



Review

COPASI and its applications in biotechnology[☆]

Frank T. Bergmann^{a,*}, Stefan Hoops^b, Brian Klahn^b, Ursula Kummer^{a,**}, Pedro Mendes^c, Jürgen Pahle^d, Sven Sahle^a

^a BioQuant/COS, Heidelberg University, Heidelberg, Germany

^b Biocomplexity Institute of Virginia Tech, Blacksburg, VA, USA

^c Center for Quantitative Medicine, UConn Health, Farmington, CT, USA

^d BIOMS/BioQuant, Heidelberg University, Heidelberg, Germany

ARTICLE INFO

Keywords:

COPASI
Biotechnology
Systems biology

ABSTRACT

COPASI is software used for the creation, modification, simulation and computational analysis of kinetic models in various fields. It is open-source, available for all major platforms and provides a user-friendly graphical user interface, but is also controllable via the command line and scripting languages. These are likely reasons for its wide acceptance. We begin this review with a short introduction describing the general approaches and techniques used in computational modeling in the biosciences. Next we introduce the COPASI package, and its capabilities, before looking at typical applications of COPASI in biotechnology.

1. Introduction

Computational modeling has been used in the context of biology for decades. Early studies on enzyme kinetics have already employed small computational models and their simulation (e.g. Chance, 1949). Also, in the context of population ecology and epidemiology, computational modeling has been involved for many years to describe and predict population sizes of species or to analyze the spread of diseases. Cell biological questions, e.g. the investigation of the mechanisms behind calcium oscillations (for a review see e.g. Schuster et al., 2002) or the understanding of the regulation of glycolysis (Sel'Kov, 1968) are another context in which models have played a role since the 1960's. However, despite the successes and the knowledge about the usefulness of the respective techniques, usage only started to virtually explode with the turn of the century, when systems biology developed as its own discipline (Hübner et al., 2011). This is probably due to the advances both on the experimental side (e.g. live-cell imaging, high-throughput data), as well as on the computational side (widespread availability of computers, faster machines, more and tailored algorithms). Another likely contributor to this virtual explosion in computational modeling, has been the emergence of user-friendly software, which makes model set-up and analysis much more accessible, and practical, for non-mathematicians.

Different computational methods are employed on different scales of abstraction and detail. In the areas of biochemistry and cell biology

the two largest classes (Hübner et al., 2011) of models are: (a) stoichiometric models including constraint based genome scale models, and (b) kinetic models. The former are based on static reaction schemes and used to calculate potential flux distributions mostly in metabolic networks. The latter allows for a more detailed picture and for more fine-grained analysis, but also demands more knowledge, namely about the kinetics of individual processes.

For these two classes of computational models the German Network for Bioinformatics Infrastructure, de.NBI, hosts two widely used resources. Stoichiometric models, including genome scale constraint based models, can be set-up and analyzed using CellNetAnalyzer (Klamt et al., 2007), whereas COPASI (Hoops et al., 2006; Mendes et al., 2009) specializes in setting up and analyzing kinetic models while also providing some basic stoichiometric analyses.

Computational models are widely employed in biotechnology. The following sections will go into some detail about this. Generally speaking, often the goal is to manipulate systems to either maximize the yield, flux, or titer of desirable end-products, or to minimize the amounts or fluxes of undesirable side-products.

Below, we will first give an overview of the current features of the COPASI software. Then we describe its application in the context of biotechnology by presenting concrete examples where COPASI was applied to systems of biotechnological interest with varying levels of complexity.

[☆] Authors are in alphabetical order.

* Corresponding author.

** Co-corresponding author.

E-mail address: frank.bergmann@bioquant.uni-heidelberg.de (F.T. Bergmann).

2. The COPASI software

COPASI (Hoops et al., 2006; Mendes et al., 2009, 2009b) is a modeling and simulation environment, jointly founded by the groups of Pedro Mendes (now at UConn Health) and Ursula Kummer (now at Heidelberg University). Currently the groups of Stefan Hoops, Jürgen Pahle, and Sven Sahle are also part of the consortium. All 3 have been former group members and are now PIs on their own, and are actively participating in development of the software. The software is platform-independent, developed as open source, and distributed available under the terms of the Artistic License 2.0,¹ which makes it free to use for any purpose. Source and binary distributions are available from the COPASI Website (<http://copasi.org>) for Windows, Linux, and Mac OS X. This makes it very easy for researchers to install COPASI and get started quickly. Those who prefer it can also download the source code and compile their own specific versions of the software.

2.1. Features

Model editing. Models are entered in (bio)chemical terms. (Bio)chemical species, located in compartments, can be transformed through reactions. The rate of each reaction is given by a kinetic rate law, that can be either selected from an included function database (filtered to apply to the selected reaction), or freely defined. Additionally, discrete events can be added to the model, that change model elements based on arbitrary expressions. From this formalism, COPASI derives the mathematical representation of the model automatically. Thus users do not have to specify the differential equations themselves. However, expert users can add additional differential equations to the model, including for compartment volumes or any other variable (i.e. not just chemical species).

Model standards. COPASI stores the model and information what to do with it in its own specialized XML format. However, wherever possible, applicable community standards are supported. Models can be imported and exported in the Systems Biology Markup Language (SBML, Hucka et al., 2003) format, where all major versions are supported. This enables COPASI to consume models from a variety of on-line model repositories (such as the BioModels Database Le Novère et al., 2006), as well as from many other software tools that support the SBML standard.

Where possible, the analysis tasks are exported to the Simulation Experiment Description Markup Language (SED-ML, Waltemath et al., 2011). Specifically, time course simulations and parameter scans can be exported to SED-ML, as SED-ML does not specify further simulation types yet.

Currently the COPASI team is extending COPASI to additionally support the COMBINE Archive (Bergmann et al., 2014) which will make it easier to exchange COPASI models together with experimental data.

Simulation algorithms. At the core, COPASI supports two main formalisms for simulating the dynamics of the defined model (time course simulations). One is the traditional chemical kinetics approach of using ordinary differential equations (ODE), where the software uses the LSODAR integrator (Petzold, 1983). The second is the stochastic formalism (Pahle, 2009), where individual reaction events are drawn from probability distributions, using either of the following algorithms: Gillespie's Direct Method (Gillespie, 1976), Gibson–Bruck (Gibson and Bruck, 2000), τ -Leap, or adaptive SSA/ τ -leap (Cao et al., 2006). To aid users in adapting deterministic rate equations into their stochastic equivalent, COPASI features an option that splits up reversible reactions into their forward and backward-directional forms. Additionally there is an option to apply stochastic corrections to rate equations.

Since stochastic simulations can be rather time consuming, a third class of simulation algorithms available consist of hybrid solvers. The

solvers “Hybrid (Runge–Kutta)” and “Hybrid (LSODA)” employ a dynamic partitioning strategy that treats all reactions involving species that have a particle number below a specified threshold with the stochastic approach, while the remaining reactions are treated with the differential equation approach. A new solver was recently introduced, “Hybrid (RK-45)”, which allows modelers to define this partitioning themselves, and thus control which reactions are to be simulated stochastically or deterministically.

In the future, solvers for Stochastic Differential Equations (SDEs) and Delay Differential Equations (DDEs) will also be added.

Analysis tasks. The model evaluations through time course simulations or steady state computations at the core of COPASI support a number of additional analysis tasks.

- **Parameter scan:** The parameter scan task can be used to run any of the other tasks repeatedly. Frequently one needs to examine how the model behaves when some parameter(s) vary, and this task allows programming this parameter variation in an easy way. Parameter values can be either “scanned” in a given interval (i.e. their values changed through regular intervals), or sampled from a random distribution; these methods can be intermixed such that some parameters can be scanned and others sampled. This can be applied to time course simulations or steady state simulations, but also to any other task in this list. This allows for complex operations, such as multi-dimensional parameter scans, repeating stochastic time course simulations to determine the time-dependent probability distributions for each model variable, creating bifurcation plots (running the cross section task for different values of a parameter), evaluating the fitness landscape around an optimization solution, or even determining profile likelihood plots (Schaber, 2012).
- **Optimization:** In the optimization task users can enter an arbitrary expression of model quantities (or even results from the execution of other tasks) that should be maximized or minimized. Additionally the model elements that should be modified, where appropriate with optional constraints, are specified. COPASI implements a wide selection of local (Hooke & Jeeves: Hooke and Jeeves, 1961, Levenberg–Marquardt: Levenberg, 1944, Nelder–Mead: Nelder and Mead, 1965, Praxis: Brent, 1973, Steepest Descent, Truncated Newton: Nash, 1984) and global optimization methods (Differential Evolution: Storn and Price, 1997, Evolutionary Strategy: Runarsson and Yao, 2000, Evolutionary Programming: Fogel et al., 1992, Genetic Algorithm: Michalewicz, 1994, Particle Swarm: Kennedy and Eberhart, 1995, Scatter Search: Egea et al., 2009, Random Search, Simulated Annealing: Corana et al., 1987), that can be used for the optimization.
- **Parameter estimation:** Parameter estimation is the activity of changing the model's parameter values in order to make the behavior of the model as close as possible to a set of experimental measurements. Parameter estimation is a special form of optimization, in which the distance between the model and the experimental data is minimized. This is achieved by changing user-selected model parameters through specific optimization algorithms (the same ones that are available for general optimization described above). COPASI supports both experimental data in the form of time courses and steady state measurements; several data sets of either type can be mixed together. Users can specify as many data sets as required. A new feature allows validation data sets to be specified as well. These experimental data sets are not used during the fitting process, but only during a validation phase to measure how robust the model predictions are. The middle window in Fig. 1 shows the definition of parameters to be modified in a parameter estimation run.
- **Metabolic control analysis:** Metabolic control analysis (MCA) is a special type of sensitivity analysis that calculates how much a perturbation in the rate of a reaction affects the steady state concentrations or fluxes (Fell, 1996). These are systems level coefficients and are called “control coefficients”. MCA establishes a link

¹ <https://opensource.org/licenses/Artistic-2.0>.

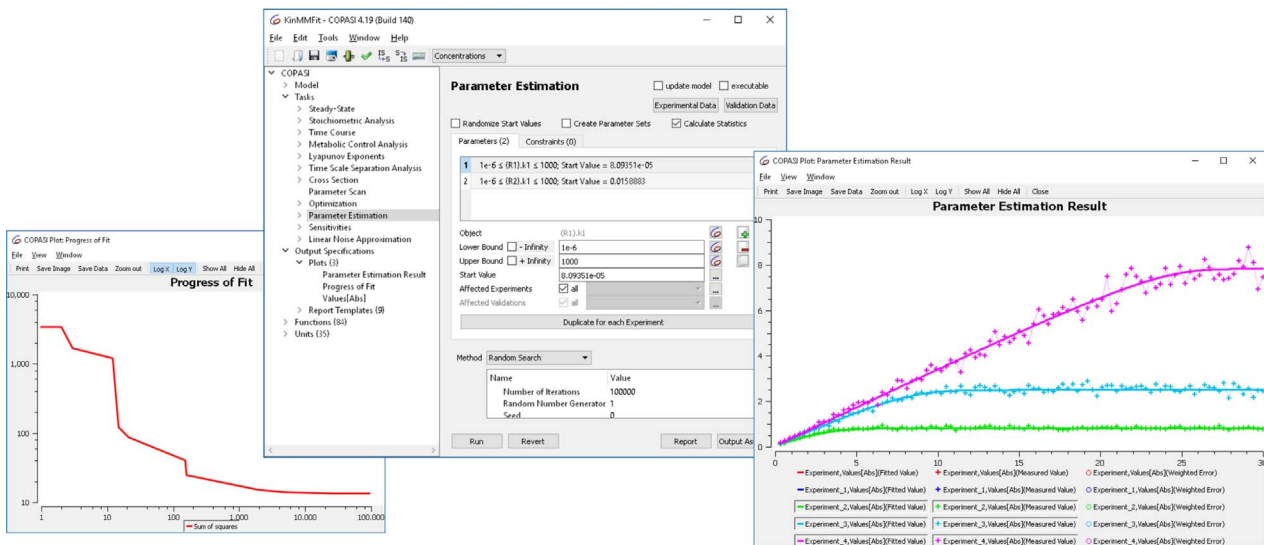


Fig. 1. Parameter estimation using COPASI. The left window shows a progress graph, that is updated during the parameter estimation run and shows the current objective value. While the numeric value is not of particular interest, any reduction in its value demonstrates an improvement in the fit. In the middle image the setup of parameters and the specification of the method is displayed. The right image shows a plot of the parameter estimation result. For each measured quantity the fit result is being displayed (line) along with the experimental data (points).

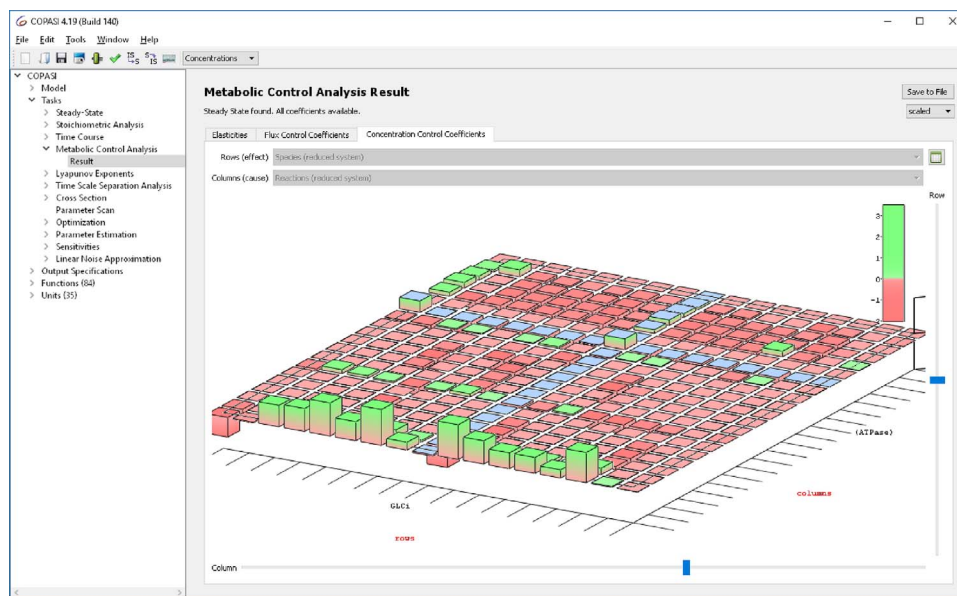


Fig. 2. Metabolic control analysis results displayed in a 3D widget that makes it easy to identify contributions of individual reactions to the control of concentrations or fluxes.

(Reder, 1988) between the control coefficients and “elasticity coefficients”, which are properties of each reaction in isolation (they are the derivatives of the kinetic rate law in terms of the reaction effectors). COPASI calculates the elasticities as well as flux and concentration control coefficients. In Fig. 2 a three dimensional representation of an MCA result is displayed.

- **Sensitivities:** Sensitivities provide a measure of how much a selected model variable (the *effect*) changes when a selected parameter (the *cause*) is changed. COPASI includes a pre-defined set of effects and causes, but users are also free to select any specific model elements as well. Sensitivities are always calculated based on another task (selected by the user): time courses, steady states, optimization, parameter estimation, or cross sections. Second order sensitivities are also available, where the user needs to select two causes (this is essentially an estimate of a second order derivative). The control coefficients of MCA are actually a subset of the sensitivities for a steady state, however they are calculated with a different approach (using Reder's matrix method).

- **Linear noise approximation:** The linear noise approximation (LNA) task estimates the covariances between species particle numbers (Pahle et al., 2012; Challenger et al., 2012) without time-demanding stochastic simulations. It sets up a Lyapunov equation for the system in steady state and then solves this equation for the covariances using the Bartels-Stewart algorithm. This task can be used in a closed-loop fashion with parameter scans or optimization to efficiently find parameter sets where, e.g. the coefficient of variation of a species' particle number is low or where the correlation between two species is high.
- **Time scale separation analysis:** The time scale separation analysis task (Surovtsova et al., 2009) provides information on time scales of different parts of the model (and at different times in the dynamics). This is accomplished by simulating the model for a user specified duration and analyzing the time-dependent Jacobian matrix in order to decompose species and reactions into different time scales.
- **Lyapunov exponents:** Lyapunov exponents provide a means to characterize the dynamical behavior of a model, including an indication

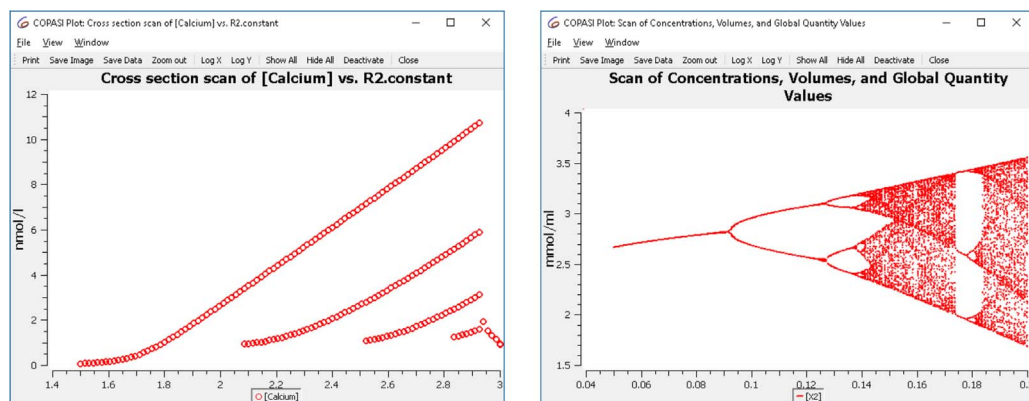


Fig. 3. Setting up a parameter scan over a cross section task makes it possible to create bifurcation diagrams in COPASI. Left we see a change of oscillation behavior in a calcium model (Kummer et al., 2000), from simple oscillation to bursting as the number of peaks per oscillation change from one to four. Right we see the period doubling route to chaos in an abstract model (Baier and Sahle, 1995). In both cases a bifurcation parameter is scanned (x-axis) and the maxima of one variable of the model are plotted after the system has settled on the respective attractor (y-axis).

if the model has chaotic behavior (sensitivity to initial conditions). COPASI implements the Wolf method (Wolf et al., 1985) for calculating Lyapunov exponents from time series data. A user only has to specify how many exponents should be calculated.

- **Cross section:** The cross section task facilitates the detection of crossings through a hyper surface in phase space. The user defines the surface by selecting a model variable and a certain value, then, when COPASI runs a simulation, the cross section task will only yield an output point when the specified value is crossed in a specified direction. In conjunction with the scan task, the cross section task makes it possible to create bifurcation diagrams (displayed in Fig. 3). A step by step tutorial to create these plots is available in our users manual.²

2.2. COPASI user interface

Users have several options for running COPASI. The most common option is to run the interactive graphical user interface (GUI), which was written in such a way that it works natively on all operating systems. A command line version of COPASI is also available, which allows for processing simulations without a graphical interface and which is useful for running simulations offline on servers, etc. The most frequent use case involving the command line version is to prepare a COPASI file using the GUI, mark the desired task (such as parameter estimation) as executable, and then move the resulting COPASI file to a cluster environment where it is run by the command line version and the results are saved in files. Additional to these two versions of COPASI, libraries are also provided with which COPASI can be used from inside programs written in all major programming languages.

While the computations executed by the command line version and the libraries are the same as with the GUI there are two features only present when using COPASI through the graphical user interface:

- **Plotting.** Plots of several types can be used for visualization of results of all analysis tasks in COPASI. This includes basic scatter and line plots, histograms, and contour plots (a recent addition). Fig. 1 displays a combination of several plots as used during parameter estimation. Whenever results are obtained in matrix form (as is the case for stoichiometric analysis of the model, calculation of sensitivities and the like), they are displayed in a sortable table by default. However, users can also opt to have the results displayed in a three dimensional representation as is shown in Fig. 2.
- **Network diagrams.** The GUI also has support for model diagrams. The diagrams can be created from scratch, using a force based layout algorithm, or they can be imported from other software tools using either the SBML Level 3 Layout (Gauges et al., 2015) and Rendering package format or the CellDesigner (Funahashi et al., 2008) format.

These diagrams can be used to display the result of a time course simulation (as an animation) or the decomposition of the network in elementary flux modes (Schuster et al., 1999).

2.3. Support/training

Apart from a user manual³ describing all COPASI features in detail, we regularly provide hands-on tutorials both in our own institutions, at conferences, or other venues. Additionally, online training videos can be found on our website that describe basic and advanced features of COPASI. The COPASI user forum⁴ is the best way to quickly get in touch with the COPASI community. Not only will the COPASI developers provide answers to questions asked, but also the other members of the forum provide answers to questions.

3. Applications in biotechnology

COPASI is widely used in a number of scientific fields. Although its original main purpose was for research on biochemical systems, COPASI is also used in other fields such as chemistry or ecology. We have investigated the usage of the software by carefully analyzing publications where it has been applied and classifying its research. We wanted to differentiate between citations that just mention COPASI, and actual studies that used the software for their scientific work. This analysis is ongoing and updated results can be seen at our webpage (<https://www.copasi.org/Research>). A visualization of the locations where these studies have been published from is displayed in Fig. 4.

Here we want to focus on applications of the software in studies targeting biotechnological applications. In this context, and similar to other fields, COPASI is used for modeling and computational analysis of systems at different levels of granularity and complexity — from single enzymes to large cellular networks. In the following we want to show typical use cases that represent the most common scenarios in literature. A complete review of the literature would be far beyond the scope of this article.

At the molecular scale, the general goal is usually to determine the exact mechanism of biotechnologically relevant enzymes and their regulation, and to use this information to influence the reaction in order to e.g. increase the yield of a desired product. Pursuing this goal involves either one of two quite common scenarios. First of all, scientists use the software often to calculate steady state properties in conjunction with structural analysis and/or molecular modeling studies. Thus, Pingoud et al. (2009) investigated the divalent metal ion dependence of restriction endonucleases of the EcoRI family. In their extensive study, they experimentally determined binding properties of different enzymes of this family with respect to Mg^{2+} and Mn^{2+} and to a lesser

² https://copasi.org/Support/User_Manual/Tasks/Cross_Section/.

³ https://copasi.org/Support/User_Manual/.

⁴ <https://groups.google.com/forum/#!forum/copasi-user-forum>.

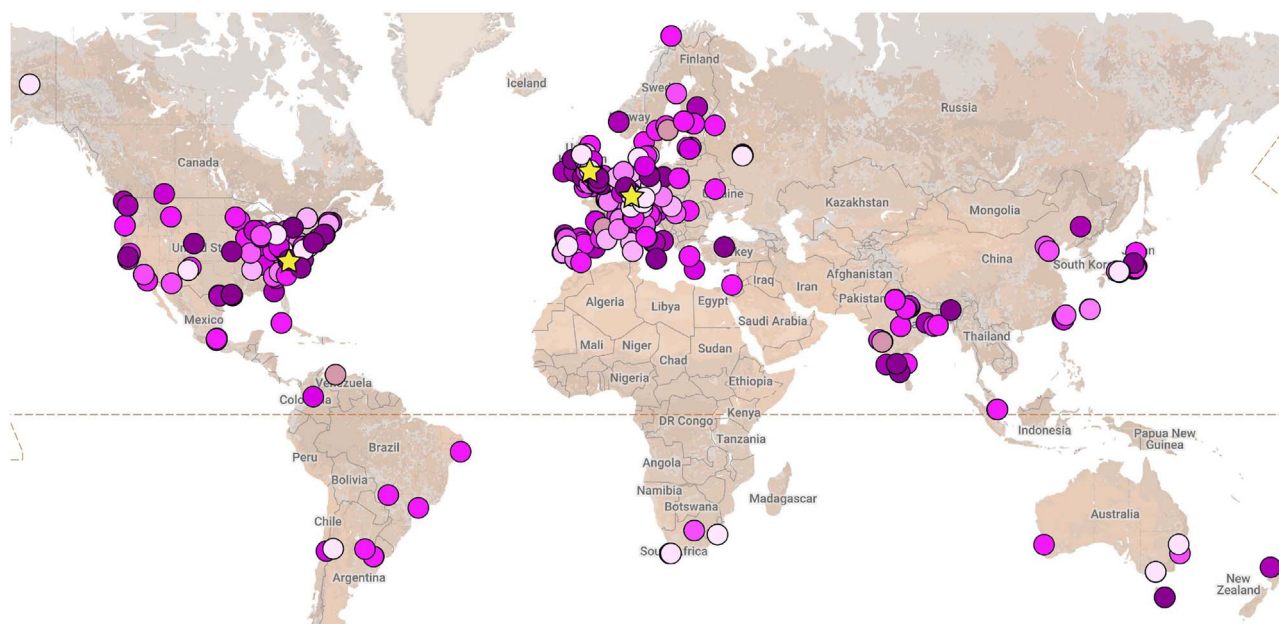


Fig. 4. Manually curated world map of locations from where research has been carried out with COPASI (judged from their author's addresses).

extend Ca^{2+} and their subsequent catalytic properties. They used the experimental results together with insights from molecular dynamics simulations to generate an extensive elementary reaction scheme for the kinetic mechanism of the enzymatic catalysis of DNA cleavage which involves binding and unbinding of different metal ions at two different binding sites. This scheme was entered as a model into COPASI and the software was used to calculate steady state distributions in different scenarios. The authors conclude that the mechanism of this class of biotechnologically relevant enzymes involves one metal ion binding site essential for the enzymatic activity, and a second one which modulates this activity.

The second common scenario on the molecular scale also involves the set-up of a model for the kinetics of an enzyme. But instead of experimentally or computationally (via molecular dynamics simulations) determined kinetic parameters it uses the parameter estimation task to estimate the values of the parameters of interest. As an example, Kashyap et al. (2015) used COPASI to investigate the kinetics and mechanism of the formation of fructo-oligosaccharides using fructosyltransferase. They experimentally measured time courses and enzyme activity for various conditions. They then set up a detailed model of the elementary reactions involved in the catalytic mechanisms of this enzyme, including binding of glucose which leads to an inhibition of the enzyme. The parameters of the model were fitted with COPASI using an extensive amount of experimental measurements. In this case, the genetic algorithm was the optimization routine of choice. The result of the fitting process was statistically analyzed and the corresponding parameter values were used in simulations reproducing further experimental conditions, now no longer fitting the model but rather validating the accuracy of its predictions. Simulations were performed with the LSODA integrator of COPASI. The authors conclude how the yield of the biotechnological product, namely the fructo-oligosaccharides can be optimized with respect to the amount of substrate and enzyme.

Moving to a more complex scale, biotechnologically relevant pathways are of course also subject to similar questions, e.g. deciphering their mechanism and regulation and thus learning how to manipulate them. Therefore, also quite similar scenarios can be found as described for the molecular scale. Thus, Ma et al. (2009) used COPASI to model nitrogen assimilation in *Escherichia coli*. They were especially interested in understanding how this process is regulated at low ammonium concentrations. They used COPASI to set up a kinetic model describing

all essential reactions participating in nitrogen assimilation in this bacterial species. Most of the parameters and concentrations had been experimentally determined. They then calculated steady-state concentrations and fluxes at different ammonium concentrations. Subsequently, they investigated the influence of the expression of AmtB (Am transport protein B) on the ability of *E. coli* to take up ammonium. Furthermore, the behavior of the system at ammonium impulses was studied. For this purpose, simulations with the LSODA algorithm were performed.

In a similar context, but with a larger system and more unknown parameters, Raganati et al. (2015) studied butanol production from different sugars by *Clostridium acetobutylicum*. The authors were interested in the different products and the yield in order to be able to optimize the respective fermenters. They set up an extensive model using COPASI and used parameter estimation with experimental time course data to estimate the unknown parameter values. In this instance they used the simulated annealing algorithm. It is noteworthy that — as often reported — every set of model and experimental data has individual requirements that can not be satisfied by one common single optimization routine. Rather, different algorithms are efficient for different data, without a clear pattern (i.e. there is no prescribed rule that associates algorithms with specific types of data or problem). Thus this is why COPASI offers many different optimization algorithms and allows users easily to switch from one to another. The final model was used to simulate the behavior of the system with respect to different sugars.

In contrast to the molecular level, kinetic models of whole pathways are often analyzed via metabolic control analysis (MCA). An example is the study by Trausinger et al. (2015) who studied the metabolic interactions influencing carbon flux from xylose to ethanol in different yeast strains. Here, a kinetic model of this pathway was first subject to parameter estimation in COPASI (using evolutionary programming as optimization routine). The model was then studied using MCA to identify those reactions that have the largest influence on the conversion and therefore, would be the most promising targets for manipulation.

Apart from the expected usage of the software as described above, we were actually surprised not to find certain usages in biotechnological applications. Of course, they could be out there and simply not yet included in our analysis. We had expected that optimization algorithms

would be more commonly used to predict conditions to optimize biotechnological strains (as some of us argued quite some time ago, Mendes and Kell, 1998). The optimization task in COPASI allows one to ask questions such as “What are the conditions for maximal yield?” and then to let parameters that are amenable to experimental manipulation (e.g. expression levels) be adjusted for the optimal combination. Perhaps we ought to put some more emphasis on this approach in our outreach and training efforts.

Acknowledgments

Over the years, COPASI has received funding from the following funding agencies and programmes: the BMBF funded de.NBI initiative (O31L0104A) and Virtual Liver Project, National Institutes of Health (NIGMS Grant: GM080219), the BBSRC (grant BB/J019259/1), BBSRC and EPSRC (grant BB/C008219/1) as well as the Klaus Tschira Foundation and the Virginia Bioinformatics Institute. We would like to thank Norman Paton and Yang Cao for their collaboration. We would also like to thank all the programmers and student helpers that worked on COPASI over the years, particularly, Ralph Gauges, Joseph Dada, Ettore Murabito, Peyman Zarrineh, Natalia Simus and Irina Surovtsova. Marina Kaiser is acknowledged for literature digging. Finally, we would like to acknowledge the members of the COPASI user forum for their active support.

References

- Baier, G., Sahle, S., 1995. Design of hyperchaotic flows. *Phys. Rev. E* 51, R2712–R2714.
- Bergmann, F.T., Adams, R., Moodie, S., Cooper, J., Glont, M., Golebiewski, M., Hucka, M., Laibe, C., Miller, A.K., Nickerson, D.P., et al., 2014. COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC Bioinform.* 15, 1.
- Brent, P.R., 1973. A new algorithm for minimizing a function of several variables without calculating derivatives. In: Brent, P.R. (Ed.), *Algorithms for Minimization Without Derivatives*. Prentice-Hall, Inc., Englewood Cliffs, NJ, pp. 117–167.
- Cao, Y., Gillespie, D.T., Petzold, L.R., 2006. Efficient step size selection for the tau-leaping simulation method. *J. Chem. Phys.* 124, 044109.
- Challenger, J., McKane, A., Pahle, J., 2012. Multi-compartment linear noise approximation. *J. Stat. Mech.* 2012, P11010.
- Chance, B., 1949. The reaction of catalase and cyanide. *J. Biol. Chem.* 179, 1299–1309.
- Corana, A., Marchesi, M., Martini, C., Ridella, S., 1987. Minimizing multimodal functions of continuous variables with the “simulated annealing” algorithm. *ACM Trans. Math. Softw.* 13, 262–280.
- Egea, J.A., Balsa-Canto, E., García, M.-S.G., Banga, J.R., 2009. Dynamic optimization of nonlinear processes with an enhanced scatter search method. *Ind. Eng. Chem. Res.* 48, 4388–4401.
- Fell, D.A., 1996. *Understanding the Control of Metabolism*. Portland Press.
- Fogel, D.B., Fogel, L.J., Atmar, J.W., 1992. Meta-evolutionary programming. In: Chen, R.R. (Ed.), *25th Asilomar Conference on Signals, Systems & Computers*. IEEE Computer Society, pp. 540–545.
- Funahashi, A., Matsuoka, Y., Jouraku, A., Morohashi, M., Kikuchi, N., Kitano, H., 2008. CellDesigner 3.5: a versatile modeling tool for biochemical networks. *Proc. IEEE* 96, 1254–1265.
- Gauges, R., Rost, U., Sahle, S., Wengler, K., Bergmann, F.T., 2015. The systems biology markup language (SBML) level 3 package: layout, version 1 core. *J. Integr. Bioinform.* 12, 550–602.
- Gibson, M.A., Bruck, J., 2000. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem. A* 104, 1876–1889.
- Gillespie, D.T., 1976. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.* 22, 403–434.
- Hübner, K., Sahle, S., Kummer, U., 2011. Applications and trends in systems biology in biochemistry. *FEBS J.* 278, 2767–2857.
- Hooke, R., Jeeves, T.A., 1961. “Direct search” solution of numerical and statistical problems. *J. ACM* 8, 212–229.
- 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, 3067–3074.
- Hucka, M., et al., 2003. The Systems Biology Markup Language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19, 524–531.
- Kashyap, R., Palai, T., Bhattacharya, P.K., 2015. Kinetics and model development for enzymatic synthesis of fructo-oligosaccharides using fructosyltransferase. *Bioprocess Biosyst. Eng.* 38, 2417–2426.
- Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. *IEEE International Conference on Neural Networks*, vol. 4. IEEE Computer Society, pp. 1942–1948.
- Klamt, S., Saez-Rodriguez, J., Gilles, E.D., 2007. Structural and functional analysis of cellular networks with CellNetAnalyzer. *BMC Syst. Biol.* 1, 2.
- Kummer, U., Olsen, L.F., Dixon, C.J., Green, A.K., Bornberg-Bauer, E., Baier, G., 2000. Switching from simple to complex oscillations in calcium signaling. *Biophys. J.* 79, 1188–1195.
- Le Novère, N., Bornstein, B., Broicher, A., Courtot, M., Donizelli, M., Dharuri, H., Li, L., Sauro, H., Schilstra, M., Shapiro, B., Snoep, J.L., Hucka, M., 2006. BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res.* 34, D689–D691.
- Levenberg, K., 1944. A method for the solution of certain non-linear problems in least squares. *Q. Appl. Math.* 2, 164–168.
- Ma, H., Booger, F.C., Goryanin, I., 2009. Modelling nitrogen assimilation of *Escherichia coli* at low ammonium concentration. *J. Biotechnol.* 144, 175–183.
- Mendes, P., Kell, D.B., 1998. Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics* 14, 869–883.
- Mendes, P., Hoops, S., Sahle, S., Gauges, R., Dada, J., Kummer, U., 2009. Computational modeling of biochemical networks using COPASI. In: Maly, I.V. (Ed.), *Syst. Biol. Humana Press, Totowa, NJ*, pp. 17–59.
- Mendes, P., Messiha, H., Malys, N., Hoops, S., 2009b. Enzyme kinetics and computational modeling for systems biology. *Methods Enzymol.* 467, 583–599.
- Michalewicz, Z., 1994. *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed. Springer-Verlag, Berlin.
- Nash, S.G., 1984. Newton-type minimization via the lanczos method. *SIAM J. Numer. Anal.* 21, 770–788.
- Nelder, J.A., Mead, R., 1965. A simplex method for function minimization. *Comput. J.* 7, 308–313.
- Pahle, J., Challenger, J., Mendes, P., McKane, A., 2012. Biochemical fluctuations, optimisation and the linear noise approximation. *BMC Syst. Biol.* 6, 86.
- Pahle, J., 2009. Biochemical simulations: stochastic, approximate stochastic and hybrid approaches. *Brief. Bioinform.* 10, 53–64.
- Petzold, L., 1983. Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *SIAM J. Sci. Stat. Comput.* 4, 136–148.
- Pingoud, V., Wendt, W., Friedhoff, P., Reuter, M., Alves, J., Jeltsch, A., Mones, L., Fuxreiter, M., Alfred, P., 2009. On the divalent metal ion dependence of DNA cleavage by restriction endonucleases of the EcoRI family. *J. Mol. Biol.* 393, 140–160.
- Raganati, F., Procentese, A., Olivieri, G., Götz, P., Salatino, P., Marzocchella, A., 2015. Kinetic study of butanol production from various sugars by *Clostridium acetobutylicum* using a dynamic model. *Biochem. Eng. J.* 99, 156–166.
- Reder, C., 1988. *Metabolic Control Theory: A Structural Approach*, vol. 135. pp. 175–201.
- Runarsson, T., Yao, X., 2000. Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.* 4, 284–294.
- Schaber, J., 2012. Easy parameter identifiability analysis with COPASI. *Biosystems* 110, 183–185.
- Schuster, S., Dandekar, T., Fell, D.A., 1999. Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *Trends Biotechnol.* 17, 53–60.
- Schuster, S., Marhl, M., Höfer, T., 2002. Modelling of simple and complex calcium oscillations. *Eur. J. Biochem.* 269, 1333–1355.
- Sel’Kov, E., 1968. Self-oscillations in glycolysis. *Eur. J. Biochem.* 4, 79–86.
- Storn, R., Price, K., 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* 11, 341–359.
- Surovtsova, I., Simus, N., Lorenz, T., König, A., Sahle, S., Kummer, U., 2009. Accessible methods for the dynamic time-scale decomposition of biochemical systems. *Bioinformatics* 25, 2816–2823.
- Trausinger, G., Gruber, C., Krahulec, S., Magnes, C., Nidetzky, B., Klimacek, M., 2015. Identification of novel metabolic interactions controlling carbon flux from xylose to ethanol in natural and recombinant yeasts. *Biotechnol. Biofuels* 8, 157.
- Waltemath, D., Adams, R., Bergmann, F.T., Hucka, M., Kolpakov, F., Miller, A.K., Moraru, I.I., Nickerson, D., Sahle, S., Snoep, J.L., et al., 2011. Reproducible computational biology experiments with SED-ML – the simulation experiment description markup language. *BMC Syst. Biol.* 5, 1.
- Wolf, A., Swift, J.B., Swinney, H.L., Vastano, J.A., 1985. Determining Lyapunov exponents from a time series. *Phys. D: Nonlinear Phenom.* 16, 285–317.