Metaheuristics for Strain Optimization using Transcriptional Information Enriched Metabolic Models

Paulo Vilaça[1,2], Paulo Maia[1,2], Isabel Rocha[2], and Miguel Rocha[1]

[1] Department of Informatics / CCTC - University of Minho
{paulo.maia,mrocha}@di.uminho.pt
[2] IBB - Institute for Biotechnology and Bioengineering
Centre of Biological Engineering - University of Minho
Campus de Gualtar, 4710-057 Braga - PORTUGAL
{pvilaca,irocha}@deb.uminho.pt

Abstract. The identification of a set of genetic manipulations that result in a microbial strain with improved production capabilities of a metabolite with industrial interest is a big challenge in Metabolic Engineering. Evolutionary Algorithms and Simulated Annealing have been used in this task to identify sets of reaction deletions, towards the maximization of a desired objective function. To simulate the cell phenotype for each mutant strain, the Flux Balance Analysis approach is used, assuming organisms have maximized their growth along evolution.
In this work, transcriptional information is added to the models using gene-reaction rules. The aim is to find the (near-)optimal set of gene knockouts necessary to reach a given productivity goal. The results obtained are compared with the ones reached using the deletion of reactions, showing that we obtain solutions with similar quality levels and number of knockouts, but biologically more feasible. Indeed, we show that several of the previous solutions are not viable using the provided rules.

Key words: Metabolic Engineering, Strain Optimization, Flux-Balance Analysis, Transcriptional Models, Set based representations

1 Introduction

Over the last few years, the combined efforts of Metabolic Engineering and Systems Biology have allowed the development of some genome-scale metabolic models for several microorganisms, with an industrial interest in Biotechnology. These have been used to predict cellular phenotypes under some simplifying assumptions, aiding in the effort of finding appropriate genetic modifications to make the microorganism fit to comply with industrial purposes, i.e. to be able to synthesize some desired compounds in significant amounts, rather then to follow their natural aims (e.g. the maximization of growth) [14][8].

The most popular approach considers the cell to be in a steady-state, i.e., the concentrations of all intracellular compounds are assumed to remain constant throughout time. Together with the known stoichiometry and reversibility

or irreversebility of the reactions, this assumption is used in a constraint-based framework to restrict the set of possible values for the fluxes of the reactions contained in the metabolic model. Therefore, cellular behavior can be predicted by addressing the underlying optimization problems, given a biologically plausible objective function. The Flux Balance Analysis approach [6] follows this path, maximizing a particular flux, typically for biomass production, using linear programming [5]. Solving this problem allows to reach the values for all the reaction fluxes.

Using this approach or others recently proposed for the same purpose (e.g. MOMA [12], ROOM [13]), it is possible to predict the behavior of a microorganism under distinct environmental and genetic conditions (such as gene deletions). Indeed, both can be represented by adding/ changing constraints under the previous framework. Therefore, both wild type and mutant strains can be simulated. This has allowed the definition of a bi-level strain optimization problem, adding a layer that searches for the best mutant that can be obtained by applying a set of selected genetic modifications. In previous work, this has been restricted to the possibility of removing reactions from the original model. The idea is to force the microorganisms to synthesize a desired product, while keeping it viable. The optimization task consists in reaching an optimal subset of reaction deletions to optimize an objective function related with the production of a given compound.

A first approach to this problem was the *OptKnock* algorithm [1], where mixed integer linear programming methods are used to reach a guaranteed optimum solution. However, this algorithm does not allow to consider nonlinear objective functions and a considerable computation time is required. An alternative was proposed by the *OptGene* algorithm [9], that uses Evolutionary Algorithms (EAs). EAs are capable of providing near optimal solutions in a reasonable amount of time and also allow the optimization of nonlinear objective functions. Extending this work, the authors [11] proposed a new encoding scheme for the problem, consisting in variable-sized sets, allowing the automatic determination of the ideal number of reactions to eliminate, since solutions with distinct cardinalities compete within the search space. Also, a Simulated Annealing (SA) based algorithm was put forward for the same task. Both algorithms were tested with four case studies and the SA presented some advantage over the EA.

A common limitation of these approaches is the fact that they rely on determining sets of reactions to be eliminated from the metabolic model, while the real purpose is to determine a set of genes to knockout. Therefore, to create the desired mutants in the lab there is the need to determine which set of genes can lead to the elimination of a given set of reactions. This would not be a problem if the rule 1 gene - 1 enzyme - 1 reaction was universal. However, this is not the case, since there are many exceptions, due to iso-enzymes, protein complexes, enzymes that catalyze several reactions or reactions that can be catalyzed by several enzymes.

The solution is, therefore, to use transcriptional information in association with the genome-scale metabolic model. This approach is mostly limited by

the lack of information available, since in most cases there is no comprehensive model of transcriptional information available. However, this situation is gradually changing and some metabolic models with transcriptional information of well known microorganisms are appearing [10].

In this work, we propose phenotype simulation and strain optimization methods that are able to take advantage on this transcriptional information. The optimization methods will be able to suggest sets of genes to knockout replacing the reaction list usually provided. Two case studies related to the production of succinate and lactate using the bacterium *Escherichia coli* will be presented to evaluate the approach. We will also study in detail the major differences between the reaction and gene based approaches and compare the results obtained.

2 Simulation algorithms for the prediction of metabolic behavior

2.1 Flux balance analysis

The Flux Balance Analysis (FBA) [6] approach is based on a steady state approximation to the concentrations of internal metabolites, which reduces the corresponding mass balances to a set of linear homogeneous equations. For a network of M metabolites and N reactions, this is expressed as:

$$\sum_{j=1}^{N} S_{ij} v_j = 0 \tag{1}$$

where S_{ij} is the stoichiometric coefficient for metabolite i in reaction j and v_j is the flux over the reaction j. The maximum/minimum values of the fluxes can be set by additional constraints in the form $\alpha_j \leq v_j \leq \beta_j$, usually used to specify both thermodynamic and environmental conditions (e.g. availability of nutrients).

For most metabolic networks, since the number of fluxes is greater than the number of metabolites, the set of linear equations obtained from the application of Eq. 1 to the M metabolites usually leads to an under-determined system, for which there exists an infinite number of feasible flux distributions that satisfy the constraints. However, if a given linear function over the fluxes is chosen to be maximized, it is possible to obtain a single solution by applying standard algorithms (e.g. *simplex*) for linear programming problems.

The combination of this technique with the existence of validated genome-scale stoichiometric models [2] allows to simulate the phenotypic behavior of a microorganism, under defined environmental conditions, without performing any experiments. The most common flux chosen for maximization is the biomass, based on the premise that microorganisms have maximized their growth along natural evolution, a premise that has been validated experimentally for some situations[5].

2.2 Integrating transcriptional information

Recently, some studies attempted to improve the characterization of organisms by inserting a transcriptional layer into the metabolic models [10, 3, 15]. Thus, adding this level of information about the behavior of biological systems, raises the metabolic models into the genetic level, where the metabolic processes depend on the genes that encode enzymes which catalyse metabolic reactions.

To create this transcriptional layer it is necessary to define the cascade of interactions between genes, proteins, peptides and reactions of a given system. These are not easy to find due to the complexity of the different types of interactions between biological entities:

- the genes encode the information that leads to the creation of peptides through the processes of transcription and translation;
- proteins can be constructed from one or more peptides;
- proteins can bind to create protein complexes;
- the reactions are catalyzed by enzymes (proteins or protein complexes);
- more than one protein can catalyze the same reaction (iso-enzymes);
- a single protein can catalyze more than one reaction.

In this work, all available transcriptional information will be transformed into gene-reaction rules. Gene-reaction rules are based on boolean logic representation. For each reaction (dependent variable), there is a boolean expression, where the independent variables are the encoding genes; their interactions are defined using logical operations (AND, OR). In Figure 1, some examples of different associations between genes, peptides, proteins and reactions are shown, as well as their simplification for gene-reaction rules.

3 Strain optimization

3.1 Problem definition, solution encoding and evaluation

The problem addressed in this work consists in selecting, from a set of genes in a microbe's genome-scale model, a subset to be deleted to maximize a given objective function. The encoding of a solution is achieved by a variable size set-based representation, where only gene deletions are represented. Each solution consists of a set of integer values representing the genes that will be deleted. Therefore, if the value i is in the set, this means the i-th gene in the model is removed. Each value in the set is an integer with a value between 1 and G, where G is the number of genes in the model.

The first step is to take the genes indexed by the solution and then calculate which reactions will be removed as a consequence of knocking out these genes, using the transcriptional information. For all reactions involved, the flux will be constrained to 0, therefore disabling that reaction in the metabolic model. The process proceeds with the simulation of the mutant using FBA. The output is the set of values for the fluxes of all reactions, that are then used to compute

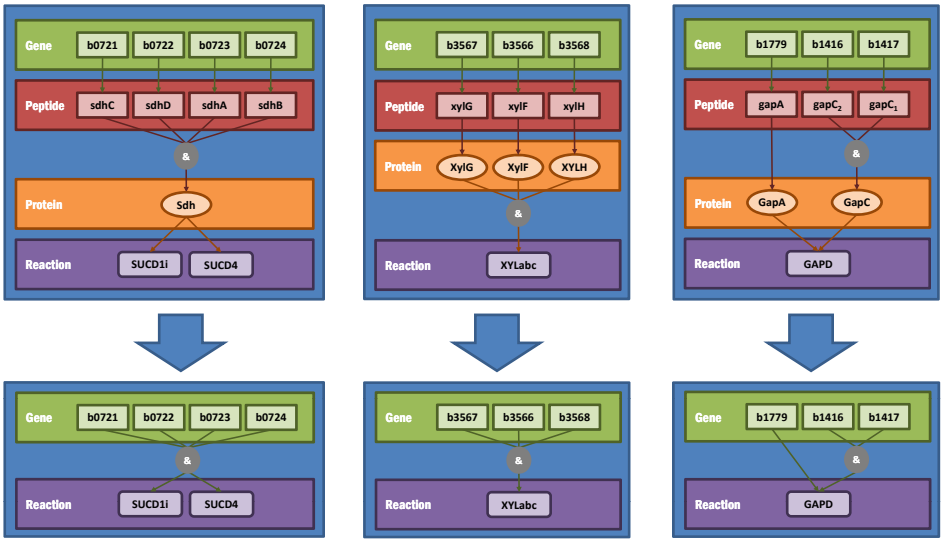

Fig. 1. Schematic representation of the transcriptional information included in the metabolic models.

the fitness value, given by an appropriate objective function. The used objective function is the Biomass-Product Coupled Yield (BPCY) [9], given by:

$$BPCY = \frac{PG}{S} \tag{2}$$

where P stands for the flux representing the excretion of the desired product; G for the organism's growth rate (biomass flux) and S for the substrate intake flux. Besides optimizing for the production of the desired product, this function also allows to select for mutants that exhibit high growth rates. The complete process of decoding and evaluation is depicted in Figure 2.

3.2 Evolutionary Algorithms

To address the previous task, we will use Evolutionary Algorithms (EAs) with a set-based representation, previously proposed in [11]. This EA uses four reproduction operators: one crossover and three mutation operators. The crossover operator is inspired on traditional uniform crossover operators 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.

A random mutation operator is used that replaces a gene by a random value in the allowed range, avoiding duplicates in the set. Two additional mutation operators are defined to be able to create solutions with a distinct size:

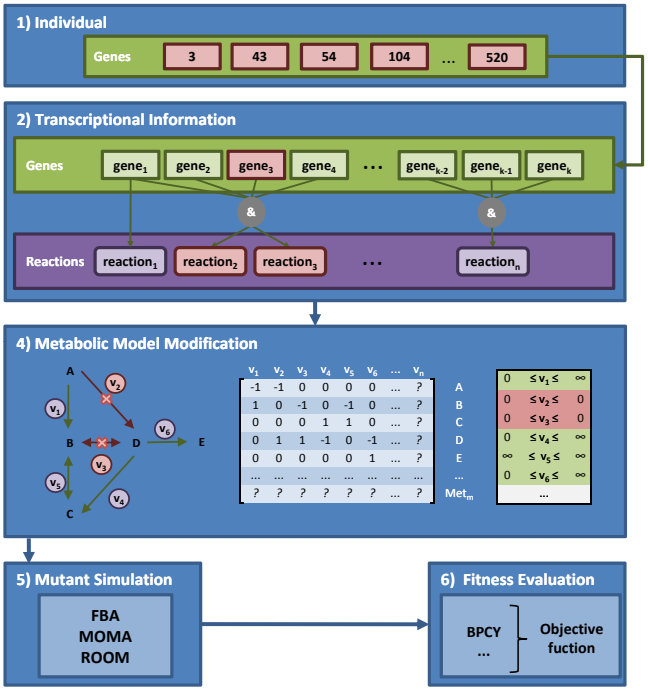

Fig. 2. Scheme of the phenotypic simulation methods using transcriptional information and their transformation into gene-reaction rules.

- *Grow*: consists in the introduction of a new gene into the solution, whose value is randomly generated in the available range (avoiding duplicates in the set).
- *Shrink*: a randomly selected gene is removed from the genome.

The Grow and Shrink mutation operators are each used with a probability of 5% each. The remaining operators are used with equal probabilities. The EA uses a selection procedure that consists in converting the fitness value into a linear ranking of the individuals in the population, and then applying a roulette wheel scheme. In each generation, 50% of the individuals are kept from the previous generation, and 50% are bred by the application of the reproduction operators. An initial population is randomly created and the termination criterion is based on a fixed number of solution evaluations.

3.3 Simulated Annealing

Also, Simulated Annealing (SA) was used to address the optimization task and compare the results. As before, the SA is also similar to the one proposed by the authors in [11]. The SA makes use of the same set-based representation used in

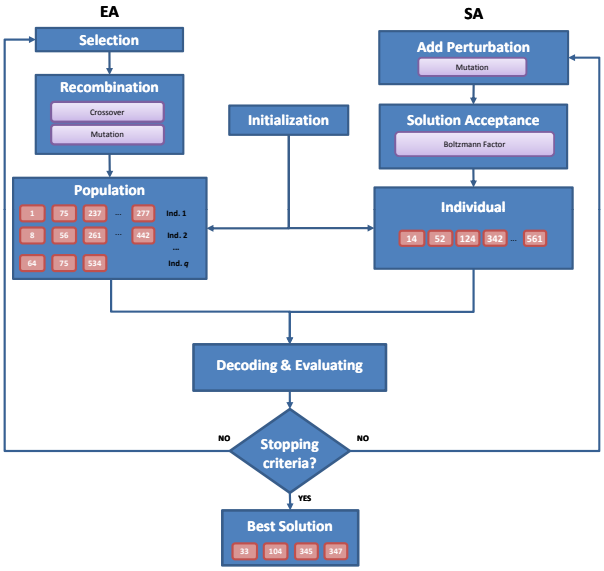

Fig. 3. Illustration of the structure of the strain optimization algorithms.

the EA, also keeping the mutation operators presented before. An illustration of the structure of both algorithms is given in Figure 3.

The cooling schedule used is exponential, decreasing the temperature T according to: $T_{n+1} = \alpha T_n$, where $0 < \alpha \leq 1$. As the choice of initial (T_0) and final temperatures (T_f) is problem dependent, it was decided to use the following configuration parameters:

ΔE_0 – The difference in energy that corresponds to an acceptance probability of 50% of worse solutions at the beginning of the run;
ΔE_f – The difference in energy that corresponds to an acceptance probability of 50% of worse solutions at the end of the run;
trials – The number of iterations per temperature;
NFEs – The number of function evaluations.

Using these parameters, the initial temperature, the final temperature and the scale parameter were computed using the following equations:

$$T_0 = -\frac{\Delta E_0}{\log 0.5} \tag{3}$$

$$T_f = -\frac{\Delta E_f}{\log 0.5} \tag{4}$$

$$\alpha = \exp\left(\frac{\log T_f - \log T_0}{\left[\frac{\text{NFEs}}{\text{trials}}\right]}\right) \tag{5}$$

The advantage of using ΔE_0 and ΔE_f is that it allows the user who knows the fitness landscape of the optimization problem to automatically define the temperatures by reasoning over the values of the objective function. Supplying the number of function evaluations instead of the scale parameter α allows the user to accurately define the number of function evaluations the optimization algorithm will use, enabling a simpler comparison with other approaches.

In the SA, the *Grow* and *Shrink* mutations are each used with a probability of 25% each, meaning that half of the new individuals are created in this way. The remaining are created by the aforementioned random mutation operator.

3.4 Pre-processing and post-processing

In genome-scale models the number of variables (genes/ reactions) is in the order of hundreds or a few thousands and therefore the search space is very hard to address. Thus, every operation that gives a contribution to reduce this number, greatly improves the convergence of the algorithms. In this work, two operations were implemented to reduce the search space:

- Removal of reactions that, given the constraints of the linear programming problem, cannot exhibit flux values different from 0. All genes only encoding those reactions are also removed.
- Discovery of essential genes that can not be deleted from the model since their removal leads to non growth (biomass flux value of zero). As these genes should not be considered as targets for deletion, the search space for optimization is reduced.

Also, the best solution in each run goes through a simplification process, by identifying all gene deletions that contribute to the fitness of the solution, and removing all deletions that keep the objective function unaltered. The aim is to keep only the necessary knockouts.

3.5 Implementation issues

The implementation of the proposed algorithms was performed by the authors in the *Java* programming language. In the implementation of FBA, the *GNU linear programming package (GLPK)*[3] was used to run the *simplex* algorithm. An user interface was also built within the OptFlux framework [4], a Metabolic Engineering open-source software platform.

4 Experiments

4.1 Experimental setup

Two case studies were used to test the algorithms, both considering the microorganism *Escherichia coli*. The aim is to produce succinate and lactate with

[3] http://www.gnu.org/software/glpk/
[4] http://www.optflux.org

glucose as the limiting substrate. The genome-scale model used in the simulations was developed by Reed et al [10]. This model considers the metabolic network of *E. coli*, including a total of $N = 1075$ fluxes, $M = 761$ metabolites, $G = 904$ genes and 873 gene-reaction rules. After the pre-processing stages, the simplified model remains with $N = 610$, $M = 383$ metabolites, 617 genes and 562 gene-reaction rules. Furthermore, 115 essential genes are identified, which leaves 502 variables to be considered by the optimization algorithms.

In the EA the population size was set to 100. The SA used $\Delta E_0 = 0.005$, $\Delta E_f = 5E^{-5}$ and $trials = 50$. In both cases, the termination criterion was defined based on 50000 fitness evaluations. For each configuration, the process was repeated for 30 runs and the mean and standard deviation were calculated.

4.2 Case studies

Succinate is one of the key intermediates in cellular metabolism and therefore an important case study for metabolic engineering [7]. The knockout solutions that lead to an improved phenotype regarding its production are not straightforward to identify since they involve a large number of interacting reactions. Succinicate and its derivatives have been used to synthesize polymers, as additives and flavoring agents in foods, supplements for pharmaceuticals, or surfactants. Currently, it is mostly produced through petrochemical processes that can be expensive and have significant environmental impacts.

Lactate and its derivatives have been used in a wide range of food-processing and industrial applications like meat preservation, cosmetics, oral and health care products. Additionally, and because lactate can be easily converted to readily biodegradable polyesters, it is emerging as a potential material for producing environmentally friendly plastics from sugars [4]. Several microorganisms have been used to produce lactate, such as *Lactobacillus* strains. However, those bacteria have undesirable traits, such as a requirement for complex nutrients which complicates acid recovery. *E. coli* has many advantageous characteristics, such as rapid growth and simple nutritional requirements.

4.3 Results

In Tables 1 and 2 we show the results for both case studies, taking the BPCY as the objective function. In both cases, we show the results for our current approach using transcriptional information, compared to the results using the previous method based on reaction deletions [11]. It should be emphasized that all the setup is the same for both cases. The first two columns show the optimization target (genes or reactions) and the algorithm used (EA or SA). In the third and fourth columns, we show the mean of the BPCY and of the number of knockouts over the 30 runs, also showing the standard deviation (surrounded by parentheses). Finally, the last column shows the BPCY and the number of knockouts of the best solution obtained over the 30 runs.

Also, we investigated how the solutions obtained for reaction based optimization can be converted into a gene knockout set. So, we analyzed the best

Table 1. Results for the succinate case study.

Optimization Type	Algorithm	Fitness (BPCY)	Number Knockouts	Best Solution
Reactions	EA	0.35345 (0.01405)	11.7 (2.6)	0.35785 (15)
Reactions	SA	0.35766 (0.00015)	9.7 (1.0)	0.35781 (11)
Genes	EA	0.23188 (0.09945)	10.6 (1.9)	0.34429 (7)
Genes	SA	0.30636 (0.07713)	10.4 (4.0)	0.34429 (10)

Table 2. Results for the lactate case study

Optimization Type	Algorithm	Fitness (BPCY)	Number Knockouts	Best Solution
Reactions	EA	0.21387 (0.09180)	9.7 (7.3)	0.34786 (5)
Reactions	SA	0.27654 (0.05713)	16.8 (8.9)	0.34843 (26)
Genes	EA	0.25447 (0.05215)	10.3 (2.9)	0.34786 (5)
Genes	SA	0.25428 (0.05039)	12.1 (4.3)	0.29328 (8)

solution obtained in each of the 30 runs according to the following: (i) we took the set of reactions to delete and calculated the minimum set of genes that had to be removed in order to inactivate those reactions; (ii) we checked if there were other reactions that would be inactivated as a result of those gene deletions; (iii) finally, we simulated the resulting mutant strain and calculated the BPCY.

The results are given in Table 3 where we show, for each case, the number of solutions (over the best solutions in each run) where the BPCY is still larger than zero (third column), the number of solutions that keep the same BPCY (fourth column) and also the mean number of knockouts that were added in step (ii) of the previous process.

Table 3. Conversion of reaction deletion based solutions to gene deletion based solutions

Case study	Algorithm	$BPCY > 0$	same $BPCY$	Additional Knockouts
Succinate	EA	0/30	0/30	12.5
Succinate	SA	0/30	0/30	6.8
Lactate	EA	8/30	5/30	8.0
Lactate	SA	8/30	3/30	15.7

4.4 Discussion

The first conclusion to retain is that the overall objective function results, when optimizing gene deletions, are quite near the ones obtained before by deleting reactions. Although in most cases the BPCY is slightly lower, the differences are generally not statistically significant. Also, the number of knockouts does not increase, even decreasing in most cases (again differences are not significant).

The differences in performance, when they exist, are small when compared to the gains obtained considering that models with transcriptional information characterize better the behavior of the organism and the simulation using this level of information is closer to the biological reality and thus more reliable. Also, the implementation of the solutions in the lab will be based on a gene list, which makes these results easier to implement.

Studying the results in more depth we see that the succinate case study seems to have larger differences between the two approaches. Looking at Table 3 we can understand the reasons, since we see that all the best solutions obtained are unfeasible at the level of genes (reporting a value of 0 for the BPCY) and therefore impossible to implement in the lab. From that table, we also conclude that in the lactate case study most of the solutions (around 70%) also have a BPCY of 0 and some of the others deteriorate the fitness value. This shows that, in general, it seems unlikely that solutions reached with reaction deletion based optimization are biologically feasible (i.e. can be implemented through gene knockouts). Comparing both meta-heuristics for optimization, we observe that the SA and EA shown very similar performances, but the SA confirms a slight advantage, already reported in [11].

5 Conclusions and further work

In this work, we have studied the effects of using transcriptional information to complement the knowledge contained in metabolic models, on the results of strain optimization algorithms such as EA and SA. The main conclusion of this analysis indicates that most solutions obtained previously, considering reaction deletion optimization, are impossible to translate to gene knockouts and therefore to implement in the lab.

We proposed improved algorithms for the tasks of phenotype simulation and strain optimization that can take advantage on the transcriptional information. The results obtained by those methods reveal an overall solution quality very similar to the previous methods and the number of suggested knockouts does not increase, also an important result considering the feasibility of the solutions.

Since these solutions are biologically more feasible, we believe that an important step has been made towards the use of these methods in Biotechnology. Also with this aim, we have implemented these methods under *OptFlux*, an open-source software platform. This allows the methods to be used freely by the Metabolic Engineering community.

As future work, we aim to apply these methods to other relevant case studies in Metabolic Engineering, considering other target compounds, as well as other

organisms and models. Also, the integration of regulatory information with these models, in the form of new constraints, is a promising path.

Acknowledgements

This work was partially funded by Portuguese FCT through the AspectGrid project and also through project MIT-PT/BS-BB/0082/2008.

References

1. A.P. Burgard, P. Pharya, and C.D. Maranas. Optknock: A bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnol Bioeng*, 84:647–657, 2003.
2. M.W. Covert, C.H. Schilling, I. Famili, J.S. Edwards, I.I. Goryanin, E. Selkov, and B.O. Palsson. Metabolic modeling of microbial strains *in silico*. *Trends in Biochemical Sciences*, 26(3):179–186, 2001.
3. N.C. Duarte, M.J. Herrgård, and B. Ø. Palsson. Reconstruction and validation of *Saccharomyces cerevisiae* ind750, a fully compartmentalized genome-scale metabolic model. *Genome Res*, 14(7):1298–309, 2004.
4. K. Hofvendahl and B. Hahn-Hagerdal. Factors affecting the fermentative lactic acid production from renewable resources. *Enzyme Microbial Technology*, 26:87–107, 2000.
5. R.U. Ibarra, J.S. Edwards, and B.G. Palsson. *Escherichia coli* k-12 undergoes adaptive evolution to achieve *in silico* predicted optimal growth. *Nature*, 420:186–189, 2002.
6. K.J. Kauffman, P. Prakash, and J.S. Edwards. Advances in flux balance analysis. *Curr Opin Biotechnol*, 14:491–496, 2003.
7. S.Y. Lee, S.H. Hong, and S.Y. Moon. *In Silico* metabolic pathway analysis and design: succinic acid production by metabolically engineered *Escherichia coli* as an example. *Genome Informatics*, 13:214–223, 2002.
8. J. Nielsen. Metabolic engineering. *Appl Microbiol Biotechnol*, 55:263–283, 2001.
9. K. Patil, I. Rocha, J. Forster, and J. Nielsen. Evolutionary programming as a platform for *in silico* metabolic engineering. *BMC Bioinformatics*, 6(308), 2005.
10. J.L. Reed, T.D. Vo, C.H. Schilling, and B.O. Palsson. An expanded genome-scale model of *Escherichia coli* k-12 (ijr904 gsm/gpr). *Genome Biology*, 4(9):R54.1–R54.12, 2003.
11. M. Rocha, P. Maia, R. Mendes, J.P. Pinto, E.C. Ferreira, J. Nielsen, K.R. Patil, and I. Rocha. Natural computation meta-heuristics for the *in silico* optimization of microbial strains. *BMC Bioinformatics*, 9, 2008.
12. D. Segre, D. Vitkup, and G.M. Church. Analysis of optimality in natural and perturbed metabolic networks. *PNAS*, 99:15112–15117, 2002.
13. T. Shlomi, O. Berkman, and E. Ruppin. Regulatory on/off minimization of metabolic flux changes after genetic perturbations. *PNAS*, 102(21):7695–7700, 2005.
14. G. Stephanopoulos, A.A. Aristidou, and J. Nielsen. *Metabolic engineering principles and methodologies*. Academic Press, San Diego, 1998.
15. I. Thiele, T.D. Vo, N.D. Price, and B. Ø. Palsson. Expanded metabolic reconstruction of *Helicobacter pylori* (iit341 gsm/gpr): an *in silico* genome-scale characterization of single- and double-deletion mutants. *J Bacteriol*, 187(16):5818–30, 2005.