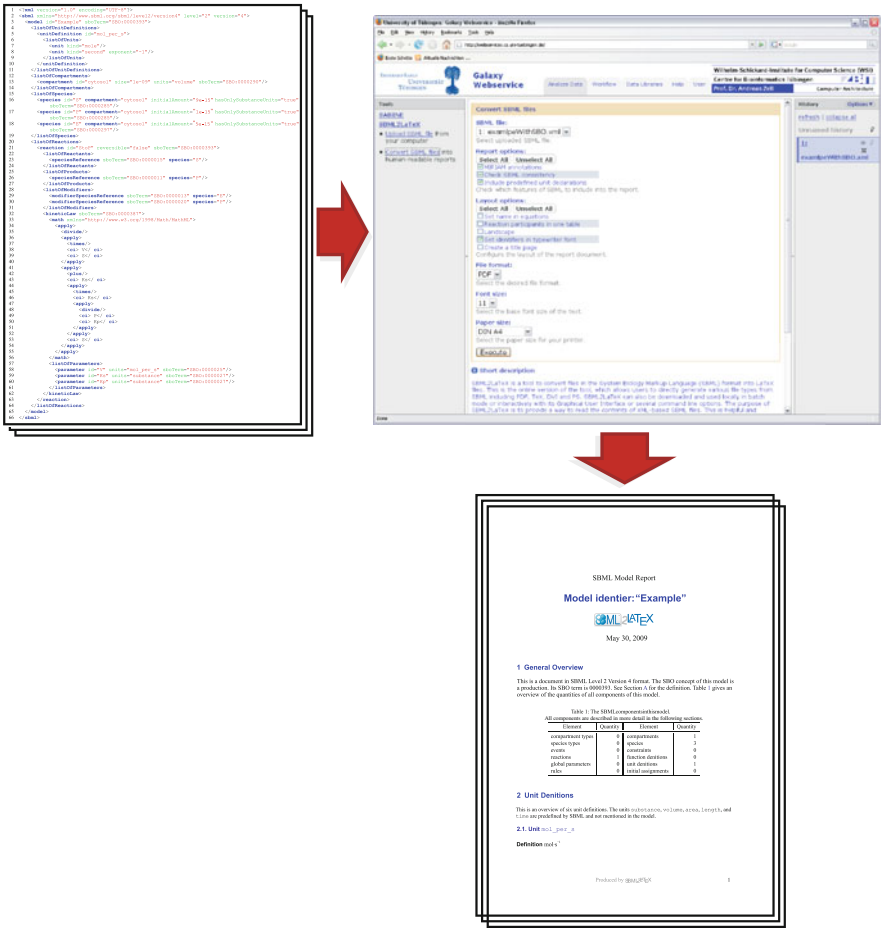


Fig. 7.11 The online version of SBML2LaTeX, adapted from (Dräger et al. 2009b). This convenient online version allows direct creation of SBML model reports from a given SBML file without local installation of any software. Several output formats are available; the font size and style, paper size, and page orientation can be changed. Landscape pages are very useful because it is often not possible to introduce a line break into long denominators and the resulting formula therefore does not fit on portrait pages. The following features can be switched off: translation of MIRIAM annotations to online resources, SBML consistency check, and the inclusion of all predefined units. Other layout options enable the user to choose whether the details of all reaction participants should be listed in one table or separate tables for each group (reactants, modifiers, and products). Instead of a simple headline, a separate title page can be created

SBML2LaTeX can also be used locally as a stand-alone tool (Fig. 7.12). SBMLsqueezer includes a flattened version of SBML2LaTeX. Like the current version of SBMLsqueezer itself, its SBML2LaTeX translator depends on CellDesigner. A comprehensive model report that supports even the latest

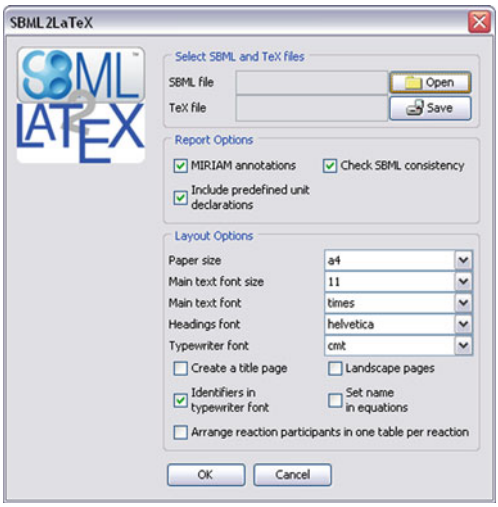


Fig. 7.12 Stand-alone version of SBML2LaTeX. In addition to the settings of the online version the GUI of the stand-alone version provides the ability to select alternative fonts for headings, main text, and typewriter passages. All settings are also available as command line options enabling the user to apply this program in a batch mode to multiple files in one go. The SBML2LaTeX project homepage http://www.ra.cs.uni-tuebingen.de/software/SBML2LaTeX provides detailed information and the latest version of this program

specification of SBML can only be obtained by using SBML2LaTeX, which is freely available at the project homepage[21] and on the SBML homepage[22].

7.7 Conclusions

In this chapter we have introduced a simple model building pipeline and discussed how this pipeline can be automated at each step. There are still several possibilities for improving this suggested procedure. In particular, what we have proposed here is to combine, in a semi-automatic way, several freely available software tools that support the modeling language called SBML (Hucka et al. 2003, 2008). The Linux live DVD SB.OS[23] (Systems Biology Operational Software) contains most of the freely available software tools described in this chapter as well as several additional programs useful for research in systems biology. SB.OS is based on the popular

[21] http://www.ra.cs.uni-tuebingen.de/software/SBML2LaTeX

[22] http://sbml.org

[23] SB.OS can be freely downloaded at www.sbos.eu for all non-commercial purposes.

Linux distribution Ubuntu and runs directly from a bootable DVD and hence does not require any installation of software on the local computer.

In the first step of the pipeline, we set up a topology in CellDesigner (Funahashi et al. 2003, 2007a) of the network we want to create a dynamic model for. To this end, we can use tools like KEGG2SBML and the information from databases like KEGG or MetaCyc (Caspi et al. 2008: Kanehisa et al. 2006). Saving the resulting process diagram in SBML, including CellDesigner-specific annotations, allows machine-readable distinctions between several specific types of reactions, modifiers, and species.

In the second step, the modeling tool known as SBMLsqueezer (Dräger et al. 2008) equips all reactions in these pathways with kinetic equations. Thereby, SBMLsqueezer derives each such equation from the context provided by CellDesigner's annotations. A stand-alone version of SBMLsqueezer would require SBO annotations (Le Novère et al. 2006b) instead. In any case, manual interaction is still needed because SBMLsqueezer can only suggest applicable kinetic equations. From these, the user selects, based on his or her knowledge, the most appropriate equations for the envisaged purpose. To simplify this process, SBMLsqueezer offers standard equations for each type of reaction and thus creates all equations of the network in one step. This mode reduces the number of necessary user interactions to a minimum. With the help of semanticSBML, the resulting model can be annotated with MIRIAM qualifiers and be combined with already existing models.

In the third step, the modeler estimates the parameters within the kinetic equations with respect to given measurement data. The combination of the MATLAB™-based SBToolbox2 (Schmidt and Jirstrand 2006; Schmidt 2007; Schmidt et al. 2007) with EvA2 (Kronfeld 2008; Streichert and Ulmer 2005) as its integral optimization core with appropriate settings is advisable. The modeling tool named COPASI (Hoops et al. 2006) or the parameter estimation tool called SBML-PET (Zi and Klipp 2006) provides valuable alternatives.

As soon as the dynamic model behavior fits the data well enough, the model should be validated experimentally (Chassagnole et al. 2002). To this end, the experimenters take additional samples at time points that have not been measured in their original experiment or under different conditions, e.g., varied initial concentrations of certain intermediates. We compute the model's output at these time points, or for the new conditions and then compare the results of the new in vivo and in vitro experiments. If both experiments strongly deviate from each other, we should return to an earlier step of our modeling pipeline and rethink our modeling approach or the structure of the system.

Finally, a model report should be created to document all parameter values and the model structure as a whole. In this way, the model can iteratively be refined by using the reporting tool named SBML2LaTeX, which provides an overview of all model equations and the reaction system (Dräger et al. 2009b). These reports uncover sources of potential mistakes.

This overall process should be viewed as an iterative procedure of topological modeling, rate law generation, model annotation, parameter estimation, model

validation, report generation, and again modification of the model's structure. It may also be advisable to produce model reports at earlier stages of the pipeline for the sake of a better overview of the model.

Two ways exist to move from the semi-automatic method to a fully automated modeling process: First, command line scripts can be written that combine the tools introduced in this chapter into a pipeline. To this end, shells such as Bash provide the special pipe command that sends the output of the first tool to the input stream of the second one. With the SBO-based stand-alone version of SBMLsqueezer, such a procedure becomes possible as long as the underlying SBML file contains SBO annotations at least for modifiers and species. The script must contain all command line settings of each program in the pipeline. The availability of source codes opens up a second possibility to obtain an automated modeling procedure. With some programing effort, another application can be created that directly combines all programs.

Despite these details, we acknowledge that several questions were not discussed within this chapter. Here we give a brief overview of other important issues regarding modeling that requires additional effort and that should be incorporated into our modeling pipeline to be complete. The mathematical space of possible models that are able to reproduce given quantitative time series data exceeds the space of biologically plausible, realistic, and meaningful models by several orders of magnitude. Therefore, physical model properties should also be taken into account. As pointed out in Section 7.3.2, the convenience rate law is designed in a thermodynamically independent form in which the values of all parameters can be estimated independently. In other cases, care must be taken to obtain thermodynamically valid parameters. In their study, Magnus et al. suggest a method to incorporate thermodynamics in a calibration procedure (Magnus et al. 2006). The time scales of very fast and very slow processes with respect to the time frame of interest constitute another important topic. By setting these significantly slower or faster reactions to constant rates (i.e., a zeroth order mass action rate), the system can be greatly simplified. Since biological systems are known to be robust (Kitano 2002b), a stability analysis and also a (global) sensitivity analysis of the system that strongly depends on the estimated parameters constitute a valuable improvement of our modeling pipeline. Tools like SBML-SAT (SBML Sensitivity Analysis Tool) (Zi et al. 2008) or SBToolbox2 (Schmidt and Jirstrand 2006) provide easily usable methods to perform such analyses. Furthermore, the existence or nonexistence of one or multiple steady-states or bifurcation analyses gives meaningful insights into the model's properties and can also be performed using existing tools like SBToolbox2.

Acknowledgments The authors are grateful to Michael J. Ziller, Marcel Kronfeld, Catherine Lloyd, Falko Krause, and Wolfram Liebermeister for helpful advice, discussion, and contribution. This work was funded by the German Federal Ministry of Education and Research (BMBF) in the two projects, *National Genome Research Network* (NGFN-II EP under grant number 0313323, later NGFN-Plus under grant number 01GS08134) and *HepatoSys* under grant number 0313080 L, and the German Federal State of Baden-Württemberg in the two projects *Identifikation und Analyse metabolischer Netze aus experimentellen Daten* under contract number 7532.22-26-18 and *Tübinger Bioinformatik-Grid* under contract number 23-7532.24-4-18/1.

References

Albert R (2007) Network Inference, Analysis, and Modeling in Systems Biology. Plant Cell 19(11):3327–3338. doi:10.1105/tpc.107.054700. http://www.plantcell.org/cgi/reprint/19/11/3327.pdf

Alves R, Antunes F, Salvador A (2006) Tools for kinetic modeling of biochemical networks. Nat Biotechnol 24(6):667–672. doi:10.1038/nbt0606-667. http://dx.doi.org/10.1038/nbt0606-667

Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G (2000) Gene ontology: tool for the unification of biology. Nat. Genet 25(1):25–29. doi:10.1038/75556. http://dx.doi.org/10.1038/75556

Balsa-Canto E, Peifer M, Banga JR, Timmer J, Fleck C (2008) Hybrid optimization method with general switching strategy for parameter estimation. BMC Syst Biol 2(1):26. doi:10.1186/1752-0509-2-26. http://dx.doi.org/10.1186/1752-0509-2-26

Banga JR (2008) Optimization in computational systems biology. BMC Systems Biol 2:47. doi:10.1186/1752-0509-2-47. http://www.biomedcentral.com/1752-0509/2/47/

Barthelmes J, Ebeling C, Chang A, Schomburg I, Schomburg D (2007) BRENDA, AMENDA and FRENDA: the enzyme information system in 2007. Nucl Acids Res 35(Suppl_1):D511–514. doi:10.1093/nar/gkl972. http://nar.oxfordjournals.org/cgi/content/abstract/35/suppl_1/D511, http://nar.oxfordjournals.org/cgi/reprint/35/suppl_1/D511.pdf

Bisswanger H (2000) Enzymkinetik – Theorie und Methoden, 3rd edn. Wiley-VCH, Weinheim

Borger S, Liebermeister W, Uhlendorf J, Klipp E (2007a) Automatically generated model of a metabolic network. Int Conf Genome Inform 18:215–224. doi:10.1142/9781860949920_0021. http://eproceedings.worldscinet.com/9781860949920/9781860949920_0021.html

Borger S, Uhlendorf J, Helbig A, Liebermeister W (2007b) Integration of enzyme kinetic data from various sources. Silico Biol 7(2 Suppl):S73–S79. http://www.bioinfo.de/isb/2007/07/S1/09/

Bornstein BJ, Keating SM, Jouraku A, Hucka M (2008) LibSBML: an API Library for SBML. Bioinformatics 24(6):880–881, doi:10.1093/bioinformatics/btn051, http://bioinformatics.oxfordjournals.org/cgi/content/abstract/24/6/880, http://bioinformatics.oxfordjournals.org/cgi/reprint/24/6/880.pdf

Bulik S, Grimbs S, Huthmacher C, Selbig J, Holzhütter HG (2009) Kinetic hybrid models composed of mechanistic and simplified enzymatic rate laws-a promising method for speeding up the kinetic modelling of complex metabolic networks. FEBS J 276(2):410–424. doi:10.1111/j.1742-4658.2008.06784.x, http://www3.interscience.wiley.com/journal/121588609/abstract

Burnham KP, Anderson DR (2002) Model selection and multimodel inference: a practical information-theoretic approach, 2nd edn. Springer, New York, NY. http://www.springer.com/statistics/statistical+theory+and+methods/book/978-0-387-95364-9?cm

Calzone L, Gelay A, Zinovyev A, Radvanyi F, Barillot E (2008) A comprehensive modular map of molecular interactions in rb/e2f pathway. Mol Syst Biol 4:173, doi:10.1038/msb.2008.7. http://dx.doi.org/10.1038/msb.2008.7

Caspi R, Foerster H, Fulcher CA, Kaipa P, Krummenacker M, Latendresse M, Paley S, Rhee SY, Shearer AG, Tissier C, Walk TC, Zhang P, Karp PD (2008) The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. Nucleic Acids Res 36(Database issue):D623–D631. doi: 10.1093/nar/gkm900. http://nar.oxfordjournals.org/cgi/content/abstract/36/suppl_1/D623

Chang A, Scheer M, Grote A, Schomburg I, Schomburg D (2009) BRENDA, AMENDA and FRENDA the enzyme information system: new content and tools in 2009. Nucleic Acids Res 37(Database issue):D588–D592. doi:10.1093/nar/gkn820. http://nar.oxfordjournals.org/cgi/content/full/gkn820, http://nar.oxfordjournals.org/cgi/reprint/37/suppl_1/D588.pdf

Chassagnole C, Noisommit-Rizzi N, Schmid JW, Mauch K, Reuss M (2002) Dynamic modeling of the central carbon metabolism of *Escherichia coli*. Biotechnol Bioengineer 79(1):54–73. doi:10.1002/bit.10288. http://www3.interscience.wiley.com/journal/93519745/abstract

Cherry JM, Adler C, Ball C, Chervitz SA, Dwight SS, Hester ET, Jia Y, Juvik G, Roe T, Schroeder M, Weng S, Botstein D (1998) SGD: *Saccharomyces* Genome Database. Nucleic Acids Res 26(1):73–79. doi:10.1093/nar/26.1.73. http://nar.oxfordjournals.org/cgi/content/abstract/26/1/73

Clerc M (2005) Particle swarm optimization. ISTE Ltd, London

Clerc M, Kennedy J (2002) The particle swarm–explosion, stability, and convergence in a multidimensional complex space. IEEE Trans on Evol Comput 6(1):58–73

Cornish-Bowden A (2004) Fundamentals of enzyme kinetics, 3rd edn. Portland Press Ltd., 59 Portland Place, London

Dampier W, Tozeren A (2007) Signaling perturbations induced by invading *H. pylori* proteins in the host epithelial cells: a mathematical modeling approach. J Theor Biol 248(1):130–144. doi:10.1016/j.jtbi.2007.03.014. http://dx.doi.org/10.1016/j.jtbi.2007.03.014

Deckard A, Bergmann FT, Sauro HM (2006) Supporting the SBML layout extension. Bioinformatics 22(23):2966–2967. http://bioinformatics.oxfordjournals.org/cgi/content/abstract/22/23/2966, http://bioinformatics.oxfordjournals.org/cgi/reprint/22/23/2966.pdf

Degtyarenko K, Matos Pd, Ennis M, Hastings J, Zbinden M, McNaught A, Alcántara R, Darsow M, Guedj M, Ashburner M (2008) ChEBI: a database and ontology for chemical entities of biological interest. Nucleic Acids Res 36(Database issue):D344–D350. doi:10.1093/nar/gkm791. http://nar.oxfordjournals.org/cgi/content/full/gkm791v1

Dräger A, Kronfeld M, Supper J, Planatscher H, Magnus JB, Oldiges M, Zell A (2007a) Benchmarking evolutionary algorithms on convenience kinetics models of the Valine and Leucine Biosynthesis in *C. glutamicum*. In: Srinivasan D, Wang L (eds) 2007 IEEE congress on evolutionary computation, IEEE computational intelligence society. IEEE Press, Singapore, pp 896–903

Dräger A, Supper J, Planatscher H, Magnus JB, Oldiges M, Zell A (2007b) Comparing various evolutionary algorithms on the parameter optimization of the valine and leucine biosynthesis in *Corynebacterium glutamicum*. In: Srinivasan D, Wang L (eds) 2007 IEEE congress on evolutionary computation, IEEE computational intelligence society, IEEE Press, Singapore, pp 620–627

Dräger A, Hassis N, Supper J, Schröder A, Zell A (2008) SBMLsqueezer: a CellDesigner plug-in to generate kinetic rate equations for biochemical networks. BMC Syst Biol 2(1):39. doi:10.1186/1752-0509-2-39. http://dx.doi.org/10.1186/1752-0509-2-39

Dräger A, Kronfeld M, Ziller MJ, Supper J, Planatscher H, Magnus JB, Oldiges M, Kohlbacher O, Zell A (2009a) Modeling metabolic networks in *C. glutamicum*: a comparison of rate laws in combination with various parameter optimization strategies. BMC Syst Biol 3:5. doi:10.1186/1752-0509-3-5. http://www.biomedcentral.com/1752-0509/3/5

Dräger A, Planatscher H, Wouamba DM, Schröder A, Hucka M, Endler L, Golebiewski M, Müller W, Zell A (2009b) SBML2LaTeX: Conversion of SBML files into human-readable reports. Bioinformatics doi:10.1093/bioinformatics/btp170. http://bioinformatics.oxfordjournals.org/cgi/content/abstract/btp170v1, http://bioinformatics.oxfordjournals.org/cgi/reprint/btp170v1.pdf

Finney A, Hucka M (2003) Systems biology markup language: Level 2 and beyond. Biochem Soc Trans 31(Pt 6):1472–1473. doi:10.1042/. http://www.biochemsoctrans.org/bst/031/1472/bst0311472.htm

Fujibuchi W, Goto S, Migimatsu H, Uchiyama I, Ogiwara A, Akiyama Y, Kanehisa M (1998) DBGET/LinkDB: an integrated database retrieval system. Pac Symp Biocomput pp 683–694

Funahashi A, Tanimura N, Morohashi M, Kitano H (2003) CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. BioSilico 1(5): 159–162. doi:10.1016/S1478-5382(03)02370-9. http://www.sciencedirect.com/science/article/B75GS-4BS08JD-5/2/5531c80ca62a425f55d224b8a0d3f702

Funahashi A, Jouraku A, Matsuoka Y, Kitano H (2007a) Integration of CellDesigner and SABIO-RK. Silico Biology 7(2 Suppl):S81–S90. http://www.bioinfo.de/isb/200707S110/main.html

Funahashi A, Morohashi M, Matsuoka Y, Jouraku A, Kitano H (2007b) Cell Designer: a graphical biological network editor and workbench interfacing simulator. In: Choi S (ed) Introduction

to systems biology, Humana Press, chap 21, pp 422–434. doi:10.1007/978-1-59745-531-2_21. http://www.springerlink.com/content/hqk374162wg70146/

Gauges R, Rost U, Sahle S, Wegner K (2006) A model diagram layout extension for SBML. Bioinformatics 22(15):1879–1885. doi:10.1093/bioinformatics/btl195. http://bioinformatics.oxfordjournals.org/cgi/content/abstract/22/15/1879, http://bioinformatics.oxfordjournals.org/cgi/reprint/22/15/1879.pdf

Gillespie DT (2000) The chemical Langevin equation. J. Chem Phys 113:297–306. doi:10.1063/1.481811. http://link.aip.org/link/?JCPSA6/113/297/1

Gruber TR (1993) Toward principles for the design of ontologies used for knowledge sharing? Int J Hum Compu Stud 43(5–6):907–928, http://dx.doi.org/10.1006/ijhc.1995.1081

Hatzimanikatis V, Bailey J (1996) MCA has more to say. theor Biolo 182(3):233–342. doi:10.1006/jtbi.1996.0160. http://dx.doi.org/10.1006/jtbi.1996.0160

Hatzimanikatis V, Floudas CA, Bailey JE (1996) Analysis and design of metabolic reaction networks via mixed-integer linear optimization. AIChE 42(5):1277–1292. doi:10.1002/aic.690420509

Heinrich R, Schuster S (1996) The regulation of cellular systems. Chapman and Hall, New York, NY

Hinze T, Hayat S, Lenser T, Matsumaru N, Dittrich P (2007) Hill Kinetics meets P systems: a case study on gene regulatory networks as computing agents *in silico* and *in vivo*. In: Eleftherakis G, Kefalas P, Paun G (eds) Proceedings of the Eight Workshop on Membrane Computing, SEERC, pp 363–381

Hoffmann R, Valencia A (2004) A gene network for navigating the literature. Nature Genetetics 36(7):664. doi:10.1038/ng0704-664. http://www.nature.com/ng/journal/v36/n7/full/ng0704-664.html

Holland JH (1975) Adaptation in natural and artificial systems. The University of Michigan Press, Cambridge, MA

Hoops S, Sahle S, Gauges R, Lee C, Pahle J, Simus N, Singhal M, Xu L, Mendes P, Kummer U (2006) COPASI–a COmplex PAthway SImulator. Bioinformatics 22(24):3067–3074. doi:10.1093/bioinformatics/btl485. http://bioinformatics.oxfordjournals.org/cgi/content/abstract/22/24/3067, http://bioinformatics.oxfordjournals.org/cgi/reprint/22/24/3067.pdf

Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H (2002) The erato systems biology workbench: enabling interaction and exchange between software tools for computational biology. Pac Symp Biocomput 450–461

Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr JHS, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Le Novère N, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED, Nakayama Y, Nelson MR, Nielsen PF, Sakurada T, Schaff JC, Shapiro BE, Shimizu TS, Spence HD, Stelling J, Takahashi K, Tomita M, Wagner JM, Wang J, the rest of the SBML Forum (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19(4):524–531. doi:10.1093/bioinformatics/btg015. http://bioinformatics.oxfordjournals.org/cgi/content/abstract/19/4/524, http://bioinformatics.oxfordjournals.org/cgi/reprint/19/4/524.pdf

Hucka M, Finney A, Bornstein BJ, Keating SM, Shapiro BE, Matthews J, Kovitz BL, Schilstra MJ, Funahashi A, Doyle JC, Kitano H (2004) Evolving a lingua franca and associated software infrastructure for computational systems biology: the systems biology markup language (SBML) project. Syst Biol IEE 1(1):41–53. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1334988

Hucka M, Finney A, Hoops S, Keating SM, Le Novère N (2008) Systems biology markup language (SBML) Level 2: structures and facilities for model definitions. Tech. rep., Nat Proce doi:10.1038/npre.2008.2715.1. http://dx.doi.org/10.1038/npre.2008.2715.1

Kanehisa M, Goto S, Hattori M, Aoki-Kinoshita KF, Itoh M, Kawashima S, Katayama T, Araki M, Hirakawa M (2006) From genomics to chemical genomics: new

developments in KEGG. Nucl Acids Res 34(1):D354–357. doi:10.1093/nar/gkj102. http://nar.oxfordjournals.org/cgi/content/full/34/suppl_1/D354

King EL, Altman C (1956) A schematic method of deriving the rate laws for enzyme-catalyzed reactions. J Phys Chem 60(10):1375–1378. doi:10.1021/j150544a010. http://pubs.acs.org/doi/abs/10.1021/j150544a010

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680. http://www.sciencemag.org/cgi/content/abstract/220/4598/671

Kitano H (2002a) Computational systems biology. Nature 420(6912):206–210. http://dx.doi.org/10.1038/nature01254

Kitano H (2002b) Systems biology: a brief overview. Science 295(5560):1662–1664. http://www.sciencemag.org/cgi/content/abstract/295/5560/1662

Kitano H, Funahashi A, Matsuoka Y, Oda K (2005) Using process diagrams for the graphical representation of biological networks. Nat Biotechnol 23(8):961–966. http://dx.doi.org/10.1038/nbt1111

Klipp E, Liebermeister W, Helbig A, Kowald A, Schaber J (2007) Systems biology standards–the community speaks. Nat Biotechnol 25(4):390–391

Krebs O, Golebiewski M, Kania R, Mir S, Saric J, Weidemann A, Wittig U, Rojas I (2007) SABIO-RK: A data warehouse for biochemical reactions and their kinetics. J Integra Bioinform 4(1). doi:10.2390/biecoll-jib-2007-49. http://journal.imbio.de/index.php?paper_id=49

Kronfeld M (2008) EvA2 Short documentation. University of Tübingen, Deptartment of Computer Architecture, Tübingen, Germany, http://www.ra.cs.uni-tuebingen.de/software/EvA2

Laible C, Le Novère N (2007) MIRIAM Resources: tools to generate and resolve robust cross-references in Systems Biology. BMC Syst Biol 13(58):58–67. doi:10.1186/1752-0509-1-58. http://www.biomedcentral.com/1752-0509/1/58

Le Novère N (2006) Model storage, exchange and integration. BMC Neurosci 7 (Suppl 1):S11. doi:10.1186/1471-2202-7-S1-S11. http://dx.doi.org/10.1186/1471-2202-7-S1-S11

Le Novère N, Finney A, Hucka M, Bhalla US, Campagne F, Collado-Vides J, Crampin EJ, Halstead M, Klipp E, Mendes P, Nielsen P, Sauro H, Shapiro BE, Snoep JL, Spence HD, Wanner BL (2005) Minimum information requested in the annotation of biochemical models (MIRIAM). Nat. Biotechnol 23(12):1509–1515. doi:10.1038/nbt1156. http://www.nature.com/nbt/journal/v23/n12/abs/nbt1156.html

Le Novère N, Bornstein BJ, Broicher A, Courtot M, Donizelli M, Dharuri H, Li L, Sauro H, Schilstra M, Shapiro B, Snoep JL, Hucka M (2006a) BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. Nucleic Acids Res 34:D689–D691. doi:10.1093/nar/gkj092. http://nar.oxfordjournals.org/cgi/content/full/34/suppl_1/D689

Le Novère N, Courtot M, Laibe C (2006b) Adding semantics in kinetics models of biochemical pathways. In: Kettner C, Hicks MG (eds) 2nd International ESCEC workshop on experimental standard conditions on enzyme characterizations. Beilstein Institut, Rüdesheim, Germany, ESEC, Rüdessheim/Rhein, Germany, pp 137–153. http://www.beilstein-institut.de/escec2006/proceedings/LeNovere/LeNovere.pdf

Le Novère N, Moodie S, Sorokin A, Hucka M, Schreiber F, Demir E, Mi H, Matsuoka Y, Wegner K, Kitano H (2008) Systems biology graphical notation: process diagram level 1. Tech. Rep., Nat Proced. doi:hdl:10101/npre.2008.2320.1. http://hdl.handle.net/10101/npre.2008.2320.1

Liebermeister W (2008) Validity and combination of biochemical models. In: Kettner C, Hicks MG (eds) Proceedings of 3rd International ESCEC Workshop on experimental standard conditions on enzyme characterizations, ESEC, Rüdessheim/Rhein, pp 163–179. http://www.molgen.mpg.de/~lieberme/data/Liebermeister_Merging_Validity_2008.pdf

Liebermeister W, Klipp E (2005) Biochemical networks with uncertain parameters. IEE Proce Syst Biol 152(3):97–107, doi:10.1049/ip-syb:20045033. http://link.aip.org/link/?BDJ/152/97/1

Liebermeister W, Klipp E (2006) Bringing metabolic networks to life: convenience rate law and thermodynamic constraints. Theor Biol Med Model 3(42):41. doi:10.1186/1742-4682-3-41. http://dx.doi.org/10.1186/1742-4682-3-41

Liebermeister W, Krause F, Klipp E (2008) Merging of systems biology models with semanticSBML. Tech. Rep., Max Planck Institute for Molecular Genetics, Berlin. http://www.molgen.mpg.de~lieberme/data/semanticSBML_heidelberg_2008.pdf

Liebermeister W, Krause F, Uhlendorf J, Lubitz T, Klipp E (2009) SemanticSBML: a tool for annotating, checking, and merging of biochemical models in SBML format. In: 3rd International biocuration conference, Nature Publishing Group. doi:10.1038/npre.2009.3093.1. http://dx.doi.org/10.1038/npre.2009.3093.1

Lister AL, Pocock M, Taschuk M, Wipat A (2009) Saint: a lightweight integration environment for model annotation. Bioinformatics p btp523. doi:10.1093/bioinformatics/btp523. http://bioinformatics.oxfordjournals.org/cgi/content/abstract/btp523v2 http://bioinformatics.oxfordjournals.org/cgi/reprint/btp523v2.pdf

Lloyd CM, Halstead MDB, Nielsen PF (2004) CellML: its future, present and past. Prog Biophys Mol Biol 85(2-3):433–450. doi:10.1016/j.pbiomolbio.2004.01.004 http://dx.doi.org/10.1016/j.pbiomolbio.2004.01.004

Machné R, Finney A, Müller S, Lu J, Widder S, Flamm C (2006) The SBML ODE Solver Library: a native API for symbolic and fast numerical analysis of reaction networks. Bioinformatics 22(11):1406–1407. doi:10.1093/bioinformatics/btl086. http://dx.doi.org/10.1093/bioinformatics/btl086 http://bioinformatics.oxfordjournals.org/cgi/reprint/22/11/1406.pdf

Magnus JB, Hollwedel D, Oldiges M, Takors R (2006) Monitoring and modeling of the reaction dynamics in the valine/leucine synthesis pathway in *Corynebacterium glutamicum*. Biotechnol Prog 22(4):1071–1083. http://dx.doi.org/10.1021/bp060072f

Moles CG, Mendes P, Banga JR (2003) Parameter estimation in biochemical pathways: a comparison of global optimization methods. Genome Res 13(11):2467–2474. doi:10.1101/gr.1262503. http://www.genome.org/cgi/doi/10.1101/gr.1262503

Nickerson DP, Buist ML (2009) A physiome standards-based model publication paradigm. Phil Trans R Soc A 367:1823–1844. doi:10.1098/rsta.2008.0296

Oda K, Matsuoka Y, Funahashi A, Kitano H (2005) A comprehensive pathway map of epidermal growth factor receptor signaling. Mol Syst Biol 1:2005.0010. doi:10.1038/msb4100014. http://dx.doi.org/10.1038/msb4100014

Ogata H, Goto S, Sato K, Fujibuchi W, Bono H, Kanehisa M (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res 27(1):29–34

Olivier BG, Snoep JL (2004) Web-based kinetic modelling using JWS Online. Bioinformatics 20 (13):2143–2144. doi:10.1093/bioinformatics/bth200. http://dx.doi.org/10.1093/bioinformatics/bth200

Rechenberg I (1973) Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution. Fromman-Holzboog, Stuttgart

Rodriguez N, Donizelli M, Le Novère N (2007) SBMLeditor: effective creation of models in the systems biology markup language (SBML). BMC Bioinform 8:79. doi:10.1186/1471-2105-8-79. http://dx.doi.org/10.1186/1471-2105-8-79

Rodriguez-Fernandez M, Egea JA, Banga JR (2006a) Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems. BMC Bioinform 7:483. doi:10.1186/1471-2105-7-483. http://dx.doi.org/10.1186/1471-2105-7-483

Rodriguez-Fernandez MR, Mendes P, Banga JR (2006b) A hybrid approach for efficient and robust parameter estimation in biochemical pathways. Biosystems 83:248–265. doi:10.1016/j.biosystems.2005.06.016. http://www.sciencedirect.com/science/article/B6T2K-4HC776X-4/2/2a48c31a0d9aa413bc616023689e55c8

Savageau MA (1969a) Biochemical systems analysis. I. Some mathematical properties of the rate law for the component enzymatic reactions. J Theor Biol 25(3):365–369

Savageau MA (1969b) Biochemical systems analysis. II. The steady-state solutions for an n-pool system using a power-law approximation. J Theor Biol 25(3):370–379

Savageau MA (1970) Biochemical systems analysis. 3. Dynamic solutions using a power-law approximation. J Theor Biol 26(2):215–226

Schauer M, Heinrich R (1983) Quasi-steady-state approximation in the mathematical modeling of biochemical reaction networks. Math Biosci 65:155–171

Schilstra MJ, Li L, Matthews J, Finney A, Hucka M, Le Novère N (2006) CellML2SBML: conversion of CellML into SBML. Bioinformatics 22(8):1018–1020. doi:10.1093/bioinformatics/btl047. http://bioinformatics.oxfordjournals.org/cgi/content/abstract/22/8/1018, http://bioinformatics.oxfordjournals.org/cgi/reprint/22/8/1018.pdf

Schmidt H (2007) SBaddon: high performance simulation for the systems biology toolbox for MATLAB. Bioinformatics 23(5):646–647. doi:10.1093/bioinformatics/btl668. http://bioinformatics.oxfordjournals.org/cgi/content/abstract/23/5/646, http://bioinformatics.oxfordjournals.org/cgi/reprint/23/5/646.pdf

Schmidt H, Jirstrand M (2006) Systems biology toolbox for MATLAB: a computational platform for research in systems biology. Bioinformatics 22(4):514–515. doi:10.1093/bioinformatics/bti799. http://dx.doi.org/10.1093/bioinformatics/bti799

Schmidt H, Drews G, Vera J, Wolkenhauer O (2007) SBML export interface for the systems biology toolbox for MATLAB. Bioinformatics 23(10):1297–1298. doi:10.1093/bioinformatics/btm105. http://dx.doi.org/10.1093/bioinformatics/btm105. http://bioinformatics.oxfordjournals.org/cgi/reprint/23/10/1297.pdf

Schomburg I, Chang A, Schomburg D (2002) BRENDA, enzyme data and metabolic information. Nucl Acids Res 30(1):47–49. doi:10.1093/nar/30.1.47. http://nar.oxfordjournals.org/cgi/content/abstract/30/1/47

Segel IH (1993) Enzyme Kinetics–Behavior and Analysis of Rapid Equilibrium and Steady-State Enzyme Systems. Wiley-Interscienne, New York, NY

Shapiro BE, Finney A, Hucka M, Bornstein BJ, Funahashi A, Jouraku A, Keating SM, Le Novère N, Matthews J, Schilstra MJ (2007) Introduction to systems biology. Humana Press, Totowa, NJ chap SBML Models and MathSBML, pp 395–421. doi:10.1007/978-1-59745-531-2. http://www.springerlink.com/content/q28j426582387022/

Snoep JL, Bruggeman F, Olivier BG, Westerhoff HV (2006) Towards building the silicon cell: a modular approach. Biosystems 83(2-3):207–216. doi:10.1016/j.biosystems.2005.07.006. http://dx.doi.org/10.1016/j.biosystems.2005.07.006

Spieth C, Streichert F, Speer N, Zell A (2004) Optimizing topology and parameters of gene regulatory network models from time-series experiments. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004), LNCS, vol 3102 (Part I), pp 461–470

Spieth C, Streichert F, Speer N, Zell A (2005a) Inferring regulatory systems with noisy pathway information. In: German conference on bioinformatics (GCB 2005), vol P-71, pp 193–203

Spieth C, Streichert F, Supper J, Speer N, Zell A (2005b) Feedback memetic algorithms for modeling gene regulatory networks. In: Proceedings of the IEEE symposium on computational intelligence in bioinformatics and computational biology (CIBCB 2005), pp 61–67

Spieth C, Supper J, Streichert F, Speer N, Zell A (2006a) JCell–a Java-based framework for inferring regulatory networks from time series data. Bioinformatics 22(16):2051–2052. doi:10.1093/bioinformatics/btl322. http://bioinformatics.oxfordjournals.org/cgi/content/abstract/22/16/2051, http://bioinformatics.oxfordjournals.org/cgi/reprint/22/16/2051.pdf

Spieth C, Worzischek R, Streichert F, Supper J, Speer N, Zell A (2006b) Comparing evolutionary algorithms on the problem of network inference. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006)

Storn R (1996) On the usage of differential evolution for function optimization. In: 1996 Biennial Conference of the North American Fuzzy Information Processing Society, IEEE, New York, Berkeley, pp 519–523

Streichert F, Ulmer H (2005) JavaEvA–A Java framework for evolutionary algorithms. Technical Report WSI-2005-06, Center for Bioinformatics Tübingen, University of Tübingen, Tübingen, Germany. doi:urn:nbn:de:bsz:21-opus-17022. http://w210.ub.uni-tuebingen.de/dbt/volltexte/2005/1702/

Teusink B, Passarge J, Reijenga CA, Esgalhado E, van der Weijden CC, Schepper M, Walsh MC, Bakker BM, van Dam K, Westerhoff HV, Snoep JL (2000) Can yeast

glycolysis be understood in terms of in vitro kinetics of the constituent enzymes? Testing biochemistry. Eur J Biochem 267(17):5313–5329. doi:10.1046/j.1432-1327.2000.01527.x. http://www3.interscience.wiley.com/journal/119181440/abstract

Tovey CA (1985) Hill climbing with multiple local optima. Alg Disc Meth 6(3):384–393. doi:10.1137/0606040. http://link.aip.org/link/?SML/6/384/1

Ulmer H (2005) Modellunterstützte evolutionäre optimierungsverfahren in javaeva. PhD thesis, Eberhard-Karls-Universität Tübingen

Visser D, Heijnen JJ (2002) The mathematics of metabolic control analysis revisited. Metab Eng 4:114–123. doi:10.1006/mben.2001.0216. http://www.sciencedirect.com/science/article/B6WN3-45V802C-3/2/d624a20d0e70ca2a1058359d7fd00cb0

Visser D, Heijnen JJ (2003) Dynamic simulation and metabolic re-design of a branched pathway using linlog kinetics. Metab Eng 5(3):164–176

Wilkinson DJ (2006) Stochastic modelling for systems biology. CRC Press, Boca Raton, FL

Wittig U, Golebiewski M, Kania R, Krebs O, Mir S, Weidemann A, Anstein S, Saric J, Rojas I (2006) SABIO-RK: Integration and curation of reaction kinetics data. In: Leser U, Naumann F, Eckmann B (eds) Data integration in the life sciences, Springer, Berlin pp 94–103. doi:10.1007/11799511. http://www.springerlink.com/content/kw1kv13614272400

Zi Z, Klipp E (2006) SBML-PET: a systems biology markup language-based parameter estimation tool. Bioinformatics 22(21):2704–2705. doi:10.1093/bioinformatics/btl443. http://bioinformatics.oxfordjournals.org/cgi/content/abstract/22/21/2704, http://bioinformatics.oxfordjournals.org/cgi/reprint/22/21/2704.pdf

Zi Z, Zheng Y, Rundell AE, Klipp E (2008) SBML-SAT: a systems biology markup language (SBML) based sensitivity analysis tool. BMC Bioinforma 9(1):342. doi:10.1186/1471-2105-9-342. http://www.biomedcentral.com/1471-2105/9/342

Ziller MJ (2009) Automatisierte mathematische Modellierung biochemischer Reaktionsnetzwerke. Master's thesis, Eberhard-Karls-Universität Tübingen, Center for Bioinformatics Tübingen, Sand 1, 72076 Tübingen