Evolutionary Approaches for Strain Optimization using Dynamic Models under a Metabolic Engineering Perspective

Pedro Evangelista[1,2], Isabel Rocha[2], Eugénio C. Ferreira[2], Miguel Rocha[1]

[1] Departament of Informatics / CCTC - University of Minho
Campus de Gualtar, 4710-057 Braga - PORTUGAL
ptiago@deb.uminho.pt mrocha@di.uminho.pt
[2] IBB - Institute for Biotechnology and Bioengineering
Center of Biological Engineering - University of Minho
Campus de Gualtar, 4710-057 Braga - PORTUGAL
ecferreira@deb.uminho.pt irocha@deb.uminho.pt

Abstract. One of the purposes of Systems Biology is the quantitative modeling of biochemical networks. In this effort, the use of dynamical mathematical models provides for powerful tools in the prediction of the phenotypical behavior of microorganisms under distinct environmental conditions or subject to genetic modifications.
The purpose of the present study is to explore a computational environment where dynamical models are used to support simulation and optimization tasks. These will be used to study the effects of two distinct types of modifications over metabolic models: deleting a few reactions (knockouts) and changing the values of reaction kinetic parameters. In the former case, we aim to reach an optimal knockout set, under a defined objective function. In the latter, the same objective function is used, but the aim is to optimize the values of certain enzymatic kinetic coefficients. In both cases, we seek for the best model modifications that might lead to a desired impact on the concentration of chemical species in a metabolic pathway. This concept was tested by trying to maximize the production of dihydroxyacetone phosphate, using Evolutionary Computation approaches. As a case study, the central carbon metabolism of *Escherichia coli* is considered. A dynamical model based on ordinary differential equations is used to perform the simulations. The results validate the main features of the approach.

1 Introduction

Systems Biology represents a new approach to research in Biology. It aims to achieve the understanding of the complex interactions in biological systems under an integrative approach, where the ultimate goal is to simulate these systems under different scenarios and perturbations [20]. One of the main purposes of this work is to provide tools for the dynamical modeling and optimization of biological processes, under a Metabolic Engineering perspective. Indeed, we aim

to provide tools to identify optimal or near-optimal sets of genetic changes in microorganisms under dynamical conditions to achieve a given industrial aim.

Mathematical dynamical models allow to study the interaction of biological compounds in cells. There are several types of dynamic models [13][2], but the most common approach is to represent metabolic networks as a system of ordinary differential equations (ODEs). One of the major drawbacks with these types of models is how to reliably estimate model parameters. Another pertinent question is how these values can change and the biological meaning of those modifications. In this work, we face the question of how to evolve a dynamical model based on predefined goals.

We take a new approach by using dynamical models and Evolutionary Computation to identify a set of knockouts or variations in some kinetic parameters that will optimize the production of a certain metabolite. Each configuration is evaluated resorting to a simulation using the dynamical model. This type of information can help to assess on how to engineer a metabolic network in order to enhance the production of a given metabolite and can also be used to infer regulatory data.

It is important to bear in mind that finding a knockout set can be seen as a change in the model structure and the corresponding problem belongs to the class of combinatorial optimization. On the other hand, the second task involves finding the best values for a number of parameters, thus a numerical optimization task. Thus, although the two tasks are quite different from the point of view of optimization, a similar and general purpose strategy will be followed in both cases.

Indeed, to study the described scenarios the concept of dynamical model evolution is introduced. In our proposal, a model will evolve based on a fitness function that is defined considering a given industrial aim (e.g. to maximize the concentration of a given metabolite along the time of the experiment). In alternative, although this is not shown in this work, the fitness can also be based on an error function, if some experimental data to fit is available.

An optimization framework was built around this concept, where the major design concern was the loose coupling between the optimization and the simulation modules. This allows us to optimize any model component independently of the optimization algorithm and of the simulation method.

The framework was applied in this work to the dynamical model of the central carbon metabolism of *Escherichia coli* [2]. This model links the sugar transport system with the reactions of glycolysis and pentose-phosphate pathway. The case study was chosen because it includes most of the reactions of the central carbon metabolism and has been validated experimentally. Moreover, *E. coli* has been the organism of choice to test novel Metabolic Engineering tools, given the simplicity in performing genetic modifications, among other factors.

The optimization of knockout sets to enhance the production of metabolites has been approached before in literature [14][16]. These studies focused in finding a knockout set using stoichiometric models, performing the simulations using steady-state approaches such as Flux Balance Analysis [4]. Rather less attention

has been paid to optimizing a knockout set based on dynamical models, using as a fitness function the concentration of a certain metabolite in a defined time interval.

Several methods have also been proposed to estimate parameter values based on experimental data [12]: Wang [19] applied an extra focus in how to use genetic algorithms to optimize model parameters. In [10], the authors describe the use of Evolutionary Algorithms (EAs) to reconstruct a metabolic network using functional Petri Nets. In the work developed by Haunschild [6], the concept of automatic generation/evolution of multiple metabolic models is introduced to try to explain some biochemical network phenomena. However, rate equations are defined by the user and are not evolved themselves. These approaches are orthogonal to the one presented in this paper, since in our problem context there is no need of using experimental data to find parameter values.

The rest of this paper is organized as follows: the description of our framework for dynamical model evolution is given in section 2; in section 3, the case study is presented; afterwards, in section 4 the results are presented and discussed; finally, section 5 provides the conclusions and the future work.

2 A framework for dynamical model evolution

This section describes our general framework for dynamical model evolution. As said before, a dynamical model can evolve based on a given fitness function that can be defined in a flexible way, i.e. no restrictions are imposed over the definition of the fitness function (it can be nonlinear, non differentiable, discontinuous, etc.). It can, for instance, be an error function that takes into account known experimental data and thus the aim will be to estimate the parameters that best fit the data. On the other hand, the fitness function can be based on the concentration of one or several metabolites, along the simulation period. In this last case, the fitness can be measured by the integral (area under the curve) of the objective function.

Our framework is divided into two logical parts: model simulation and optimization (Figure 1). The simulation part is based on the numerical integration of the ODEs of the model, specifying a time interval and considering a fixed model structure with pre-defined values for the model parameters. A set of initial values (e.g. representing environmental constraints) can also be defined by the user for the state variables of the model.

The optimization part allows modifications both in the model structure and in several types of parameters, including kinetic formulas and corresponding parameters. The purpose is to reach model configurations that optimize a given fitness function. A user can impose changes over the model in order to simulate specific cases. Furthermore, optimization algorithms can be defined to search over the space of potential solutions, given the type of allowed changes, that can be summarized in the following:

- Changes in the initial values of the variables (e.g. initial metabolite concentrations);

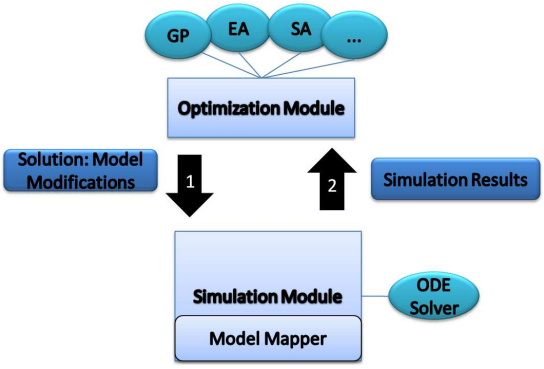

Fig. 1. Framework for dynamical model evolution.

- Changes in the kinetic parameters (e.g. global model parameter or parameters of a specific kinetic expression);
- Changes in the structure of the reaction kinetics (e.g. algebraic expression);
- Changes in the overall model structure (e.g. reaction participant metabolites).

In this paper, the main focus will be on the kinetic parameter variation and model structure, considering the possible deletion of a number of reactions from the model.

In our framework, an optimization process is represented by a model, an optimization algorithm and corresponding parameters, a decoder and an override model, while a simulation is only characterized by a model and the override model component. The algorithm and the parameters represent the optimization method and the variables that will be used during a optimization run.

The model integrates all components that describe the dynamical system (the ODEs, the kinetic laws and parameters, etc.). For this purpose, a unified model representation is built, called *model mapper*, that will answer any queries about the model components (e.g. about the model structure or parameter values). This model view is composed by three layers in the following order (Figure 2): (3) the original model, (2) the decoder and (1) the override model. When a query is made it is passed along the chain of entities (in the order 1,2,3) until one of them can answer the query.

Therefore, the decoder and the override model are fractional model representations. The decoder gives a partial model view based on a specific encoding. This layer is used mainly to provide a way to decode the solutions of possible optimization algorithms from their internal representations, namely decoding the genome of an EA. The override model can be used to redefine a set of model components, thus enabling to set conditions that remain constant throughout the optimization process.

In more detail, a *model* is composed by the following elements:

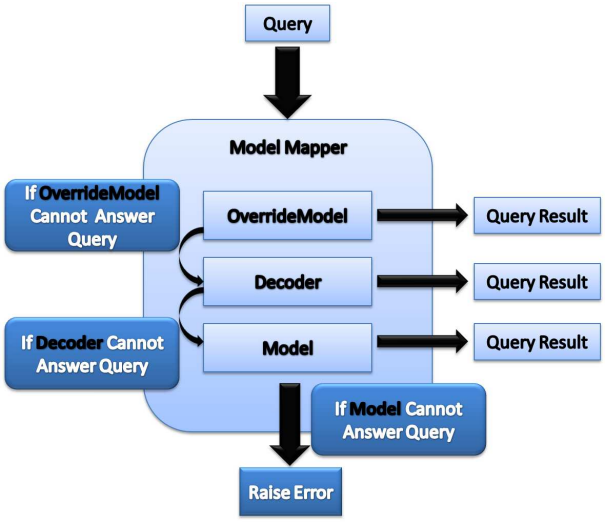

Fig. 2. Framework layers for dynamical model evolution.

- A set of parameters. Each parameter is denoted by a name and has a numerical value.
- A set of variables. A variable is defined by an upper and a lower limit, an initial value and an ODE, that is represented by a sum of terms, where each term has a multiplicative coefficient and a function.
- A set of functions. A function can be any mathematical entity that receives as its parameters the current time and a model representation, returning a numerical result. Functions can also have a local parameter space that overrides the model global parameter scope.

Regarding the optimization layer, several algorithms can be employed, providing they are able to deal with the type of fitness functions described before. Given the complexity of the underlying problems, the available options are meta-heuristics that range from Multistart Local Search to Simulated Annealing, contemplating also several evolutionary approches, such as EAs, Genetic Programming or Differential Evolution (DE). In this work, EAs will be used to perform combinatorial optimization and a DE will perform the numerical optimization task. The specific features of these algorithms will be presented in the next section.

Besides running the simulation and optimization of dynamical models, the framework also allows to input models using the Systems Biology Markup Language (SBML) [7] format (a standard in these kind of models). The results of a simulation or optimization process can also be saved in a text file. A number of visualization tools are also available to allow the user to perform a graphical analysis over the outputs.

Regarding its implementation, the software for the proposed framework was developed using the Java programming language and the following additional libraries: a library for EAs developed by the authors, *CVODE* [3] using JNI that solves systems of ODEs, *JFreeChart* [9] that displays graphical simulation results and *LibSBML* [1] that parses SBML encoded files.

3 Case Study

3.1 Dynamical model of the central carbon metabolism of *E. coli*

In this paper, a case study on the dynamical model of glycolysis and the pentose-phoshate pathway in *Escherichia coli*[2] was used. One of the main ambitions in Metabolic Engineering is the re-engineering of biological pathways with the aid of mathematical models. The model was delineated and corroborated based in metabolite concentration measurements obtained at transient conditions. This model allows to explore this network as supplier of precursors. For example, dihydroxyacetone phosphate (DHAP) can be produced and used in the lipid synthesis pathway. The maximization of the production of this compound was used as case study since it has several industrial applications, including synthetic chemistry using the enzymatic Aldol Syntheses[5][18].

The model of the central carbon metabolism of *Escherichia coli* consists of mass balance equations for extra-cellular glucose and for intracellular metabolites. The mass balances take the following form:

$$\frac{dC_i}{dt} = \sum_j v_{ij} r_j - \mu C_i \tag{1}$$

Where C_i represents the concentration of metabolite i, μ is the specific growth rate and v_{ij} is the stoichiometric coefficient for this metabolite in reaction j, the rate of which is r_j.

3.2 Optimization tasks and algorithms

In this section, the optimization tasks and techniques employed are described. Two distinct scenarios were studied in this work, both using the model aforementioned. In the first, the problem at hand consists in determining the optimal knockout set that maximizes the production of a given metabolite (in this case DHAP) along a given time interval (in this case is was set to [0, 20] seconds). Therefore, the fitness function consists on the numerical integration of the target metabolite's concentration. The integration of the ODE is performed using the method provided by CVODE (that is suitable for both stiff and non-stiff ODE problems) with a step size of 0.1. In the simulations, the initial values for the model variables (i.e. initial concentrations) were set to the values supplied by [2].

In both cases, since the algorithms are stochastic, the optimization process was run for 30 times and the results are the means, presented within a 95% confidence interval.

In the first task, an EA with a set-based representation was used [16], where an individual encodes a subset of the full set of reactions in the model. To evaluate each solution, a simulation is run where the model is changed by removing all reactions encoded in the individual's genome. The fitness function is therefore calculated using this modified model.

It should be mentioned that the representation used in the EA employs a variable-sized genome, therefore allowing the competition of knockout sets with distinct cardinalities within the same population. Within this EA, the following reproduction operators are used to breed new individuals [16]:

- Grow mutation: consists in the introduction of a number of new elements into the set, whose values are randomly generated within the available range, avoiding duplicates;
- Shrink mutation: a number of randomly selected elements are removed from the set;
- Random mutation: replaces an element of the set by another, randomly generated in the allowed range; and,
- Modified Uniform crossover: it is inspired on the traditional uniform crossover operator and works as follows: the genes that are present in both parent sets are kept in both offspring; the genes that are present in only one of the parents are sent to one of the offspring, selected randomly with equal probabilities.

The following steps present the general structure of the EA:

1. Generate a population of NP individuals. Each individual represents a potential solution to the problem, initially created randomly.
2. For each individual in the population, evaluate its fitness by running the correspondent model simulation and computing the fitness function. If the stopping criteria is met, the algorithm stops and returns the best solution found.
3. Selection: First the set of E best individuals is copied to the next generation (elitism). Afterwards, a pool of $NP/2$ individuals (parents) is created using a *roulette wheel* scheme.
4. Reproduction: The set of available reproduction operators (crossover and mutation) are applied to the selected pool of parents, in order to generate the offspring ($NP/2$ new individuals are created that are inserted into the new generation). All reproduction operators available have the same probability of being chosen to breed each new individual.
5. The new population is completed by selecting $NP/2 - E$ individuals from the original population (a substitution rate of 50% is adopted). Return to step 2.

The EA was ran for 500 iterations with a population of 100 individuals. An elitism value of $E = 1$ is used.

In the second scenario, a similar approach was taken, but instead of finding a knockout set, the purpose is to modify the value of one of the kinetic parameters of each reaction, in this case the v_{max}. The v_{max} parameter represents the

maximum enzyme reaction rate under the conditions of the experiment. This value can be changed in a wet lab by changing the level of expression of certain enzymes in the re-engineered microbial strains.

In this second scenario, a DE algorithm was employed. The individuals encode the level of change for the v_{max} parameter of each reaction, when compared to the base value present in the original model. The level of change can vary between 0 and 2; a value of 1 means the parameter remains unchanged.

In this work, a variant of the DE algorithm called *DE/rand/1* was considered that uses a binomial crossover [17]. In this case, the following scheme is followed, in every generation, for each individual i in the population:

1. Randomly select 3 individuals r_1, r_2, r_3 distinct from i;
2. Generate a trial vector based on: $\boldsymbol{t} = \boldsymbol{r}_1 + F \cdot (\boldsymbol{r}_2 - \boldsymbol{r}_3)$;
3. Incorporate coordinates of this vector with probability CR;
4. Evaluate the candidate and use it in the new generation if it is at least as good as the current individual.

The DE was ran for 500 iterations with a population of 20 individuals. The F parameter was set to 0.5 and CR to 0.6.

4 Results

4.1 Gene deletion scenario

In Table 1 we show the results for the gene deletion task. The mean and confidence interval of the fitness function value (DHAP production) obtained for the best solution in each run is shown, as well as the mean number of knockouts. On the other hand, Figure 3 shows the histogram of the reaction knockouts, i.e. the number of times a given reaction is knocked-out in the best solution for the 30 runs.

Table 1. Results for the gene deletion task.

Total Number Of Runs	30
Mean of fitness function (mM.s)	36.268 ± 2.8E-14
Mean Number Of Knockouts	15.2 ± 3.6
Confidence Level	95%

4.2 Kinetic parameter optimization

In Table 2 the results obtained for the v_{max} parameter optimization scenario are shown. Figure 4 shows the boxplot concerning the level of change in the v_{max} parameter for a number of reactions. To better compare with the previous experiment, this set is composed of the reactions that were most frequently the target of a knockout.

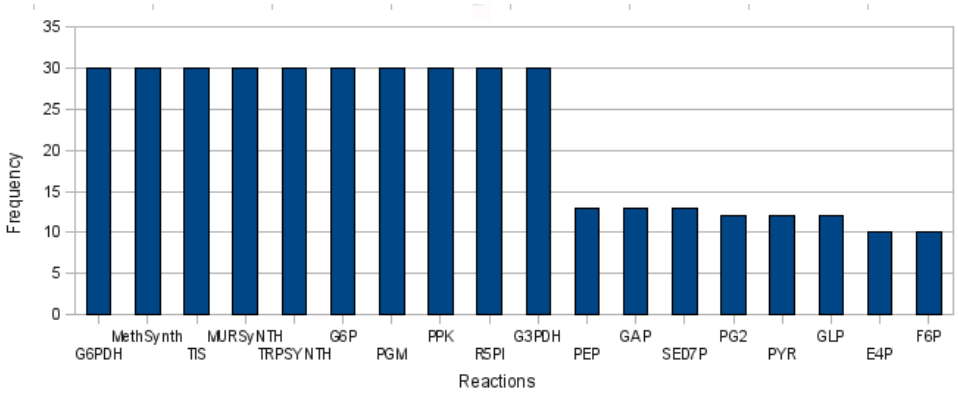

Fig. 3. Knockout frequency graph: only the reactions that have a frequency of at least 10 are shown.

Table 2. Results for the kinetic parameter optimization task.

Total Number Of Runs	30
Mean of fitness function (mM.s)	77.011 ± 3.88
Confidence Level	95%

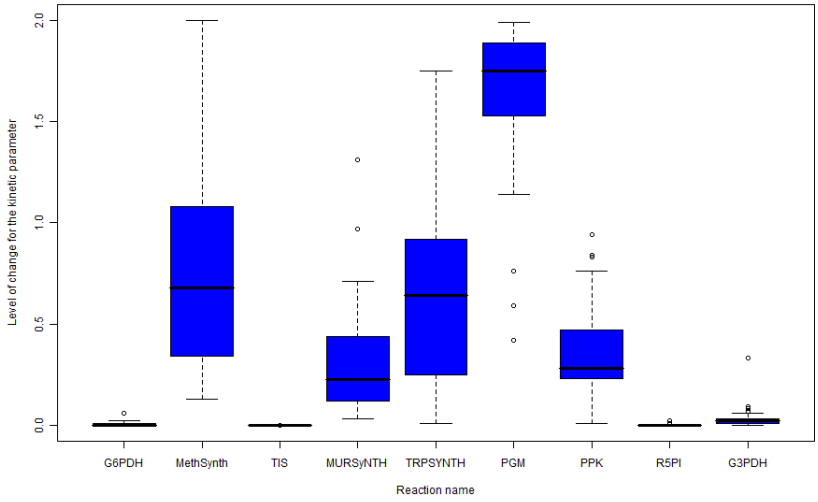

Fig. 4. Level of change for the v_{max} parameter in a selected set of reactions (selected from the set of reactions that were frequently knocked out in the previous experiment).

4.3 Discussion

Regarding the first task, it is interesting to note that none of the main reactions that lead to the production of DHAP (PTS, PGI, PFK and ALDO) are knocked out in any of the best solutions for each run. The reactions that may impact negatively in the production of DHAP, even if not directly, have a higher chance of being knocked out as it can be seen in Figure 3. This result validates the proposed approach.

However, it is important to mention that the obtained knockout sets are not likely to produce viable mutants due to the fact that there are no restrictions regarding the set of possible reactions to inactivate. The reaction set composed by G6PDH, TIS, G3PDH and R5PI is always deleted in all the best individuals, thus inhibiting metabolic pathways like nucleotide and glycerol synthesis and thus suppressing biomass formation.

During the v_{max} parameter optimization, the best solutions lead to an increased production of DHAP. This is explained by the fact that, when tuning the v_{max} parameter for each reaction, we are allowing the reactions to have a reduced activity (if the reduction factor is 0 the reaction can even be knocked out as before) or an increased activity (the v_{max} value can be doubled). This provides much more flexibility and leads to higher values of the fitness function. Also, in contrast with the previous scenario, the mutants are more likely to be viable because most of the metabolic pathways are not completely inactivated.

The v_{max} values obtained in the best solutions typically show an inactivation of the reactions unnecessary to the production of DHAP, resembling the knockout sets produced in the first scenario, as it is obvious comparing the results of Figure 4 and Figure 3. The v_{max} parameter value for the PGM reaction is the only one that increases its base level value. This is due to the fact that it is the only parameter that lies in the denominator of the kinetic expression. All other parameters under optimization are being multiplied by the numerator of the reaction rate laws.

The optimization results for the studied scenarios emphasize the complex interactions involved, even in this very simple model. The methods presented merely serve as a proof of concept, since most of the solutions are likely to be biologically non-viable. Those limitations of the approach are mainly due to the nature of the models (the used model is known to be incomplete) and the constraints imposed over the solutions in these experiments.

If a more complete model is used and the constraints are biologically correct, the proposed framework can be used to reach biologically meaningful results. For example, is this case, to use this approach in a real metabolic engineering approach, some of the reactions would have to be constrained not to be a target for deletion and the limits over the v_{max} parameters would have to be carefully imposed.

5 Conclusions and further work

In recent years, several methods have been developed *in silico* with the purpose of identifying and characterizing microrganisms' metabolic functioning. So far, research has been mostly confined to explore parameter estimation problems, based on fitting experimental data. On the other hand, Metabolic Engineering related approaches are based in steady state models. This study focus on studying novel ways of exploring dynamical models to optimize model modifications (e.g. model structure or parameter values) in different settings using as objective function the maximization of the production of a given metabolite of industrial interest.

The modular architecture of the proposed framework allows to replace any component of the dynamical model. For instance, when the rate law of a reaction has an unknown mathematical expression for a given model it can be replaced by a model built based on experimental data (e.g. a trained neural network).

In future work, the main issues to be tackled are the validation of this framework with other real-world case studies and also to make the computational tools available to the research community by integrating them in a proper platform with appropriate graphical user interfaces.

Regarding the optimization layer, a number of other algorithms have to be integrated in the framework, namely Genetic Programming [11] and Artificial Immune Systems [8] should be considered. The use of multi-objective optimization algorithms [15] in the optimization layer is also a promising route.

References

1. Benjamin J. Bornstein, Sarah M. Keating, Akiya Jouraku, and Michael Hucka. LibSBML: an API Library for SBML. *Bioinformatics*, 24(6):880–881, 2008.
2. Christophe Chassagnole, Naruemol Noisommit-Rizzi, Joachim W. Schmid, Klaus Mauch, and Matthias Reuss. Dynamic modeling of the central carbon metabolism of *Escherichia coli*. *Biotechnology and Bioengineering*, 79(1):53–73, 2002.
3. S. Cohen and C. Hindmarsh. Cvode, a stiff/nonstiff ode solver in c. *Computers in Physics*, 10(2):138–143, March 1996.
4. Jeremy S Edwards and Bernhard O Palsson. Metabolic flux balance analysis and the in silico analysis of escherichia coli k-12 gene deletions. *BMC Bioinformatics*, 1(1):1–1, 2000.
5. Thierry Gefflaut, Marielle Lemaire, Marie-Lise Valentin, and Jean Bolte. A novel efficient synthesis of dihydroxyacetone phosphate and bromoacetol phosphate for use in enzymatic aldol syntheses. *The Journal of Organic Chemistry*, 62(17):5920–5922, 1997.
6. Marc Daniel Haunschild, Bernd Freisleben, Ralf Takors, and Wolfgang Wiechert. Investigating the dynamic behavior of biochemical networks using model families. *Bioinformatics*, 21(8):1617–1625, 2005.
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. 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. Le 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(4):524–531, 2003.

8. Yoshiteru Ishida. The immune system as a self-identification process: A survey and a proposal. In *In Proceedings of the IMBS96*, 1996.
9. JFreeChart. http://www.jfreechart.org/jfreechart.
10. Junji Kitagawa and Hitoshi Iba. *Identifying metabolic pathways and gene regulation networks with evolutionary algorithms.* Morgan Kauffman, 2003.
11. John R. Koza. Genetic programming, 1992.
12. P Mendes and D Kell. Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics*, 14(10):869–883, 1998.
13. Jeremiah Nummela and Bryant A. Julstrom. Evolving petri nets to represent metabolic pathways. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 2133–2139, New York, NY, USA, 2005. ACM.
14. K. Patil, I. Rocha, J. Forster, and J. Nielsen. Evolutionary programming as a platform for in silico metabolic engineering. *BMC Bioinformatics*, 6(308), 2005.
15. P.Maia, Eugenio C.Ferreira, I.Rocha, and M.Rocha. Evaluating evolutionary multiobjective algorithms for the in silico optimization of mutant strains. In *8th IEEE International Conference on BioInformatics and BioEngineering Workshops(BIBE2008).Athens, Greece*, 2008.
16. Miguel Rocha, Paulo Maia, Rui Mendes, Eugenio C. Ferreira, Kiran Patil, Jens Nielsen, and Isabel Rocha. Natural computation meta-heuristics for the in silico optimization of microbial strains. *BMC Bioinformatics*, 9(499), 2008.
17. R. Storn and K. Price. Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11:341–359, 1997.
18. Joachim Thiem. Applications of enzymes in synthetic carbohydrate chemistry. *FEMS Microbiology Reviews*, 16(2-3):193–211, 1995.
19. Q.J. Wang. Using genetic algorithms to optimise model parameters. *Environmental Modelling and Software with Environment Data News*, 12:27–34(8), 1997.
20. Kun Yang, Wenzhe Ma, Huanhuan Liang, Qi Ouyang, Chao Tang, and Luhua Lai. Dynamic simulations on the arachidonic acid metabolic network. *PLoS Computational Biology*, preprint(2007):e55.eor+, February 2007.