Kinetic Modelling with the Systems Biology Modelling Environment SyBME

Björn H. Junker, Dirk Koschützki* and Falk Schreiber

Leibniz Institute of Plant Genetics and Crop Plant Research,
Corrensstr. 3, 06466 Gatersleben, Germany
koschuet@ipk-gatersleben.de

Summary

Kinetic modelling and simulation is an important approach in systems biology. While the focus of current modelling tools is on simulation, model development is a highly iterative process which is currently only partly supported. To support the development of biochemical models, their simulation, and graphical understanding, we designed and implemented SyBME, the Systems Biology Modelling Environment. Here we present the architecture and the main components of SyBME and show its use by modelling sucrose breakdown in developing potato tubers.

1 Introduction

Biological research is currently observing a shift from bottom-up to top-down approaches, in which the detailed investigation of single components is giving way to the observation of whole systems. Aimed at in-depth understanding of biological processes, kinetic modelling and simulation is likely to gain more and more importance in systems biology. The advantage over purely structural analysis of biological networks is that through the high detail of kinetic models it is possible to derive quantitative statements about the investigated system, a feature that is desirable for example in metabolic engineering.

A variety of tools for kinetic modelling and simulation exist and these can be divided into three categories. The first group consists of generic mathematical modelling tools that makes use of universal mathematical software such as MATLAB, for example SimTool [1]. In the second group are command-line driven simulators such as Jarnac [17] or PySCeS [13]. The software that is probably most accepted among biologists constitutes the third group: simulation tools with graphical user interfaces such as Gepasi [12] or E-Cell [18]. All these tools have advantages and disadvantages. Usually they are either designed to be intuitively understandable or to be easily extendable, and they offer different functionalities. However, there is currently no tool available that combines different functionalities and is easily extendible as well as understandable. While the user could analyse the same model in different tools to maximise the available functionalities, the diversity of the input and output formats of the simulation tools makes this a tedious task and in most cases the output has to be evaluated manually. Furthermore, as kinetic modelling is an iterative process, modellers will end up having many different versions of one model, probably in a number of different formats. To address these problems, we have

*Corresponding author

designed the tool SyBME - a Systems Biology Modelling Environment. SyBME supports the development of models, their export to several simulation tools and an easy understanding of the model via a graphical model representation. It consists of a model repository, a graphical user interface written in Java and an interface to existing simulators.

The paper is structured as follows: in the Methods section we describe the architecture of SyBME and focus on some of the features that distinguish SyBME from other modelling or simulation environments. In the Results section, we describe how the system is being used for the analysis of a kinetic model of sucrose breakdown in the potato tuber [9] and how the output of the simulation can be visualised. At the end, we give some conclusions.

2 Methods

2.1 Implementation Environment of SyBME

SyBME is written in the Java programming language, realised as a fat client application [2] and deployed via Java Web Start which allows the easy deployment onto clients running different operating systems. The supported simulators (Gepasi and Jarnac) are currently available for the Windows operating system only. Therefore simulations can be performed either on Windows based systems directly or with the help of emulation (Wine [20]) or virtualisation (VMWare [19]) solutions.

2.2 Architecture of SyBME

To accelerate the development of SyBME we integrated some third-party products. Fig. 1 shows a UML component diagram of these and their connections to SyBME.

For the underlying graph data structure as well as the visualisation of the network of biochemical reactions the Java Universal Network/Graph Framework (JUNG) [8] is used. We use JUNG as it provides a solid graph data structure and a fast visualisation engine. Several graph layout algorithms are distributed with JUNG and have been integrated into SyBME.

The focus of our project was to develop a tool that facilitates model design and management, rather than serving simulation purposes. Therefore we decided to attach already existing simulators. Two of them were chosen as our initial candidates: Gepasi [12] and Jarnac [17]. The former has a graphical user interface (GUI) and the latter has a command line interface which allows other programs, in our case SyBME, to control the application and perform simulations.

Persistence of models is archived via the object/relational persistence system Hibernate [4]. Our internal object oriented representation of the biochemical reaction network is stored into a relational database system and all required transformations are performed automatically by Hibernate. Currently, we have experience with two relational database management systems: HSQLDB [6] and PostgreSQL [15]. A test with the present Oracle [14] version 10g is planned.

For the visualisation of time series simulation results we connected SyBME to VANTED [10].

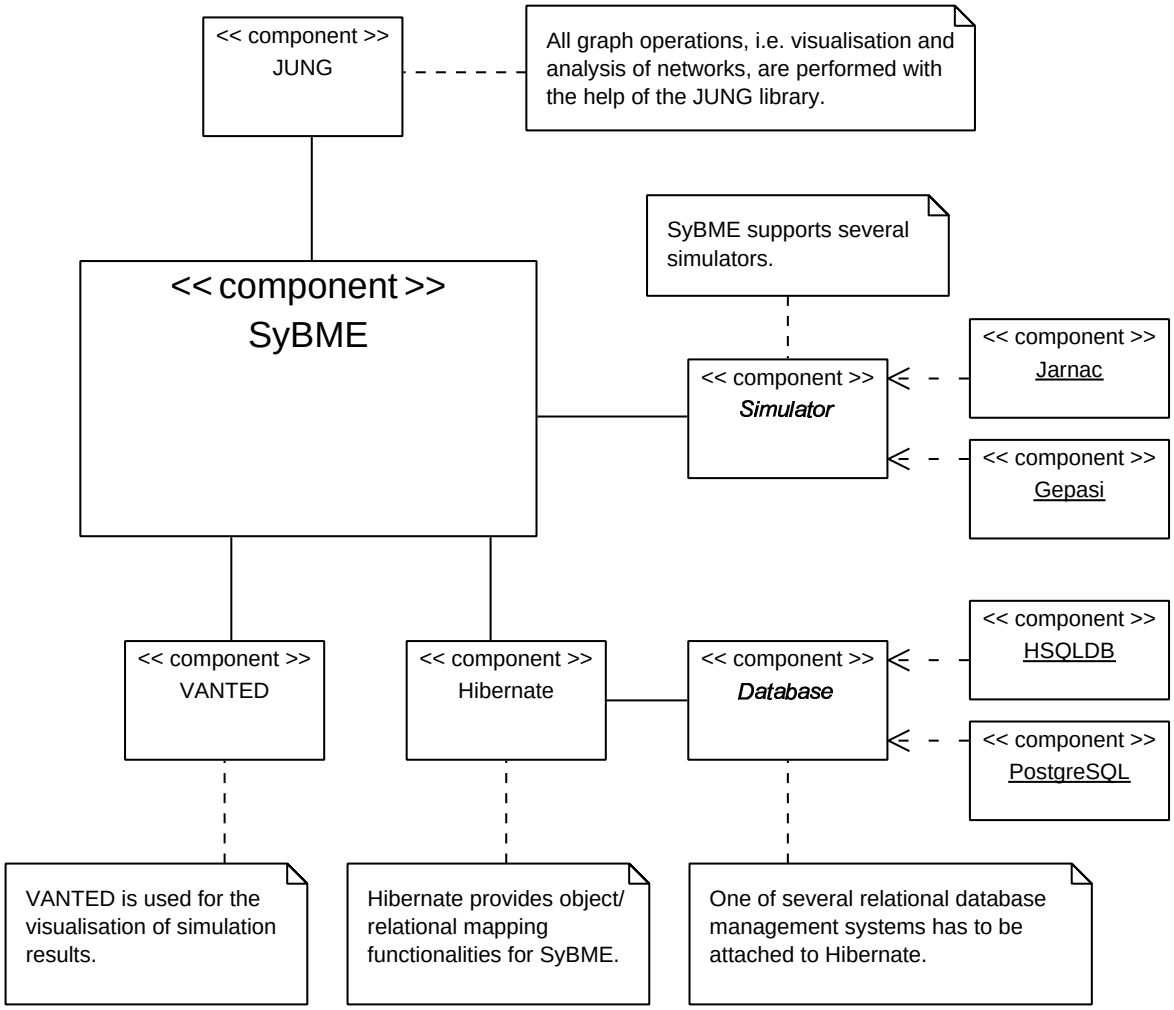

Figure 1: The architecture of SyBME shown as a UML component diagram. Components that are integrated into SyBME are attached with solid lines. Components connected with dashed arrows are required but interchangeable, e.g. only one of the database management systems is required. A dashed line connects a note to the corresponding element.

2.3 Model Structure and Model Repository

Modelling of a biochemical pathway is not a straightforward process. Usually many versions of the model are created and evaluated until a final fit is found for the evaluation criteria specified and the existing parameters. During model development usually some components of the model are fixed very early and some aspects are changed very frequently. In increasing order of changes these components are:

Stoichiometric Equations The stoichiometry of the system under investigation, e.g. `UDPglc + PP <=> G1P + UTP`, is usually fixed very early during a modelling approach.

Rate Laws Rate laws for the catalysing enzymes, e.g. Michaelis-Menten equations, can only be fixed after the stoichiometry of the reaction is known. They are very seldom changed after one has selected an equation.

Rate Law Constants and Initial Values The kinetic constants, e.g. k_m or V_{max} values, can

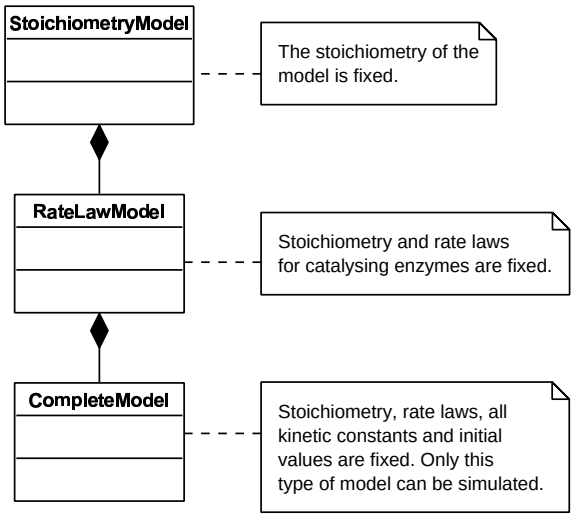

Figure 2: UML class diagram of the hierarchical structure of the model repository. Each stoichiometry model can be reused within several rate law models and each rate law model can be reused within several complete models. Only the complete models at the bottom can be simulated within Gepasi or Jarnac.

only be fixed after the rate laws are known. Kinetic constants are changed very frequently during modelling as they are the key parameter to fit a model into an existing setting. For each metabolite concentration in a model an initial value has to be declared. This value is used by the simulator to initialise the concentration of the metabolite at the start of the simulation. Conceptually they are similar to the kinetic constants but in contrast their values are not changed very frequently.

Usual modelling and simulation environments do not cover this hierarchical structure. They allow the user only to work on a single model and store this model into a single file. Changes of a model have to be traced manually and the dependency of the different layers is not visible to the user.

Within SyBME we have realised such a hierarchical structure of model layers, Fig. 2 shows this structure as a UML class diagram. The user has to generate a `StoichiometryModel` first, has to create a `RateLawModel` next, and finally complete this model into a `CompleteModel` which can be simulated, for example, in Gepasi. In contrast to other systems the different models are stored in a database and files are created only on request, i.e. if a model has to be simulated or transferred into a different system.

2.4 Graphical Representation of Enzymatic Reactions

Among biologists, metabolic networks are usually represented as directed hypergraphs (see Fig. 3a and Fig. 5). A directed hypergraph consists of a set of vertices representing metabolites and a set of hyperedges representing the reactions. Each hyperedge has one to many start vertices and one to many end vertices. The direction of a hyperedge is represented by arrows at the end vertices of the hyperedge. As most biochemical reactions are reversible, this is usually represented by arrows at both ends of a hyperedge and by placing start and end vertices of a

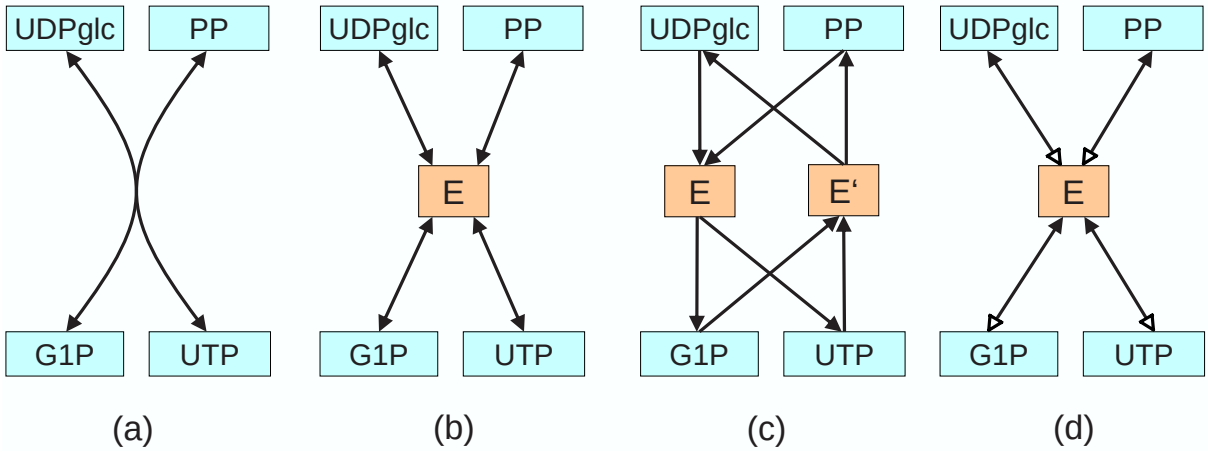

Figure 3: Different representations of an enzymatic reaction by the example of the UDPglucose pyrophosphorylase (E), which is catalysing the reversible reaction `UDPglc + PP <=> G1P + UTP`. The Different representations are: a) directed hypergraph (with reversible reaction), b) directed bipartite graph (with reversible reaction), c) one vertex for each direction of the reaction, and d) method used within SyBME. Abbreviations: see Fig. 6.

reaction on opposite sides of a hyperedge (e.g. substrates of the reaction at the top, products at the bottom, see Fig. 3a).

Directed hypergraphs are an unusual structure to model networks, because common methods for the analysis and visualisation of directed graphs are not immediately applicable to hypergraphs. Therefore a transformation of the directed hypergraph into a directed graph has to be performed. The usual representation of a hypergraph is a bipartite graph, a graph which has two sets of vertices: one set covers the former vertices of the hypergraph (the metabolites in metabolic networks) and the other set of vertices covers the former edges of the hypergraph (the reactions), see Fig. 3b. This transformation allows the application of the usual methods for the analysis and visualisation of graphs to biochemical reaction networks.

The graphical representation of reversible reactions by arrows at both ends of a hyperedge or an edge has a disadvantage: the information about substrates and products is easily lost. In Fig 3a the reaction is reversible and by using the convention that substrates are at the top and products are at the bottom it is clear from the drawing, that the stoichiometric equation is `UDPglc + PP <=> G1P + UTP`. The bipartite representation in Fig. 3b shows the same information, however, this information might be lost in usual applications with analysis steps not considering the relative positions of the elements and automatic layouts which may mix substrates and products. It may be, for example, possible that the equation could be read as `UDPglc <=> G1P + UTP + PP`. A simple solution to this problem is shown in Fig. 3c. Here the reversible reaction is split into two separate reactions. This representation is usually used by the Petri net modelling community [5, 11, 16]. However, often the two reactions are represented by one “meta” reaction again, called a hierarchical reaction. This representation is similar to Fig. 3b. As hierarchical reactions improve the overall readability but remove a clear direction from reactions within SyBME we have chosen to adopt a similar representation as shown in Fig. 3d. We show the direction from substrates to products with a filled arrow and for the inverse reaction we use an open arrow. This representation is close to the hypergraph representation (Fig. 3a), is unambiguous and allows the use of common graph algorithms for the analysis and visualisation of biochemical reaction networks as well as the inference of the stoichiometric equations from this network.

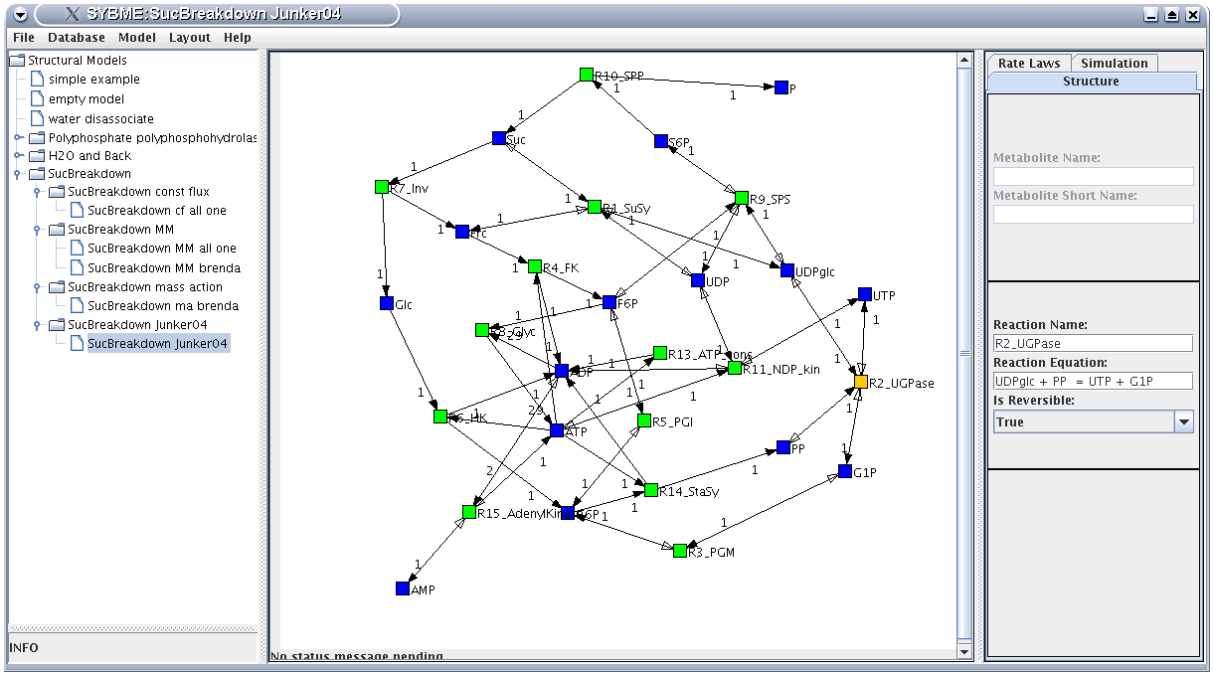

Figure 4: The SyBME graphical user interface. On the left side the hierarchy of models is visible. In the middle the active model, sucrose breakdown, is shown and on the right side the reaction equation for reaction UGPase is shown.

2.5 The SyBME Graphical User Interface

Giving the user the opportunity to see the structure of the biochemical reaction network was one of the key motivations for developing SyBME. Therefore the visualisation of the network consumes the largest amount of space within the graphical user interface shown in Fig. 4. The internal representation and the visualisation of the networks uses the Java Universal Network/-Graph Framework (JUNG) [8].

On the left side of the window the hierarchy of models is shown (see Sect. 2.3). Several different complete models exists, e.g., “SucBreakdown MM brenda”. Some of them are based on the same rate law model (both “SucBreakdown MM all one” and “SucBreakdown MM brenda” are based on “SucBreakdown MM”) and all are based on the same stoichiometry model “SucBreakdown”. The user is able to copy models and create new sub models of existing ones.

On the right side of the window the different parts of the model can easily be modified. The three tabs on the top allow the access to the different layers of the model: for example, the “Structure” tab shows either the information for a selected metabolite (top part) or, as currently shown, the information for the selected reaction.

As explained above (see Sect. 2.4) the visualisation of reactions in SyBME uses different arrowheads to distinguish substrates from products. This can be seen at the arrows for the substrates and products of the reaction “R2_UGPase” on the right side of the network window.

2.6 Simulation with SyBME

Numerical simulations are important for biological research as they allow *in silico* experiments which might give an idea about the results of tedious laboratory experiments altering the ex-

pression level of enzymes. In the current version of SyBME two simulators for biochemical reaction networks are available to the user: Gepasi [12] and Jarnac [17]. Both of them facilitate the computation of steady states of the metabolites and flux information for the reactions.

The simulators are connected to SyBME via text files im- and export. For Jarnac a script file as described in the manual and for Gepasi an SBML (Systems Biology Markup Language, [7]) Level 1 file are generated. SyBME controls Jarnac and performs the desired computation based on the generated script file. Gepasi is started by the user and the necessary steps have to be performed by hand.

The result of a simulation can be visualised as time series data within VANTED, see Sect. 3.2.

3 Results

3.1 A kinetic model of sucrose breakdown in developing potato tubers

The potato tuber is a model system for sink organs. The pathway by which incoming sucrose is transformed into starch is well understood through numerous experiments altering the activity of one or more enzymes of this pathway [3]. To better understand these processes, a detailed kinetic model has been constructed in an earlier study [9]. The starting point of the model is sucrose being imported into the cytosol, and the end points are hexose phosphates utilised for glycolysis and starch synthesis (Fig. 5). The kinetic model is described by starting concentrations for each of the 15 metabolites. The velocity of each enzymatic step is determined by a rate law that is in most cases based on Michaelis-Menten kinetics including inhibitory terms. The 14 enzymatic steps are characterised by a total number of 74 constants [9]. Upon simulation the model finds a steady state with metabolite concentrations similar to those reported in the literature. When overexpression of the enzyme invertase is simulated with the model, the observed changes in steady state metabolite concentrations and fluxes are in close accordance to the published values [9].

3.2 Visualisation of simulation results

In SyBME, the final version of the sucrose breakdown model has been initialised with metabolite concentrations reported in the literature [9]. Subsequently, a SBML file has been automatically created. Then the model has been simulated in Gepasi for 1000 steps of 0.1 seconds. The results were imported and visualised with VANTED [10], see Fig. 6. It is obvious that the system takes some time to find the right direction that allows it to converge towards a steady state.

4 Conclusion

SyBME is a versatile tool for modellers. Models can be designed and visualised in the GUI and stored in the built-in model repository. For simulation purposes, SyBME communicates

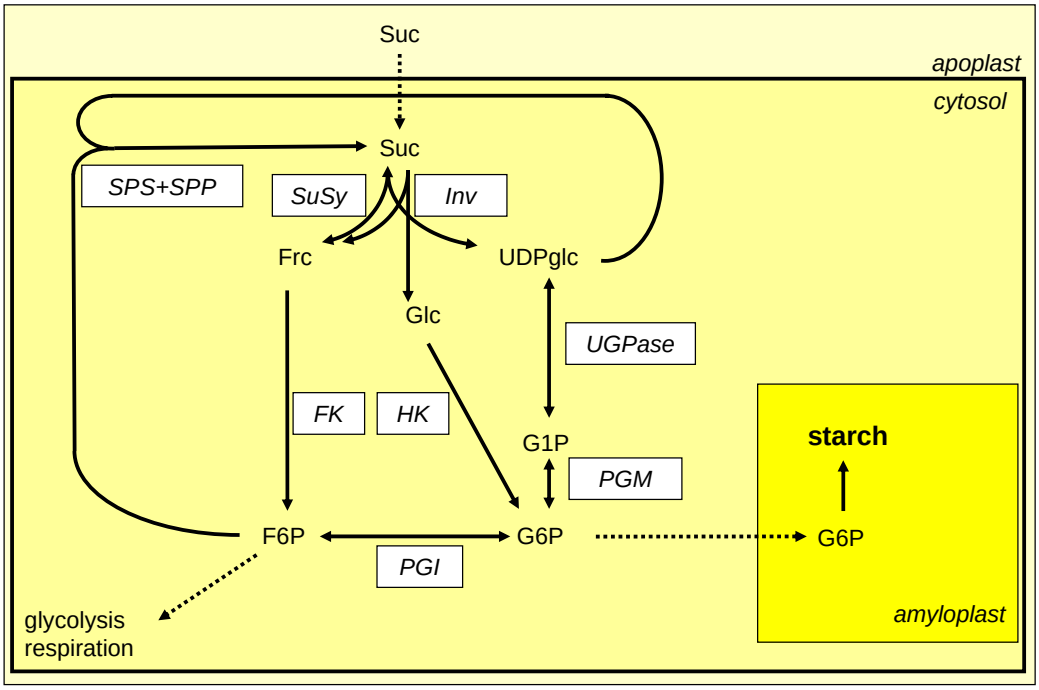

Figure 5: The pathway by which sucrose is transformed to starch in the potato tuber. The reactions take place in different cellular compartments. Abbreviations: FK, fructokinase; HK, hexokinase; Inv, invertase; PGI, phosphoglucose isomerase; PGM, phosphoglucomutase; SPS, sucrose phosphate synthase; SPP, sucrose phosphate phosphatase; SuSy, sucrose synthase; UGPase, UDP glucose pyrophosphorylase; others, see Fig. 6.

with the simulation tools Gepasi and Jarnac. With the help of the tool VANTED, the simulation results obtained by the simulators can also be visualised in the context of the underlying networks.

As a future development, it is planned to enable in addition to the existing SBML output also model import in this format. Furthermore, we plan to integrate extensive reporting functionalities, a tighter coupling to the simulators and VANTED, and the automatic extraction of models from databases.

5 Acknowledgements

The German Ministry of Education and Research (BMBF) supported part of this work under grants 0312706A and 0313115. We like to thank Christian Klukas for his work on VANTED and the developers of the open source software packages for their valuable work.

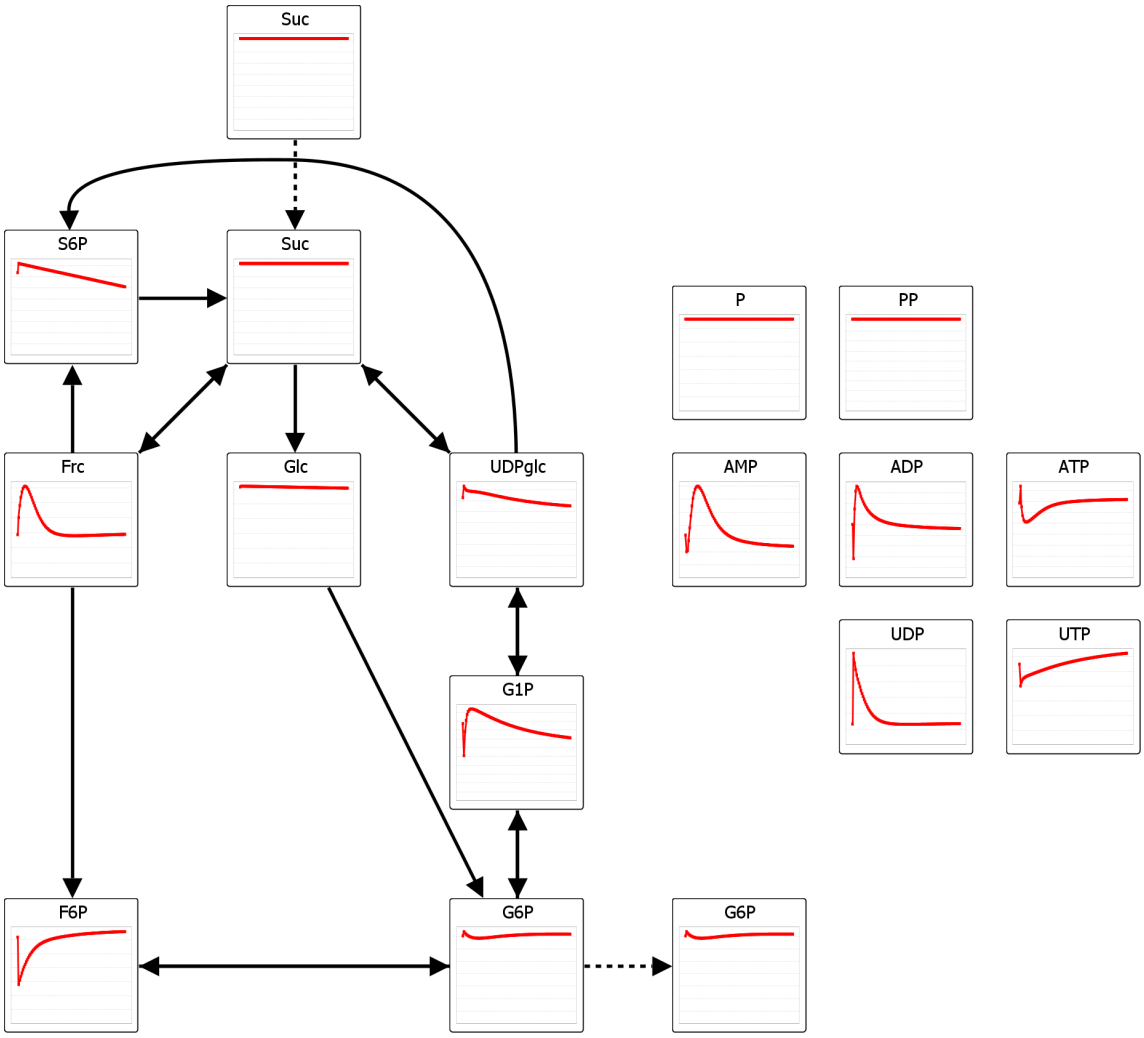

Figure 6: Simulation results of Gepasi for the pathway shown in Fig. 5, visualised in VANTED. Suc, PP and P concentrations were fixed. Abbreviations: ADP, adenosine diphosphate; AMP, adenosine monophosphate; ATP, adenosine triphosphate; Frc, fructose; F6P, fructose 6-phosphate; Glc, glucose; G1P, glucose 1-phosphate; G6P, glucose 6-phosphate; P, phosphate; PP, pyrophosphate; Suc, sucrose; S6P, sucrose 6-phosphate; UDPglc, uridine diphosphate glucose; UDP, uridine diphosphate; UTP, uridine triphosphate.

References

[1] T. Aho. Simulation tool for genetic regulatory networks. Master's thesis, Tampere University of Technology, Finland, 2003.

[2] Fat Client. http://en.wikipedia.org/wiki/fat_client.

[3] P. Geigenberger. Regulation of sucrose to starch conversion in growing potato tubers. *Journal of Experimental Botany*, 54:457–465, 2003.

[4] Hibernate. http://www.hibernate.org.

[5] R. Hofestädt and S. Thelen. Quantitative modeling of biochemical networks. *In Silico Biology*, 1:39–53, 1998.

[6] HSQLDB. http://www.hsqldb.org.

[7] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. Goryanin, W. J. Hedley, T. C. Hodgman, J. H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. L. Novere, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19:524–531, 2003.

[8] Java Universal Network/Graph Framework. http://jung.sourceforge.net.

[9] B. H. Junker. *Sucrose breakdown in the potato tuber*. PhD thesis, Potsdam University, Germany, 2004.

[10] B. H. Junker, C. Klukas, and F. Schreiber. VANTED: A system for advanced data analysis and visualization in the context of biological networks. Submitted, 2005.

[11] I. Koch, B. H. Junker, and M. Heiner. Application of petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. *Bioinformatics*, 21:1219–1226, 2005.

[12] P. Mendes. GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems. *Computer Applications in the Biosciences*, 9:563–571, 1993.

[13] B. G. Olivier, J. M. Rohwer, and J.-H. S. Hofmeyr. Modelling cellular systems with PySCeS. *Bioinformatics*, 21:560–561, 2005.

[14] Oracle Corporation. http://www.oracle.com.

[15] PostgreSQL. http://www.postgresql.org.

[16] V. N. Reddy, M. L. Mavrovouniotis, and M. N. Liebman. Petri net representations in metabolic pathways. In L. Hunter, D. B. Searls, and J. W. Shavlik, editors, *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 328–336. AAAI, 1993.

[17] H. M. Sauro. Jarnac: a system for interactive metabolic analysis. In J.-H. Hofmeyr, J. M. Rohwer, and J. L. Snoep, editors, *Animating the Cellular Map: Proceedings of the 9th International Meeting on BioThermoKinetics*, pages 221–228. Stellenbosch University Press, 2000.

[18] M. Tomita, K. Hashimoto, K. Takahashi, T. S. Shimizu, Y. Matsuzaki, F. Miyoshi, K. Saito, S. Tanida, K. Yugi, J. C. Venter, and C. A. Hutchison. E-cell: software environment for whole-cell simulation. *Bioinformatics*, 15:72–84, 1999.

[19] VMWare, Inc. http://www.vmware.com.

[20] WINE HQ. http://www.winehq.com.