Highlighting Metabolic Strategies Using Network Analysis over Strain Optimization Results

José Pedro Pinto[1,2], Isabel Rocha[2], and Miguel Rocha[1]

[1] Department of Informatics / CCTC - University of Minho
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
irocha@deb.uminho.pt

Abstract. The field of Metabolic Engineering has been growing, supported by the increase in the number of annotated genomes and genome-scale metabolic models. *In silico* strain optimization methods allow to create mutant strains able to overproduce certain metabolites of interest in Biotechnology. Thus, it is possible to reach (near-) optimal solutions, i.e. strains that provide the desired phenotype in computational phenotype simulations. However, the validation of the results involves understanding the strategies followed by these mutant strains to achieve the desired phenotype, studying the different use of reactions/ pathways by the mutants. This is quite complex given the size of the networks and the interactions between (sometimes distant) components. The manual verification and comparison of phenotypes is typically impossible.

Here, automatic methods are proposed to analyse large sets of mutant strains, by taking the phenotypes of a large number of possible solutions and identifying shared patterns, using methods from network topology analysis. The topological comparison between the networks provided by the wild type and mutant strains highlights the major changes that lead to successful mutants. The methods are applied to a case study considering *E. coli* and aiming at the production of succinate, optimizing the set of gene knockouts to apply to the wild type. Solutions provided by the use of Simulated Annealing and Evolutionary Algorithms are analyzed. The results show that these methods can help in the identification of the strategies leading to the overproduction of succinate.

Keywords: Metabolic Engineering, Strain optimization, Metabolic networks, Network visualization.

1 Introduction

Recent efforts in Bioinformatics and Systems Biology allowed the development of genome-scale metabolic models for several microorganisms [1]. These models have been used to guide biological discovery promoting the comparison between

M. Loog et al. (Eds.): PRIB 2011, LNBI 7036, pp. 109–120, 2011.

predicted and experimental data, to foster Metabolic Engineering (ME) efforts in finding appropriate genetic modifications to synthesize desired compounds, to analyze global network properties and to study bacterial evolution [2].

The most popular approach to phenotype simulation considers the cell to be in steady-state and takes reaction stoichiometry/ reversibility in a constraint-based framework to restrict the set of possible values for the reaction fluxes. Cellular behaviour is thus predicted using for instance Flux Balance Analysis (FBA), based on the premise that microorganisms have maximized their growth along natural evolution [3]. Using FBA, it is possible to predict the behaviour of microbes under distinct environmental/ genetic conditions.

The combination of reliable models with efficient simulation methods has been the basis for different strain optimization algorithms. Their goal is to find the set of genetic modifications to apply to a given strain, to achieve an aim, typically related with the industrial production of a metabolite of interest.

In previous work, an approach based in the use of metaheuristics, such as Evolutionary Algorithms (EAs) and Simulated Annealing (SA), has been proposed to solve the optimization task of reaching an optimal (or near optimal) subset of reaction deletions to optimize an objective function related with the production of a given compound [9]. The idea is to force the microbe to synthesize a desired product, while keeping it viable.

The next logical step is to validate these results in the lab, a task that given its associated costs should be preceded by a thorough analysis of the solutions provided using computational methods. This screening process could identify more promising approaches and, thus, save resources in wet lab experiments.

In a first stage, the validation of the results involves the understanding of the strategies followed by these mutant strains to achieve the desired phenotype, by studying the different use of reactions/ pathways to achieve the desired metabolite and still keep the strain viable. This becomes quite complex given the size of the networks involved in genome-scale models and the interactions between (sometimes distant) components. The manual verification and comparison of the phenotypes of different mutants is typically impossible.

In this work, the major aim is the development of automatic methods to analyse large sets of mutant strains for specific ME problems. These methods take the phenotypes of a large number of possible solutions obtained by running strain optimization algorithms and attempt to identify shared patterns, taking advantage of methods from network topology analysis. The topological comparison between the networks provided by the wild type and mutant strains highlights the major changes, thus highly contributing to elucidate the strategies that lead to successful mutants.

The methods are applied to a case study considering *Escherichia coli* as the host and aiming at the production of succinate, by optimizing the set of gene knockouts to apply to the wild type. Large sets of solutions (mutants) provided by the use of SA and EAs are analysed. To provide for large sets of possible solutions, the strain optimization algorithms were modified to keep all interesting solutions found during their execution.

The paper is organized as follows: next, a description of the computational methods is provided; this is followed by a description of the case study, the results obtained and its discussion; finally, conclusions and further work are provided.

2 Methods

2.1 Overall Workflow

In this work, the workflow used in the experiments can be summarized in the following steps:

- **Inputs:** a genome scale metabolic model of a host organism; a set of currency metabolites; a metabolite of interest to be overproduced;
- **Step 1:** the strain optimization algorithms (EA and SA) are executed with the provided configuration (see section 2.3); each algorithm is executed a given number of runs and the result from each run is a set of solutions (mutant strains) of interest;
- **Step 2:** the solution sets from the previous set are merged in a single set and filtered (see details in section 2.4);
- **Step 3:** each solution in the set from step 2 is simulated using FBA (section 2.2) and the corresponding network is created according to the methods described in section 2.5;
- **Step 4:** each of the networks from step 3 is compared to the wild type network, as described in section 2.6;
- **Step 5:** the comparisons from step 4 are analysed for common patterns of variability analysis (see details in section 2.7;
- **Step 6:** the results from the previous step are compiled in a sub-network that can be also visualized and manually analysed.

2.2 Flux Balance Analysis

In this work, FBA was used as the phenotype simulation method in the strain optimization tasks and to provide for the network filtering. FBA 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 [4]. For a network of M metabolites and N reactions, this is expressed as:

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

for every metabolite i, where S_{ij} is the stoichiometric coefficient for this metabolite 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 $\alpha_j \leq v_j \leq \beta_j$, also used to specify nutrient availability.

The set of linear equations obtained 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 solution by applying standard algorithms (e.g. *simplex*) for Linear Programming. The most common flux chosen for maximization is the biomass given the premise of optimal evolution that underlies FBA.

2.3 Strain Optimization

The problem addressed in this work consists of selecting, from a set of reactions 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 each solution consists of a set of reactions from the model that will be deleted. For all reactions deleted, the flux will be constrained to 0, therefore disabling them from the metabolic model. The process proceeds with the simulation of the mutant using the chosen phenotype simulation method (in this work, FBA). The output of these methods is the set of fluxes for all reactions, that are then used to compute the fitness value, given by the objective function.

Here, the objective function used is the Biomass-Product Coupled Yield (BPCY) [6], given by: $BPCY = \frac{PG}{S}$, where P stands for the flux representing the excreted 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. To address this task, we will use Simulated Annealing (SA) and Evolutionary Algorithms (EAs) as proposed previously in [9], where full details can be found regarding operators and other configuration details.

The implementation of the original EA and SA methods was only modified in order to keep not only the best solutions obtained during the run, but rather the whole set of solutions deemed to be interesting for analysis. This does not change the optimization process, but stores more intermediate results. Therefore, each run of the algorithms generated a *solution set* containing all solutions where both the value of P and G where larger than 0. This set includes only simplified solutions, i.e. solutions where the removal of a given knockout would reduce the fitness function value. All solutions are simplified by removing unnecessary knockouts before entering this set.

2.4 Solution Set Pre-processing

The first step in the pre-processing is to merge all the solutions coming from each individual run of the optimization. In this task, all duplicate solutions are removed. Also, the final solution set is checked for the existence of solutions where the set of knockouts is a superset of other solutions and these are only kept if its fitness value is higher.

The next step is to filter this solution set, since using all the solutions can be an undesirable option because in many cases they do not provide acceptable results from the biological standpoint. Also, the comparison process can be computationally heavy if the solution sets are too large. So, the number of solutions used was reduced filtering solutions: (i) with a low growth, by setting a minimal threshold for the biomass production flux; (ii) with a low production of the desired flux, by setting a minimal threshold for the flux associated with the excretion of the metabolite; (iii) filtering solutions that lead to the same set of reactions in the network after simulation filtering (see next section). In the experiments, the thresholds for (i) and (ii) correspond to 40% of the maximum value (this value was empirically set to keep solutions near to the best values obtained).

2.5 Network Representation and Simulation Filtering

The metabolic networks used in this project are directed bipartite graphs, where the nodes represent either metabolites or reactions and the edges the consumption or production of metabolites, pointing from a metabolite to a reaction in the former case and from a reaction to a metabolite in the latter.

The first step in this project was to create a network with all reactions and metabolites contained in the metabolic model, thus creating a network which can function as a map of the organism's metabolism, or more precisely a network with all the possibilities which can occur in a simulation. This network is called the *base network*.

All other networks were derived from the base network using a method denoted as *simulation filtering*. This is a process which creates a sub-network starting with the base network, taking the results of a phenotype simulation, and executing the following steps:

1. nodes which correspond to reactions with a flux of zero in the simulation are removed;
2. nodes which correspond to currency metabolites are removed (the list of currency metabolites is provided by the user for each model);
3. nodes which are left isolated (with no neighbours) after step 1 and 2 are removed.

The result of the simulation filtering is a network which is a ”snapshot” of how the metabolism behaves according to the results provided by a phenotype simulation method.

2.6 Network Comparison

The network comparison process is basically a series of operations in which each mutant network is compared with the wild type network used as a reference. These operations are typically followed by some global variation analysis to identify the most common patterns of network variation (see next section).

During the development of the methods and the tools used in this project it was noticed that the majority of the viable mutant networks are very similar to the wild type network, which limits the use of many global metrics for graph topology, such as centrality values, shortest path analysis or clustering coefficients as comparison metrics. This led us to define some novel network comparison metrics adapted to the purposes of the work and focused on the identification of local patterns of interest in metabolic networks. The set of analyses conducted and the metrics used are defined next.

Exclusivity. This comparison metric used is based on the set of exclusive nodes, i.e. those existing in one of the networks but not the other. In this case, for each mutant-wild type comparison two lists are created containing the set of nodes exclusive to each of the networks. After all mutants are compared to the wild type these lists are used to determine the frequency of each node in the exclusive lists.

Decision points. When analysing metabolic networks it is important to look not only at the topology but also to understand the flows over the network. For instance in the case of linear pathways with no splits, the existence of a flow in a reaction may be determined by another upstream reaction that can be distant. The decision point concept was thought as a way to determine the upstream network metabolite of pathways that exist on one of the networks and not the other. The first step in identifying decision points is to identify the decision metabolites, i.e. that are common to both networks, but that are consumed by different sets of reactions. The next step is to identify which reactions consuming these metabolites are present in one of the networks and not in the other.

Inversions. Many reactions in the metabolic model are classified as reversible, meaning that they can occur on both directions. The inversion metric is based in the fact that manipulations of the metabolism can result in a flux changing signs, meaning the reaction changes its direction. All the inversions that occur in the mutants compared to the wild type are identified and their frequency is calculated.

2.7 Variation Analysis

After all network comparisons between the mutants and the reference are conducted, the results obtained are used to identify common patterns. As a first step, the reactions that are exclusive in a significant part of the comparisons are identified. In this case, all reactions that are in the mutant exclusive lists with a frequency exceeding a threshold (in this work 80%) are identified. This group includes reactions typically used in mutant phenotype but not on the wild type strains. The same process is conducted for the wild type exclusive lists, thus identifying reactions used in the wild type but typically absent from the mutant strains resulting from the strain optimization algorithms.

Each of these sets of reactions is used to create a network also including the metabolites involved in those reactions. This typically creates several independent modules not connected by any metabolite. This network can be visualized using, for instance, the Cytoscape tool (`http://www.cytoscape.org`). This allows to color nodes differently highlighting important nodes that are decision points or typical knockouts identified by the strain optimization algorithms.

In some cases, the networks obtained are manually modified to include nodes and data which could not be identified by purely computationally process but which are nonetheless important for the analysis. The final result is a network that contains the parts of the metabolism which are more commonly altered when the organism is manipulated to produce the metabolite of interest. This network is named the *variation network*.

The last step of our methodology is the analysis of the variation network. This analysis was based in the observation of each module to determine how its existence relates with the knockouts conducted in the mutants, to the production of the target metabolite and to the production of biomass precursors.

During the analysis, the variation network was also compared with pathway maps obtained from KEGG and EcoCyc to determine the relationship of the variations with the organism metabolism as a whole and if they were related with any known important metabolic cycles.

3 Experiments and Results

3.1 Case Study and Experimental Setup

The implementation of the proposed algorithms was performed by the authors in *Java*, within the OptFlux open-source ME platform (http://www.optflux.org) [8]. Some of the methods in network analysis have been added as a plug-in for this platform and the workflow described here will be added in the future.

The case study considered uses the microorganism *Escherichia coli* and the aim is to produce succinate with glucose as the limiting substrate. Succinate is one of the key intermediates in cellular metabolism and therefore an important case study for ME [5]. It has been used to synthesize polymers, as additives and flavouring agents in foods, supplements for pharmaceuticals, or surfactants.

The genome-scale model used is given in [7] and includes a total of $N = 1075$ fluxes and $M = 761$ metabolites. A number of pre-processing steps were conducted to simplify the model and reduce the number of targets for the optimization (see [9] for details) leaving the simplified model with $N = 550$ and $M = 332$; 227 essential reactions are identified, leaving 323 variables to be considered when performing strain optimization.

In this study, we have used both EA and SA executing each algorithm for 30 runs. Each solution has 6 knockouts, since this was the minimum number of knockouts able to provide high quality solutions and an increase in this value does not provide significant gains. After the preprocessing was done we had a total of 4949 distinct networks from an initial batch of 8018 solutions.

3.2 Results and Discussion

The results of the first stage of the analysis, i.e. the results of the comparison between mutant networks and the wild type is provided in a spreadsheet provided as supplementary material in http://darwin.di.uminho.pt/prib2011.

After conducting the variation analysis, the final sub-network was composed by three apparently independent modules. Subsequent observations revealed that these were not in fact independent changes but in fact were all directly or indirectly related to a common set of alterations which occur in practically all mutant organisms. To show the results and address the discussion we will analyse these three modules in more detail. Figure 1 shows the meaning of the colour code used in the following figures to identify the different nodes coming from the analysis.

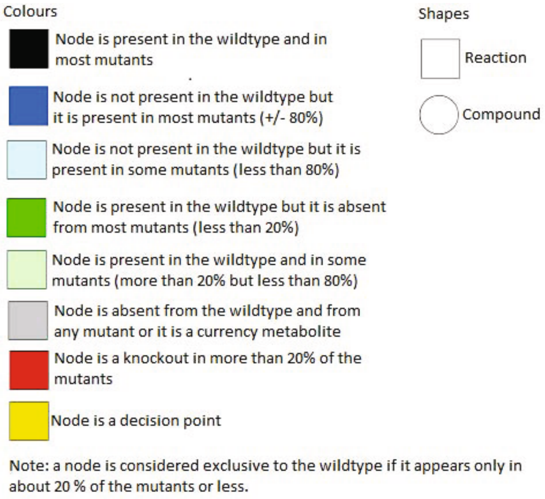
Colours
Node is present in the wildtype and in most mutants
Node is not present in the wildtype but it is present in most mutants (+/- 80%)
Node is not present in the wildtype but it is present in some mutants (less than 80%)
Node is present in the wildtype but it is absent from most mutants (less than 20%)
Node is present in the wildtype and in some mutants (more than 20% but less than 80%)
Node is absent from the wildtype and from any mutant or it is a currency metabolite
Node is a knockout in more than 20% of the mutants
Node is a decision point
Note: a node is considered exclusive to the wildtype if it appears only in about 20 % of the mutants or less.
Shapes
Reaction
Compound

Fig. 1. Colour code used in the analysis of the variation network.

Main Knockouts and Succinate Production. This module, shown in Figure 2, is particularly interesting because it contains two of the most frequently knocked out reactions (SUCD4 and SUCD1i) and it is directly related with the production of succinate. Of all the modules in the variation network this is the one whose analysis is the most straightforward, since the reaction SUCD1i, central of this module, is the only consumer of succinate in the wild type. Thus, to achieve the production of succinate, the most direct method is to remove it. SUCD1i is one of the reactions with the higher value of wild type exclusivity and also of frequency of selection as a knockout in the solutions. SUCD1i is also the first fumarate production reaction mentioned in the dGDP Consumption

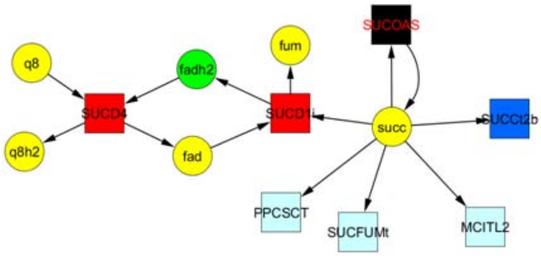

Fig. 2. Sub-network for the main knockouts in succinate production

Module (shown next). Its indirect effect on the former module illustrates how alterations in the metabolism can have unexpected effects.

Besides the direct removal of SUCD1i, another simple way of suppressing this reaction is the knockout of SUCD4. This reaction is the main way in which FADH2 is consumed and SUCD1i is the only producer of FADH2. The removal of SUCD4 will necessarily lead to the suppression of SUCD1i.

Another reaction of note is the transporter SUCCt2b which excretes succinate from the cell naturally. This is a mutant exclusive reaction for obvious reasons, since in the wild type no succinate is excreted. There are other reactions that only occur in the mutants, but the only one that appears in a significant number of mutants is SUCOAS. This reaction is also present in the wild type in its reversed form.

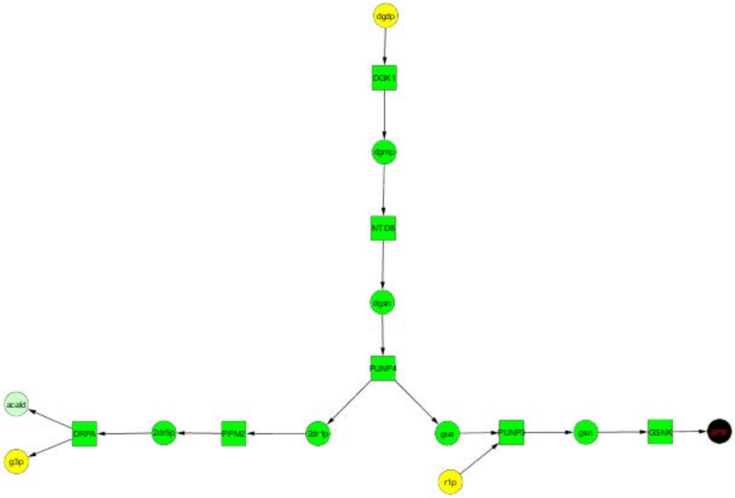

Fig. 3. Sub-network for dGDP consumption

dGDP Consumption. Originally, this module (Figure 3) appeared as two independent reaction chains exclusive to the wild type:

Guanine (gua) + alpha-D-Ribose 1-phosphate (r1p) → PUNP3 → Guanosine (gsn) → GSNK

2-Deoxy-D-ribose 1-phosphate (2dr1p) → PPM2 → 2-Deoxy-D-ribose 5-phosphate (2dr5p) → DRPA

The analysis of the variation network revealed that these two chains were actually a consequence of the wild exclusive chain:

dGDP (dgdp) → DGK1 → dGMP (dgmp) → NTD8 → Deoxyguanosine (dgsn) → PUNP4 → Guanine (gua) + Deoxy-D-ribose 1-phosphate (2dr1p)

This chain was not initially identified because its value of exclusivity was slightly below the defined threshold. It was determined that this module is wild type exclusive because the reaction DGK1 rarely appears on mutants. However, neither DGK1 or any of the reactions which produce the metabolites it consumes is a common knockout, which means that some alteration in the mutants' metabolism provokes the redirection of the flux of some of the compounds used by DGK1. Initial observations of the flux values of the reactions which compose this cycle and their immediate neighbors revealed that the reason for its wild type exclusivity is a consequence of the significant reduction of dGDP in the mutants. This module does not produce any essential metabolite that can not be obtained by other reactions. Also, dGDP is necessary for the production of dGTP which is a biomass precursor. The reduction of the concentration of dGDP leads to all dGDP being channelled to the production of dGTP to ensure the survival an growth of the organism.

A more thorough analysis of the reactions revealed that the reduction of the dGDP production in the mutants is ultimately due to the reduced production of a compound used in its synthesis: 3-phosphohydroxypyruvate. This compound is obtained from a reaction which uses 3-phospho-D-glycerate as a substrate, while the production of 3-phospho-D-glycerate is not being reduced in the mutants (in fact it is somewhat increasing). Most of it is being used to produce D-glycerate 2-phosphate, which in turn is used for the production of phosphoenolpyruvate.

Continuing the analysis of the reactions which use phosphoenolpyruvate, it was determined that the increased production of phosphoenolpyruvate in the mutants is a consequence of the change in the TCA cycle due to a reduction in the production of L-malate. This, in turn, leads to a need for an increase of the production of phosphoenolpyruvate in order to maintain the cycle.

The alterations in the production of L-malate are due to the reduction of the production of fumarate, a metabolite used by a reaction external to the TCA cycle which produces L-malate. Its reduction is a consequence of the flux reduction of the two major fumarate production reactions in the mutant:

1. The main fumarate production reaction is also the main succinate consuming reaction; since the objective is to maximize succinate production this means that this reactions is a frequent knockout and even when it is not, it tends to be inactive.

2. The other reactions which produce fumarate compete with for the consumption of L-Aspartate 4-semialdehyde with the alternative L-threonine production chain which how it will shown later is essential for the survival of most mutants,

It is interesting to note that the flux reduction of the fumarate production reactions is related with two other modules which indicates that the modules of the variation network are not as independent as the initial observations of the network implied.

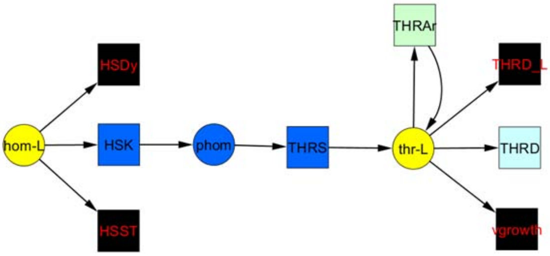

Fig. 4. Sub-network for Alternative L-threonine production

Alternative L-Threonine Production. This module, shown in Figure 4, is characterized by a mutant exclusive reaction chain:

HSK → O-Phospho-L-homoserine (phom) → THRS → L-Threonine (thr-L)

Our analysis revealed that the reason for this chain being mutant exclusive lies in the reaction THRAr which normally produces L-Threonine, which is inverted in most mutants. This chains is necessary to compensate this inversion. The inversion of reaction THRAr occurs because in the inverted form it produces glycine and the alternative reaction for the production of glycine (GHMT2) is wild type exclusive.

Initially, the fact that GHMT2 was wild type exclusive appeared strange. However, we eventually concluded that this reaction is a major producer of nadph and its removal forces the mutants to compensate by using a reaction which produces succinate as a byproduct, thus increasing the production of succinate. This fact was difficult to determine because nadph is a currency metabolite removed from the networks in the pre-processing stages.

It should be noted that the second fumarate production reaction mentioned in the dGDP Consumption Module is the producer of L-Homoserine which is at the beginning of the reaction chain central to this module. Again, this shows the unity of the metabolism and it gives further evidence to the idea that the variations are not distinct modules but a closely related group of metabolic changes.

4 Conclusions

The recent development of computational methods for strain optimization based on the use of genome-scale metabolic models has opened new avenues for

Metabolic Engineering. This work aims to contribute to this effort proposing computational methods for the analysis of the solutions of strain optimization metaheuristics, such as EAs and SA. The aim is to provide tools that allow to identify the most common patterns used by successful mutant strains and therefore understand the strategies used, prior to wet lab experiments that will ultimately validate the results.

Further work will address the full implementation of these features in the OptFlux platform, making them available to the community. Also, the work will be extended to other interesting case studies in ME, by considering other metrics of the network topology and by improving the analysis methodology so that it takes partial flux variations between the mutants and the wild type into account.

Acknowledgments. This work is supported by project PTDC/EIA-EIA/115176/2009, funded by Portuguese FCT and Programa COMPETE. JPP work is funded by a PhD grant from the Portuguese FCT (ref. SFRH/BD/41763/2007).

References

1. Feist, A.M., Herrgard, M.J., Thiele, I., Reed, J.L., Palsson, B.Ø.: Reconstruction of biochemical networks in microorganisms. Nature Reviews Microbiology 7(2), 129 (2008)
2. Feist, A.M., Palsson, B.Ø.: Nature Biotechnology 26(6), 659–667 (2008)
3. Ibarra, R.U., Edwards, J.S., Palsson, B.G.: Escherichia coli k-12 undergoes adaptive evolution to achieve in silico predicted optimal growth. Nature 420, 186–189 (2002)
4. Kauffman, K.J., Prakash, P., Edwards, J.S.: Advances in flux balance analysis. Curr. Opin. Biotechnol. 14, 491–496 (2003)
5. Lee, S.Y., Hong, S.H., Moon, S.Y.: In silico metabolic pathway analysis and design: succinic acid production by metabolically engineered escherichia coli as an example. Genome Informatics 13, 214–223 (2002)
6. Patil, K., Rocha, I., Forster, J., Nielsen, J.: Evolutionary programming as a platform for in silico metabolic engineering. BMC Bioinformatics 6(308) (2005)
7. Reed, J.L., Vo, T.D., Schilling, C.H., Palsson, B.O.: An expanded genome-scale model of escherichia coli k-12 (ijr904 gsm/gpr). Genome Biology 4(9), R54.1–R54.12 (2003)
8. Rocha, I., Maia, P., Evangelista, P., Vilaça, P., Soares, S., Pinto, J.P., Nielsen, J., Patil, K.R., Ferreira, E.C., Rocha, M.: Optflux: an open-source software platform for in silico metabolic engineering. BMC Systems Biology 4(45) (2010)
9. Rocha, M., Maia, P., Mendes, R., Pinto, J.P., Ferreira, E.C., Nielsen, J., Patil, K.R., Rocha, I.: Natural computation meta-heuristics for the in silico optimization of microbial strains. BMC Bioinformatics 9 (2008)