

## Journal Information

Journal ID (publisher-id): vb  
Title: Virtual Biology  
ISSN (electronic): 2306-8140

## Article Information

Publication date (electronic): 25 March 2013  
Electronic Location Identifier: e10  
DOI: 10.12704/vb/e10

---

# BioUML plug-in for nonlinear parameter estimation using multiple experimental data

Elena Kutumova<sup>1, 2</sup>, Anna Ryabova<sup>1</sup>, Tagir Valeev<sup>1, 3</sup>, Fedor Kolpakov<sup>1, 2</sup>

[1] Institute of Systems Biology, Ltd, Novosibirsk, Russia

[2] Design Technological Institute of Digital Techniques SB RAS, Novosibirsk, Russia

[3] Institute of Informatics Systems SB RAS, Novosibirsk, Russia

---

## Abstract

*Motivation:* Systems biology deals with many different types of experimental data representing individual components of biological systems. Behavior of these systems over time could be described using systems of ordinary differential equations (ODE). In order to analyze dynamics of the ODEs and estimate their parameters based on data obtained in different experimental conditions, biologists need a flexible framework that allows them to create dynamic models and perform multi-experiment parameter fitting.

*Results:* We present optimization tools of the BioUML software (<http://biouml.org>) developed for modeling and analysis of biochemical systems. We created optimization plug-in to solve non-linear optimization problems via minimization of the function of deviations between experimental data and model simulation results. Experimental data can be considered as separate sets of time courses or steady states stored in different tab-separated files. BioUML includes several deterministic and stochastic optimization methods which find reasonably accurate solutions faster than the COPASI software. Some of these methods provide constrained optimization and some of them were parallelized.

## Keywords

BioUML, parameter estimation, multi-experiment parameter fitting

---

## Introduction

Development of experimental technologies in molecular biology led to accumulation of huge volumes of data relating to various levels of life organization. However, the data alone cannot be used to reconstruct the full organization of biological systems. Therefore, the interests of bioinformatics are now focused on the problems of data processing, including the problems of integration and systematization of primary experimental data and the problems of knowledge production based on mathematics and modern information technologies. The challenge of systems biology is construction of mathematical models to describe dynamic behavior of biological systems based on experimental data. Such problems involve studying a large volume of data and require software for their processing and interpreting.

The standard tools for working with biological data include access to biological databases, formalized description of biological systems, as well as visualization, simulation, parameter fitting and analysis of ODE models representing these systems. The BioUML software is an integrated environment that was developed to span all of these capabilities. Here we present optimization tools of this software intended for multi-experiment training of the models created using BioUML notation or imported in the SBML format [Hucka et al, 2003]. These tools are available both in the desktop and web editions of BioUML.

## Optimization problem in BioUML

The general nonlinear optimization problem [Runarsson and Yao, 2000] can be formulated as follows: find a minimum of the objective function  $\phi(x)$ , where  $x$  lies in the intersection of the  $N$ -dimensional search space

$$\Omega = \left\{ y \in \mathbb{R}^N \mid \underline{y}_i \leq y_i \leq \bar{y}_i, \quad \underline{y}_i, \bar{y}_i \in \mathbb{R}, \quad i = 1, \dots, N \right\},$$

and the admissible region  $\mathcal{F} \subseteq \mathbb{R}^N$  defined by a set of equality and/or inequality constraints on  $x$ . Since the equality  $g_s(x) = 0$  can be replaced by two inequalities  $g_s(x) \leq 0$  and  $-g_s(x) \leq 0$ , the admissible region can be defined without loss of generality as

$$\mathcal{F} = \{y \in \mathbb{R}^N \mid g_s(y) \leq 0, \quad s = 1, \dots, p\}.$$

In order to get solution situated inside  $\mathcal{F}$ , we minimize the penalty function

$$\psi(x) = \sum_{i=1}^s \max\{0, g_s(x)\}^2.$$

The problem could be solved by different optimization methods. We implemented the following of them in the BioUML software:

- stochastic ranking evolution strategy (SRES) [Runarsson and Yao, 2000];
- cellular genetic algorithm MOCcell [Nebro et al, 2009];
- particle swarm optimization (PSO) [Sierra and Coello, 2005];
- deterministic method of global optimization glbSolve [Björkman and Holmström, 1999];
- adaptive simulated annealing (ASA) [Ingber, 1996].

Table 1 shows the generic scheme of the optimization process for these methods. SRES, MOCcell, PSO and glbSolve run a predefined number of iterations  $N_{it}$  considering a sequence of sets (populations)  $P^i$ ,  $i = 0, \dots, N_{it} - 1$ , of potential solutions (guesses). In the case of the first three methods, the size  $s \in \mathbb{N}^+$  of the population is fixed, whereas in glbSolve the initial population  $P^0$  consists of one guess, while the size  $s_{k+1}$  of the population  $P^{k+1}$  is found during the iteration with the number  $k = 0, \dots, N_{it} - 1$ . The method ASA considers sequentially generated guesses  $x^k \in \Omega$ ,  $k \in \mathbb{N}^+$ , and stops if distance between  $x^k$  and  $x^{k+1}$  defined as Euclidean norm  $x^k - x^{k+1} = \sqrt{\sum_{i=1}^N (x_i^k - x_i^{k+1})^2}$  becomes less than a predefined accuracy  $\varepsilon$ .

Step	SRES, MOCcell, PSO	glbSolve	ASA
1	Set $k = 0$ .		
2	Generate $P^0 = \{x_i^0 \in \Omega, i = 1, \dots, s\}$ , $s \in \mathbb{N}^+$ . Find the best guess $y \in P^0$ . Set $x^{\min} = y$ .	Generate $P^0 = \{x^0 \in \Omega\}$ . Set $x^{\min} = x^0$ .	Generate $x^0 \in \Omega$ . Set $x^{\min} = x^0$ , $\text{err} = +\infty$ .
3	Evaluate values of the functions $\phi$ and $\psi$ for all guesses $P^0$ .		Evaluate $\phi(x^0)$ and $\psi(x^0)$ .
4	If a predefined number of iterations $N_{it}$ is passed, then go to step 9, otherwise go to step 5.		If $\text{err} < \varepsilon$ , where $\varepsilon$ is a predefined accuracy, then go to step 9, otherwise go to step 6.
5	Set $s_{k+1} = s$ .	Find $s_{k+1}$ for the current iteration.	–
6	Generate $P^{k+1} = \{x_i^{k+1} \in \Omega, i = 1, \dots, s_{k+1}\}$ .		Generate $x^{k+1} \in \Omega$ .
7	Evaluate values of the functions $\phi$ and $\psi$ for all guesses $P^{k+1}$ .		Evaluate $\phi(x^{k+1})$ and $\psi(x^{k+1})$ . Set $\text{err} = x^k - x^{k+1}$ .
8	Update $x^{\min}$ . Increment value $k$ by one, go back to step 4.		
9	Return $x^{\min}$ as the solution.		

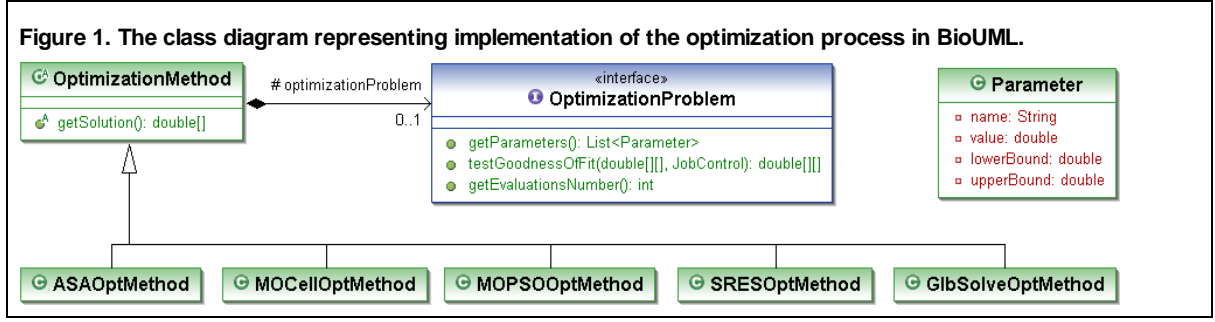
All methods, excepting glbSolve, are stochastic and seek global minimum of the function  $\phi$  taking into account the admissible region  $\mathcal{F}$ . Thus, a guess  $x \in \Omega$  is more preferable than a guess  $y \in \Omega$  at some iteration of methods, if  $\psi(x) = 0$  and  $\psi(y) \neq 0$  or  $\psi(x) < \psi(y)$ . The method glbSolve is suited to solve only the problems with  $\Omega \subseteq \mathcal{F}$ . Values of the function  $\psi$  are calculated but do not affect on the generation of potential solutions.

Implementing the optimization scheme in BioUML, we designed the *OptimizationProblem* interface (fig. 1) comprising the following procedures:

- *getParameters* specifies a list of parameters to fit including identification of initial values and variation intervals (upper and lower bounds);
- *testGoodnessOfFit* defines type of the functions  $\phi$  and  $\psi$  and evaluates their values for a population of guesses;
- *getEvaluationsNumber* returns the number of passed evaluations during optimization process.

An abstract class *OptimizationMethod* provides the number of subclasses representing implementation of the foregoing methods. These subclasses involve search of optimal parameters by calling a procedure *getSolution* depending on the settings of optimization problem.

Figure 1. The class diagram representing implementation of the optimization process in BioUML.



## Application of non-linear optimization to systems biology

We assume that a mathematical model of some biological process consists of a set of chemical species  $S = \{S_1, \dots, S_m\}$  associated with variables  $C(t) = (C_1(t), \dots, C_m(t))$  representing their concentrations, and a set of biochemical reactions  $\mathcal{R} = \{R_1, \dots, R_n\}$  with rates  $v(t) = (v_1(t), \dots, v_n(t))$  depending on a set of kinetic constants  $K$ . Reaction rates are modeled by standard laws of chemical kinetics. A Cauchy problem for ordinary differential equations representing a linear combination of reaction rates is used to describe the model behavior over time:

$$\frac{dC(t)}{dt} = N \cdot v(C, K, t), C(0) = C^0. \quad (1)$$

Here  $N$  is a stoichiometric matrix of  $n$  by  $m$ . We say that  $C^{SS}$  is a steady state of the system (1) if

$$N \cdot v(C^{SS}, K, t) = 0, \lim_{t \rightarrow \infty} C_i(t) = C_i^{SS}.$$

Identification of parameters  $K$  and initial concentrations  $C^0$  is based on experimental data represented by a set of points  $C_i^{exp}(t_{ij})$  defining dynamics of variables  $C_1(t), \dots, C_l(t)$ ,  $l \leq m$ , at given times  $t_{ij}$ ,  $j = 1, \dots, r_i$ , where  $r_i$  is the number of such points for the concentration  $C_i(t)$ ,  $i = 1, \dots, l$ . The problem of parameter identification consists in minimization of the function of deviations defined as the normalized sum of squares [Hoops et al, 2006]:

$$\phi(C^0, K) = \sum_{i=1}^l \sum_{j=1}^{r_i} \frac{\omega_{\min}}{\omega_i} \cdot (C_i(t_{ij}) - C_i^{exp}(t_{ij}))^2, \quad (2)$$

where normalization factors  $\omega_{\min}/\omega_i$  with  $\omega_{\min} = \min_i \omega_i$  are used to make all concentration trajectories have similar importance. The weights  $\omega_i$  are calculated by one of the formulas on experimentally measured concentrations:  $\omega_i^{sq} = \sqrt{r_i^{-1} \cdot \sum_j (C_i^{exp}(t_{ij}))^2}$  (mean square value),  $\omega_i^{mean} = \left| r_i^{-1} \cdot \sum_j C_i^{exp}(t_{ij}) \right|$  (mean value) and  $\omega_i^{st} = \sqrt{\omega_i^{sq} \cdot \omega_i^{sq} - \omega_i^{mean} \cdot \omega_i^{mean}}$  (standard deviation).

If we want to consider additional constrains

$$g_s(C, K) \leq 0, s = 1, \dots, p, \quad (3)$$

holding for concentrations  $C(t)$  and parameters  $K$  for some period of time  $t \in [t_s^{start}, t_s^{end}]$ , the penalty function is defined as

$$\psi(C^0, K) = \sum_{s=1}^p \left( \frac{\sum_{t=t_s^0}^{t_s^{end}} \max\{0, g_s(C, K)\}}{t_s^{end} - t_s^{start} + 1} \right)^2. \quad (4)$$

This function assumes summation of values  $g_s(C, K)$  in the nodes of grid defined by an ODE solver to find a numerical solution of the system (1).

In the particular case, experimental data could be represented by steady state values of species concentrations. Then functions  $\phi$  and  $\psi$  have the simpler forms:

$$\phi(C^0, K) = \sum_{i=1}^l \frac{\omega_{\min}}{\omega_i} \cdot (C_i^{ss} - C_i^{exp-ss})^2, \psi(C^0, K) = \sum_{s=1}^p (\max\{0, g_s(C^{ss}, K)\})^2$$

where  $C_i^{exp-ss}$  and  $C_i^{ss}$ ,  $i = 1, \dots, l$ , denote experimental and simulated steady state values.

Typically, researchers want to perform evaluation of model parameters using experimental data obtained with different experimental conditions, i.e. different initial concentrations  $C^{01}, \dots, C^{0k}$  of species. In such case, we will consider the functions

$$\phi(C^{01}, \dots, C^{0k}, K) = \sum_{i=1}^k \phi(C^{0i}, K) \text{ and } \psi(C^{01}, \dots, C^{0k}, K) = \sum_{i=1}^k \psi(C^{0i}, K). \quad (5)$$

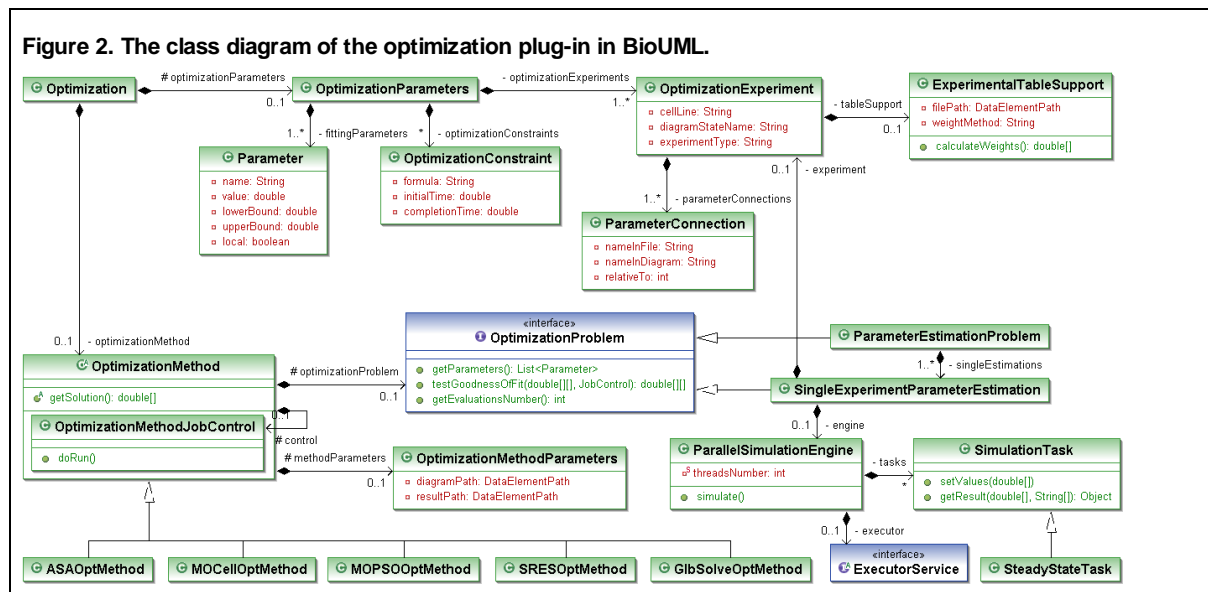
# Implementation of the parameter estimation process in BioUML

Initiation of the parameter estimation process requires definition of many details including specification of the search space, the admissible region, settings of numerical methods to solve ODE system and optimization problem, links to the files with experimental data and model description, etc. In order to structure this information, we designed an appropriate hierarchy of classes (fig. 2) taking into account the following rules:

- experimental data must be represented by time-courses or steady states of chemical species concentrations; in the first case, these data may be expressed as percentage values obtained, for example, on the basis of the Western-blot technology;
- an initial state of the Cauchy problem must be specified for each considered file with experimental data (these states may be different for some experiments);
- parameters to fit may be divided into local and global:
  - the local parameters can take different values for some groups of experiments (for example, conducted for different cell lines);
  - the global parameters have the same value for all experiments.

The main class *Optimization* in fig. 2 comprises definition of the optimization method and parameters including the model parameters to fit, parametric constraints holding for given time intervals, and experiments. The following fields keep information about each experiment:

- *cellLine* defines an experimental group to evaluate local parameters of the model;
- *diagramStateName* corresponds to the model initial state;
- *experimentType* defines the time-course or steady state type of experimental data;
- *tableSupport* contains the link to the file with experimental data, and specifies a method for calculation of weights  $\omega_i$  in the formula (2);
- *parameterConnections* is a list of correspondences between variable names in the model and in the experimental file including information whether data in this file are expressed as exact values or as values related to a given time point.



An object of the *OptimizationMethod* class contains a set of parameters (*methodParameters*) including links to the file with the processed model (*diagramPath*) and the directory to save results when the parameter estimation will be completed (*resultPath*). The estimation process begins immediately after the object *control* gets command *doRun*. Goodness of the current fit is defined through the object *optimizationProblem* providing correct simulation of the model. Firstly, evaluation of functions (2) and (4) is performed separately for all initial states of the model by calling a method *testGoodnessOfFit* for each object of the class *SingleExperimentParameterEstimation* associated with the certain experiment. Then the total values of these functions are calculated in the class *ParameterEstimationProblem* by the formulas (5).

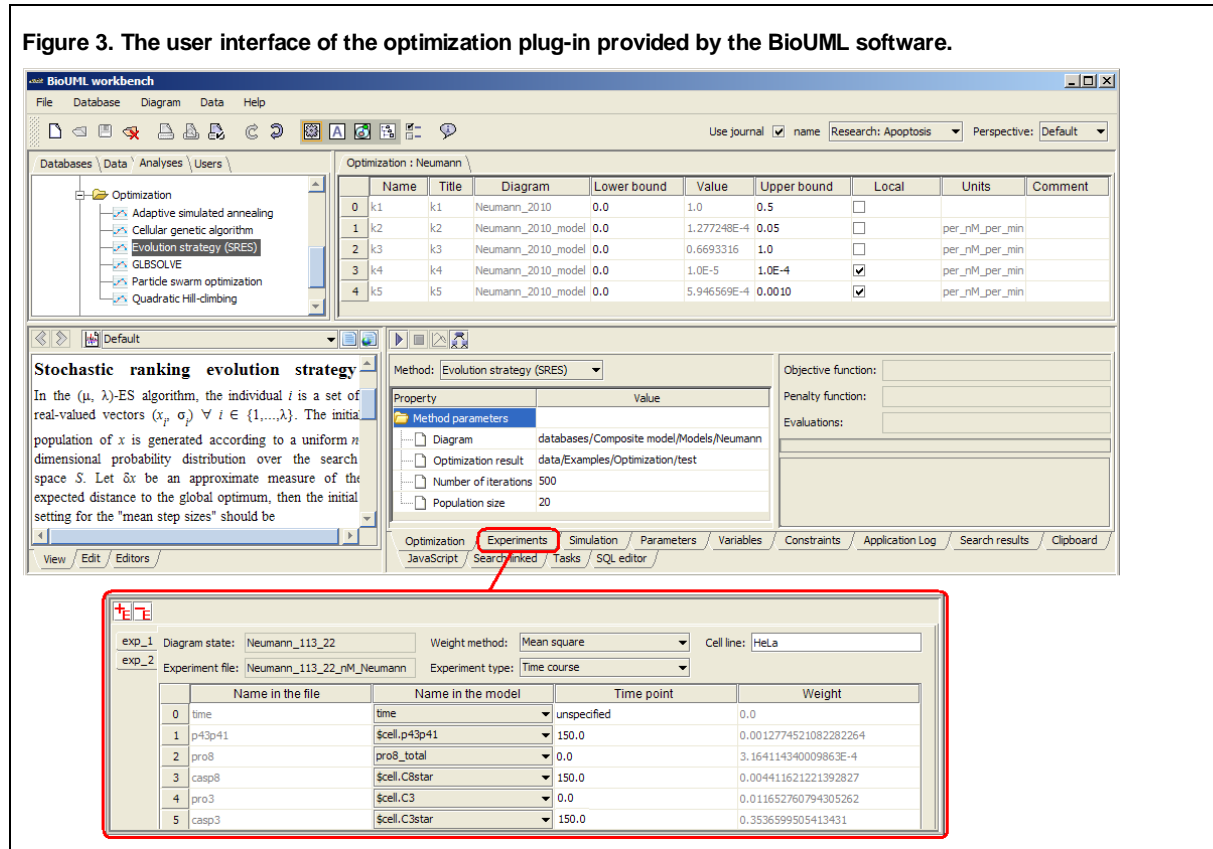
The parameter estimation process is optimized using the following technologies:

1. Acceleration of simulation of the Cauchy problem for different values of fitted parameter is achieved by automatic generation and compilation of the Java class file at the first iteration of the optimization algorithm. At the subsequent iterations, current values are passed to the object of this compiled class and a solution of the Cauchy problem is found.
2. Acceleration of the optimization methods considering a population of guesses is achieved by parallelization of calculations. The following task (*SimulationTask* in fig. 2) is generated for each guess:

- find a solution or a steady state of the system (1) for the adjustable parameter values;
- evaluate values of the objective function (2) and the penalty function (4)

When all tasks are generated, they are passed to the executor service, which distributes their performance between the predefined threads.

Fig. 3 shows a graphical user interface of optimization methods in the BioUML workbench (desktop edition). The upper left panel includes a list of methods. Description of the selected method is provided below. The upper right panel defines the search space. Under it you can find the tab panel with settings of optimization problem. In fig. 3 the selected tab contains method parameters and fields displaying intermediate values of objective/penalty functions and the number of passed evaluations. The next tab includes description of experimental data and specifies settings for all experiment fields listed above.



## Web edition optimization

BioUML web edition (<http://ie.biouml.org/bioumlweb/>) is a web application providing access to BioUML tools and data via the Internet. The user can manipulate the data stored on the BioUML server and run analyses through the web browser. The web interface is a set of HTML pages with interactive JavaScript content. Ajax technology is used to communicate with the server and process user activities without page reloading.

The web edition provides a set of optimization tools, like in the workbench edition. The user can set up the boundaries and initial values of optimization parameters, select and adjust an optimization method, manage experimental data, set up constraints. After all parameters are adjusted, optimization process can be launched. Optimization results can be saved and then viewed as a table of fitted parameter values. Graphical representation of the optimization process is also available.

## Analysis of the methods convergence

Stochastic methods (SRES, MOCcell, PSO and ASA) for global optimization rely on probabilistic approaches and have weak theoretical guarantees of convergence to the global optimum. However, they can locate its vicinity with relative efficiency [Moles et al, 2003]. In contrast, deterministic method glbSolve guarantees global optimality, if the objective function is continuous or at least continuous in the neighborhood of a global optimum [Björkman and Holmström, 1999]. However, it can not solve general global optimization problems with certainty in finite time.

To analyze convergence rate of the implemented methods, we considered a reaction chain (fig. 4, table 2) extracted from the model by L. Neumann *et al.* [Neumann et al, 2010] and representing activation of caspase-8 triggered by the receptor CD95 (APO-1/Fas).

Figure 4. The test model of caspase-8 activation.

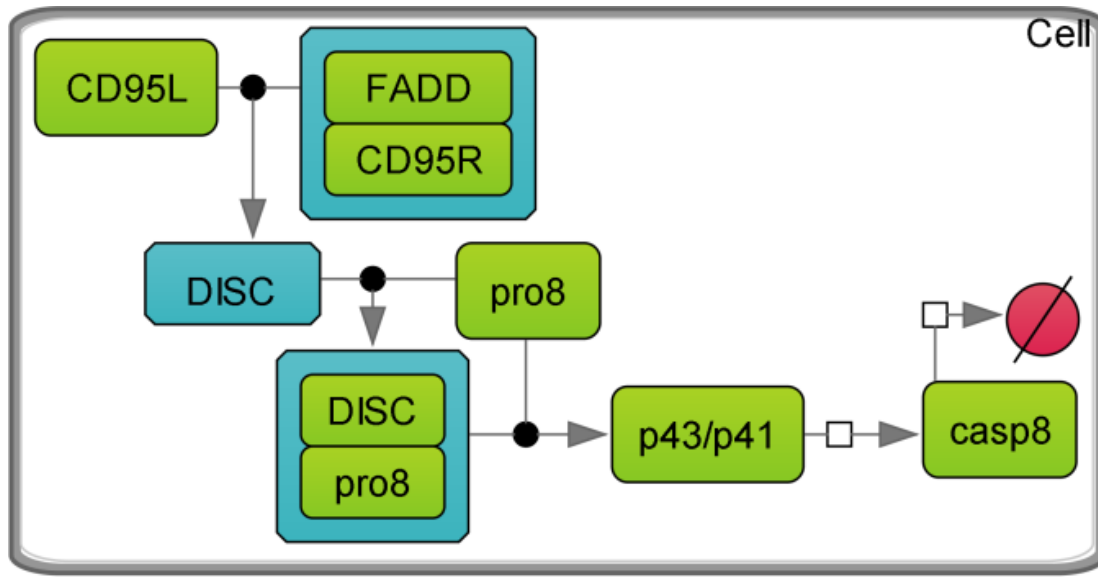


Table 2. List of reactions of caspase-8 activation.

N <sub>o</sub>	Reactions	Kinetic laws
r1	CD95L + FADD:CD95R → DISC	$k_1 \cdot C_{CD95L} \cdot C_{CD95R:FADD}$
r2	DISC + pro8 → DISC:pro8	$k_2 \cdot C_{DISC} \cdot C_{pro8}$
r3	DISC:pro8 + pro8 → 2 · p43/p41	$k_3 \cdot C_{DISC:pro8} \cdot C_{pro8}$
r4	2 · p43/p41 → casp8	$k_4 \cdot C_{p43/p41}^2$
r5	casp8 →	$k_5 \cdot C_{casp8}$

We performed estimation of parameters using the search space defined as

$$0 \leq k_1 \leq 1, 0 \leq k_5 \leq 0.1, 0 \leq k_i \leq 10^{-3}, i = 2, 3, 4,$$

where upper bounds were chosen based on the order of magnitude of parameter values proposed by authors of the original model. Initial values of variables were fixed according to [Neumann et al, 2010]:

$$C_{CD95L}(0) = 113.220, C_{CD95R:FADD}(0) = 91.266, C_{pro8}(0) = 64.477.$$

Estimation was based on the experimental data obtained by Neumann *et al.* for procaspase-8 and its cleaved products p43/p41 and caspase-8 (table 3).

Table 3. Experimental data obtained by Neumann *et al.* for total procaspase-8 (pro-8), p43/p41 and caspase-8 (casp-8).

Time (min <sup>-1</sup> )	Concentrations (nM)		
	p43/p41	pro-8	casp-8
0.0	0.058	59.963	0.000
10.0	0.268	57.565	0.041
20.0	4.760	58.590	0.316
30.0	8.252	59.422	1.397
45.0	16.144	48.190	3.520
60.0	17.021	38.950	3.947
90.0	15.269	23.502	4.871
120.0	12.530	13.127	4.878
150.0	10.335	10.703	4.228

Fig. 5 shows dependence of the objective function mean values on the number of considered guesses for 100 runs of the optimization process. Statistics of the best, mean and worst values of  $\phi$ , as well as the best guesses found by the methods after consideration of  $10^7$  guesses are listed in the tables 4 and 5 correspondently.

As can be seen from these tables, the best result was obtained by the particle swarm optimization (PSO) and the cellular genetic algorithm (MOCeLL). Methods SRES, MOCeLL and PSO found similar solutions. Methods ASA and glbSolve found dissimilar values of parameters  $k_1$  and  $k_2$  resulting in lower efficiency compared to the first three methods.

For comparison, some test cases considered in the study by Moles *et al.* [Moles et al, 2003] resulted in superiority of SRES. However, the authors did not explore such methods as genetic algorithms and particle swarm optimization.

Figure 5. Dynamics of mean values of the objective function for 100 runs of the optimization process. The best value obtained by the particle swarm optimization is marked by the red line.

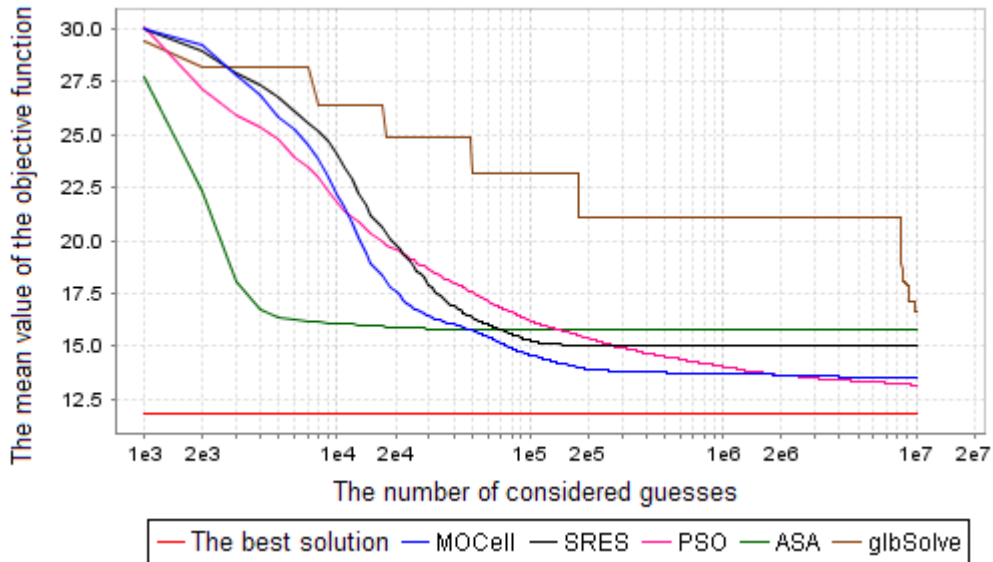


Table 4. Statistics of values of the function  $\phi$  for 100 runs of the optimization process (the number of considered guesses was  $10^7$ ).

Methods	The best value of $\phi$	The mean value of $\phi$	The worst value of $\phi$
PSO	11.787	13.164	14.703
MOCcell	12.082	13.484	14.771
SRES	12.466	14.987	18.283
ASA	13.728	15.794	16.610
glbSolve	16.614	16.614	16.614

Table 5. The best guesses obtained by optimization methods for 100 runs of the optimization process.

Parameters	SRES	MOCcell	PSO	ASA	glbSolve
$k_1$	0.0004691	0.0004611	0.0004277	0.0001028	0.0020576
$k_2$	0.0002059	0.0002046	0.0002155	0.0007875	0.0001228
$k_3$	0.0009999	0.0010000	0.0009984	0.0009930	0.0009527
$k_4$	0.0007915	0.0008225	0.0008419	0.0008117	0.0007790
$k_5$	0.0325900	0.0336720	0.0334167	0.0334118	0.0313443

## Comparison of parameter estimation features of BioUML with other software

We compared optimization tools of BioUML with the following software applications assigned for analysis of biochemical networks and supporting the procedure of model fitting:

- COPASI (Complex Pathway Simulator) – a stand-alone program providing an C++ API [Hoops et al, 2006];
- AMIGO (Advanced Model Identification using Global Optimization) – a multi-platform (Windows and Linux) MATLAB-based toolbox [Balsa-Canto and Banga, 2011];
- SBToolbox 2 (Systems Biology Toolbox 2) – a part of SBPOP Package requiring MATLAB [Schmidt and Jirstrand, 2005];
- PET (Parameter Estimation Toolkit) – a graphical user interface intended to run under Windows, Mac OS X, and Unix [Shaffer et al, 2009];
- PottersWheel – a framework designed as a MATLAB toolbox [Maiwald and Timmer, 2008].

Details of the comparison are given in table 6.

**Table 6. Comparison of the parameter estimation features for different software applications.**

Features	BioUML	COPASI	AMIGO	SBToolbox 2	PET	PottersWheel
Environment	Java	C++	MATLAB	MATLAB	Perl, Gtk+	MATLAB
Experimental data:						
– Multi-experiment fitting	+	+	+	+	+	+
– Experiment types:						
– time course	+	+	+	+	+	+
– steady-state	+	+	–	–	+	–
– Individual initial state of the model for each experiment	+	–	+	+	+	+
– Error bars	–	–	+	+	–	+
– Normalization of data using weights	+	+	+	+	+	+
Local (experiment dependent) and global parameters	+	–	+	+	–	–

Further, we compared computation speed of the optimization methods implemented in BioUML and COPASI. For this purpose, we considered a series of test cases. A brief description of the models used in these test cases is provided below. For more details, including specification of experimental data and fitting parameters, see Additional file 1 of the supplementary materials.

In the first test case, we analyzed three models of CD95-induced caspase-8 activation constructed on the basis of the model by Neumann *et al.* [Neumann *et al.*, 2010] with varying degrees of detail (fig. 6, A-C). The second test case was proposed by Mendes *et al.* [Mendes *et al.*, 2009] and corresponded to the model of the MAP kinase cascade (fig. 7) developed by Kholodenko *et al.* [Kholodenko, 2000]. Finally, we tested the model of Bageci *et al.* [Bageci *et al.*, 2006] representing the mitochondria-dependent apoptosis resulting from the cooperative formation of heptameric apoptosome complex and activation of caspase-9 and caspase-3 (fig. 8).

**Figure 6. The models of caspase-8 activation constructed based on the model by Neumann *et al.* with the varying number of species: 7 (A), 13 (B), and 18 (C).**

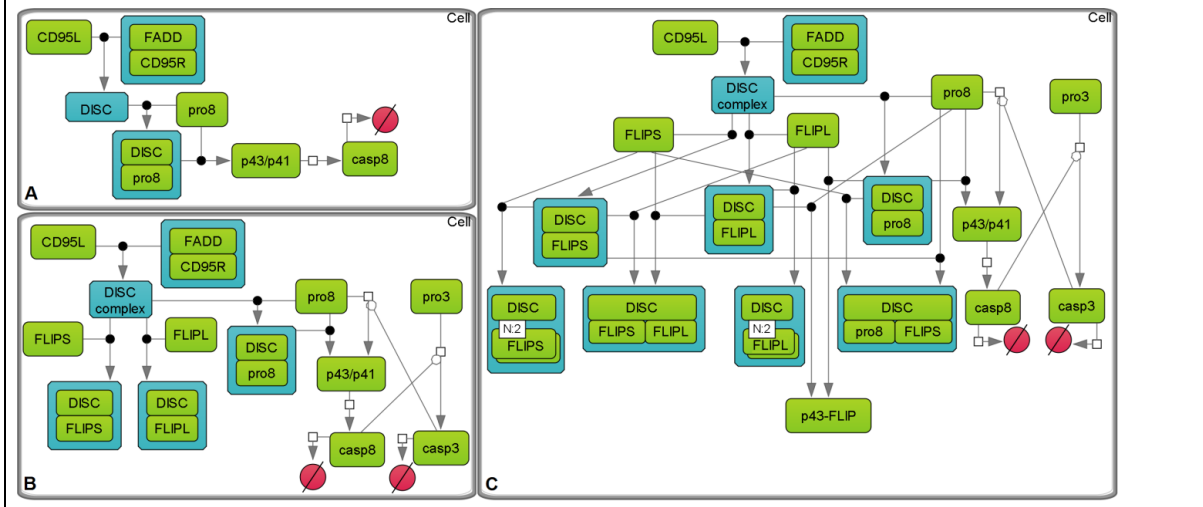




Figure 7. The model of the MAP kinase cascade constructed by Kholodenko *et al.*

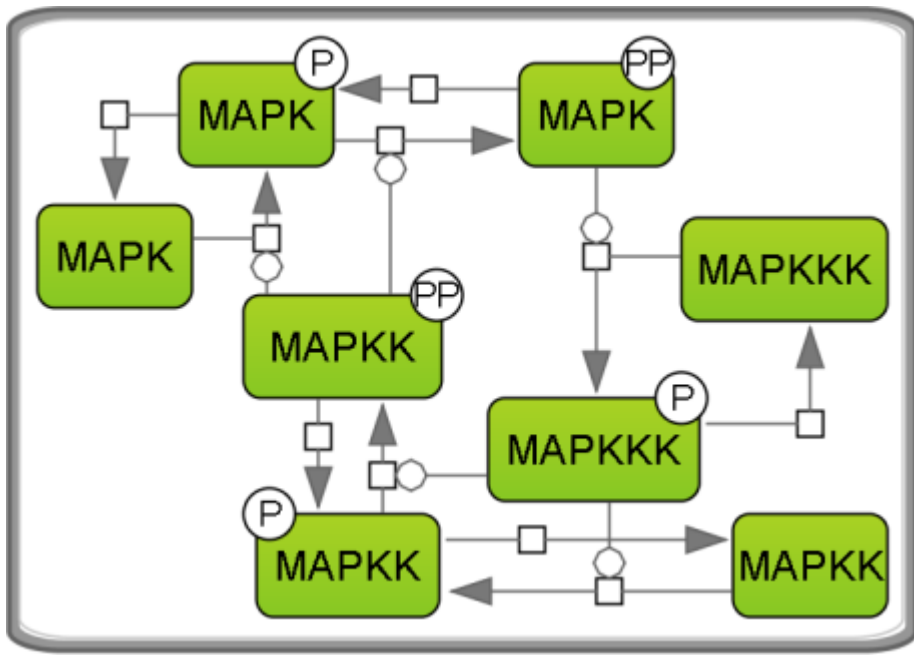
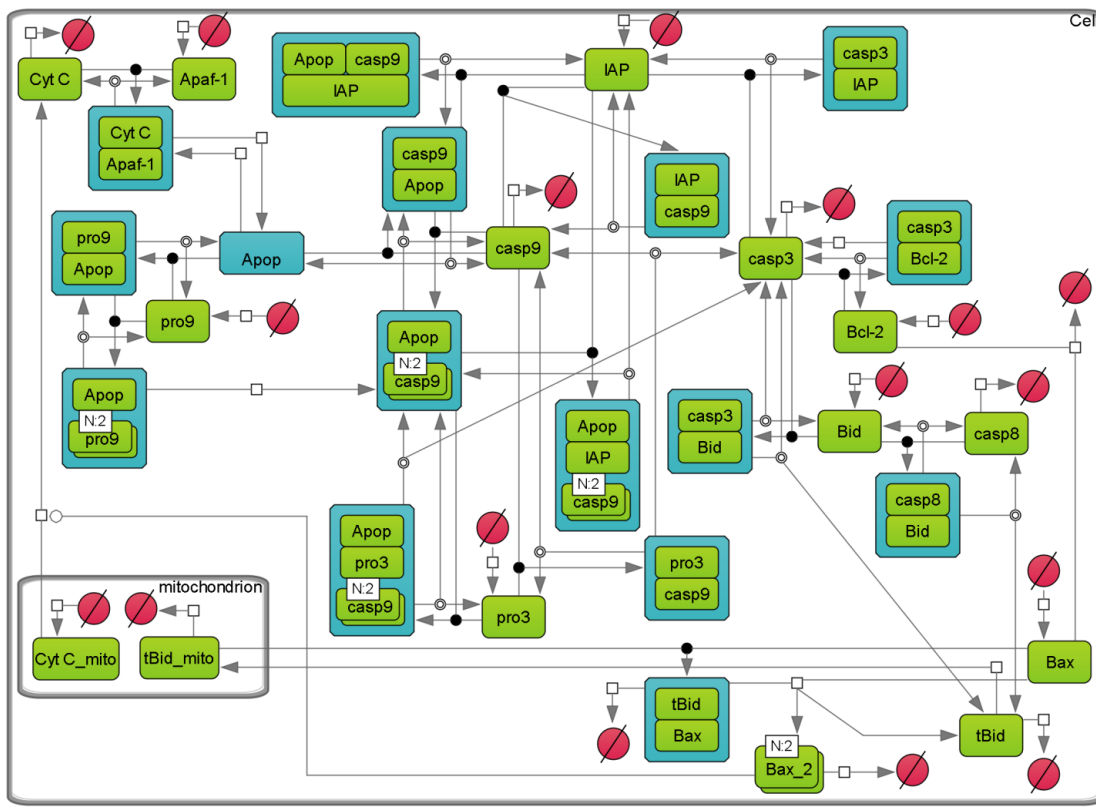
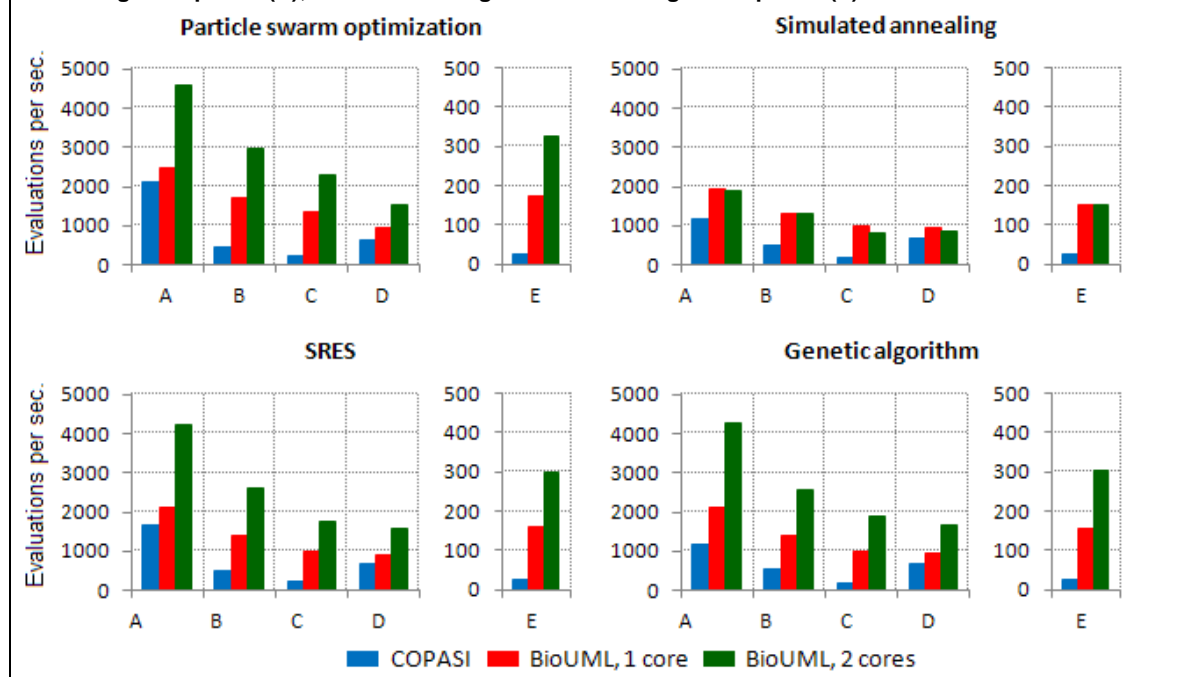


Figure 8. The model of the mitochondria-dependent apoptosis proposed by Bagci *et al.*



Analyzing the number of the objective function evaluations per second for these test cases, we found that BioUML showed a better result than COPASI (fig. 9).

Figure 9. The number of the objective function evaluations per second for different test cases in COPASI and BioUML: the model by Neumann *et al.* including 7 (A), 13 (B), and 18 (C) species; the model by Kholodenko *et al.* consisting of 8 species (D); the model of Bagci *et al.* consisting of 32 species (E).



## Discussion

In this paper, we considered parameter estimation tools of the BioUML software. These tools can be applied to biological systems characterized with a set of ODEs. The fitting process is based on experimental time course or steady state measurements, and assumes minimization of the function of error between these measurements and the corresponding model prediction. We implemented several stochastic and deterministic global optimization methods as new plug-in for BioUML. None of these methods is effective for all cases. Nevertheless, on the basis of our observations, we concluded that adaptive simulated annealing can be used when it is necessary to quickly find the vicinity of the solution. In the case when adequacy of solution is more preferable than the rate of convergence, it is better to use such methods as MOCcell, PSO and SRES.

Parameter fitting is an important part of the quantitative biological modeling. However, if the model includes more elements than are necessary to approximate experimental data with the given accuracy, we face the problem of overfitting [Hawkins, 2004]. In this case, there is no overall best solution and it is expedient to find distribution of the parameter values which are compatible with observed experimental dynamics. For this purpose, we should run parameter estimation process many times (it is better with different optimization methods) and evaluate bounds for all parameters. Implementation of this technique in BioUML is a task for the future work.

We successfully applied our optimization plug-in for creation of the combined model of CD95 and NF- $\kappa$ B signaling pathways [Kutumova et al, 2013], where the problem of parameter overfitting was solved using the methodology of model reduction [Gorban et al, 2010].

## References

- Bagci E Z, Vodovotz Y, Billiar T R, Ermentrout G B, Bahar I, authors. **Bistability in apoptosis: roles of bax, bcl-2, and mitochondrial permeability transition pores.** *Biophys J.* 1–3;2006;(5)90:1546–1559. DOI:10.1529/biophysj.105.068122 [PMID:16339882]
- Balsa-Canto Eva, Banga Julio R, authors. **AMIGO, a toolbox for advanced model identification in systems biology using global optimization.** *Bioinformatics.* 17–6;2011;(16)27:2311–2313. DOI:10.1093/bioinformatics/btr370 [PMID:21685047]
- Björkman M, Holmström K, authors. **Global Optimization Using the DIRECT Algorithm in Matlab.** *Advanced Modeling and Optimization.* 1999;(2)1:17–37
- Brown Peter N, Byrne George D, Hindmarsh Alan C, authors. **VODE: A Variable-Coefficient ODE Solver.** *SIAM J. Sci. and Stat. Comput.* 1989;(5)10:1038–1051. ISSN: 0196-5204 DOI:10.1137/0910062
- Dormand J R, Prince P J, authors. **A family of embedded Runge-Kutta formulae.** *Journal of Computational and Applied Mathematics.* 1980;(1)6:19–26. ISSN: 03770427 DOI:10.1016/0771-050X(80)90013-3
- Gorban A N, Radulescu O, Zinovyev A Y, authors. **Asymptotology of chemical reaction networks.** *Chemical Engineering Science.* 2010;(7)65:2310–2324. ISSN: 00092509 DOI:10.1016/j.ces.2009.09.005
- Hairer E, Wanner G, authors. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems.* 1996.

- 2nd revised. Berlin: Springer;
- Hawkins Douglas M, author. **The problem of overfitting**. *J Chem Inf Comput Sci*. 2004;(1)44:1–12. DOI:10.1021/ci0342472 [PMID:14741005]
- Hoops Stefan, Sahle Sven, Gauges Ralph, Lee Christine, Pahle Jürgen, Simus Natalia, Singhal Mudita, Xu Liang, Mendes Pedro, Kummer Ursula, authors. **COPASI--a COMplex PATHway SIMulator**. *Bioinformatics*. 10–10;2006; (24)22:3067–3074. DOI:10.1093/bioinformatics/btl485 [PMID:17032683]
- Hucka M, Finney A, Sauro H M, Bolouri H, Doyle J C, Kitano H, Arkin A P, Bornstein B J, Bray D, Cornish-Bowden A, Cuellar A A, Dronov S, Gilles E D, Ginkel M, Gor V, Goryanin I I, Hedley W J, Hodgman T C, Hofmeyr J-H, Hunter P J, Juty N S, Kasberger J L, Kremling A, Kummer U, Le Novère N, Loew L M, Lucio D, Mendes P, Minch E, Mjolsness E D, Nakayama Y, Nelson M R, Nielsen P F, Sakurada T, Schaff J C, Shapiro B E, Shimizu T S, Spence H D, Stelling J, Takahashi K, Tomita M, Wagner J, Wang J, authors. **The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models**. *Bioinformatics*. 1–3;2003; (4)19:524–531. [PMID:12611808]
- Ingber L, author. **Adaptive simulated annealing (ASA): Lessons learned**. *Control and Cybernetics*. 1996;(1)25:33–54
- Kholodenko B N, author. **Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades**. *Eur J Biochem*. 2000;(6)267:1583–1588. [PMID:10712587]
- Kutumova Elena, Zinovyev Andrei, Sharipov Ruslan, Kolpakov Fedor, authors. **Model composition through model reduction: a combined model of CD95 and NF- $\kappa$ B signaling pathways**. *BMC Syst Biol*. 15–2;2013;7:13 DOI:10.1186/1752-0509-7-13 [PMID:23409788]
- Maiwald Thomas, Timmer Jens, authors. **Dynamical modeling and multi-experiment fitting with PottersWheel**. *Bioinformatics*. 9–7;2008;(18)24:2037–2043. DOI:10.1093/bioinformatics/btn350 [PMID:18614583]
- Mendes Pedro, Hoops Stefan, Sahle Sven, Gauges Ralph, Dada Joseph, Kummer Ursula, authors. **Computational modeling of biochemical networks using COPASI**. *Methods Mol Biol*. 2009;500:17–59. DOI:10.1007/978-1-59745-525-1\_2 [PMID:19399433]
- Moles Carmen G, Mendes Pedro, Banga Julio R, authors. **Parameter estimation in biochemical pathways: a comparison of global optimization methods**. *Genome Res*. 14–10;2003;(11)13:2467–2474. DOI:10.1101/gr.1262503 [PMID:14559783]
- Nebro Antonio J, Durillo Juan J, Luna Francisco, Dorronsoro Bernabé, Alba Enrique, authors. **MOCeLL: A cellular genetic algorithm for multiobjective optimization**. *Int. J. Intell. Syst*. 2009;(7)24:726–746. ISSN: 08848173 DOI:10.1002/int.20358
- Neumann Leo, Pforr Carina, Beaudouin Joel, Pappa Alexander, Fricker Nicolai, Krammer Peter H, Lavrik Inna N, Eils Roland, authors. **Dynamics within the CD95 death-inducing signaling complex decide life and death of cells**. *Mol Syst Biol*. 9–3;2010;6:352 DOI:10.1038/msb.2010.6 [PMID:20212524]
- Runarsson T P, Yao Xin, authors. **Stochastic ranking for constrained evolutionary optimization**. *IEEE Trans. Evol. Computat*. 2000;(3)4:284–294. ISSN: 1089778X DOI:10.1109/4235.873238
- Schmidt Henning, Jirstrand Mats, authors. **Systems Biology Toolbox for MATLAB: a computational platform for research in systems biology**. *Bioinformatics*. 29–11;2005;(4)22:514–515. DOI:10.1093/bioinformatics/bti799 [PMID:16317076]
- Shaffer Clifford A, Zwolak Jason W, Randhawa Ranjit, Tyson John J, authors. **Modeling molecular regulatory networks with JigCell and PET**. *Methods Mol Biol*. 2009;500:81–111. DOI:10.1007/978-1-59745-525-1\_4 [PMID:19399431]
- Sierra Margarita R, Coello Carlos A Coello, authors. **Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and  $\epsilon$ -Dominance**. *Evolutionary Multi-Criterion Optimization*. 2005. p. 505–519. Berlin, Heidelberg: Springer Berlin Heidelberg; DOI:10.1007/978-3-540-31880-4\_35