Computational tools for strain optimization by tuning the optimal level of gene expression

Emanuel Gonçalves, Isabel Rocha, and Miguel Rocha

Abstract In this work, a plug-in for the OptFlux Metabolic Engineering platform is presented, implementing methods that allow the identification of sets of genes to over/under express, relatively to their wild type levels. The optimization methods used are Simulated Annealing and Evolutionary Algorithms, working with a novel representation and operators. This overcomes the limitations of previous approaches based solely on gene knockouts, bringing new avenues for Biotechnology, fostering the discovery of genetic manipulations able to increase the production of certain compounds using a host microbe. The plug-in is made freely available together with appropriate documentation.

1 Introduction

To reach biological ways of production for compounds of industrial interest, the metabolism of the selected host microbe usually needs to be modified. Metabolic Engineering (ME) is an essential field addressing these issues [1], being one of its main tasks to identify gene manipulations that increase production yields. To attain this goal, there is the need to develop accurate and efficient computational methods at three different levels: reconstruction of genome-scale metabolic models (GSMs), phenotype simulation and strain optimization. Currently, several GSMs containing information regarding the microbes' metabolism and the transcriptional information are available, for several microbes (e.g. *Escherichia coli* [2]).

Gonçalves, E. and Rocha, M.
CCTC, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal, e-mail: pg17598@di.uminho.pt,mrocha@di.uminho.pt

Rocha, I.
CEB / IBB, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal, e-mail: irocha@deb.uminho.pt

One popular method to simulate the phenotype of cells, assuming a pseudo-steady state, is Flux Balance Analysis (FBA) [3]. FBA aims to maximize cell growth, by solving a linear programming problem, subject to stoichiometric and reversibility constraints. Other approaches have been suggested, such as parsimonious enzyme usage FBA [4].

A bi-level strain optimization problem can be formulated assuming an inner-layer, the phenotype simulation method, and an outer-layer, optimization algorithms that search for the best set of genetic modifications to apply to a microbe. Several methods based on this structure were suggested, such as OptKnock [5] which aims to identify a set of reactions to be deleted from the metabolic model. Evolutionary Algorithms (EA) have been proposed as an alternative, providing methods such as OptGene [6], identifying near optimal solutions for large problems in a reasonable amount of time. Recent work by the authors' research group proposed alternatives based on Simulated Annealing (SA) and a variable size set-based solution representation enabling the identification of the optimal number of reaction deletions [7]. Also, they proposed methods that use the transcriptional/translational information available in some GSMs, allowing the prediction of a set of gene knockouts, instead of reaction deletions [8].

Another approach was suggested in OptReg [9] that aims to overcome one of the major limitations of previous methods, which restrict modifications to gene knockouts. OptReg works with reaction under/overexpression, where a reaction is over (under) expressed if it has a flux considerably higher (lower) when compared to the wild type fluxes. OptKnock and OptReg have some limitations, mainly due to the computational burden imposed by the mixed integer linear programming formulation, as well as the restriction that imposes the linearity of the objective functions. EAs and SA have proven to address well these issues in previous work [7].

In this work, the focus is in the development of computational tools for ME that allow to work with sets of over/under expressed genes. These tools provide the implementation of *(i)* methods for phenotype simulation of under and overexpression mutant strains and *(ii)* strain optimization algorithms based on SA and EA allowing to reach sets of genes to over/underexpress optimizing the production of a specific target compound. These tools are integrated as a plug-in within the OptFlux platform [10], being validated with a simple model for consistency check and a real case with *E. coli* as a host for the production of succinic and lactic acid.

2 Methods

2.1 Phenotype simulation with over/under expressed genes

Flux Balance Analysis (FBA) [3] depicts the mass balance of all internal metabolites as linear equations. It assumes that the sum of all consumptions and productions is null. Therefore, for M metabolites and N reactions we have the following constraints: $\sum_{j=1}^{N} S_{ij}v_j = 0, i = 1,\ldots,M$. Other restrictions (reversibility, thermodynamics, capacity constraints) are represented as inequalities: $\alpha_j \leq v_j \leq \beta_j, j = 1,\ldots,N$. Using linear programming, FBA maximizes a specific optimization function, usually the biomass formation, subject to the previous restrictions.

In this work, we use the parsimonious enzyme consumption FBA (pFBA) method [4]. The pFBA firstly runs a FBA simulation optimizing the biomass flux value. Afterwards, it adds a constraint which forces the biomass flux value to be equal to the optimal value found in the previous step. Finally, a new optimization problem is formulated minimizing the sum of all fluxes and it is solved. This method tends to predict better the phenotype of the microbe [4].

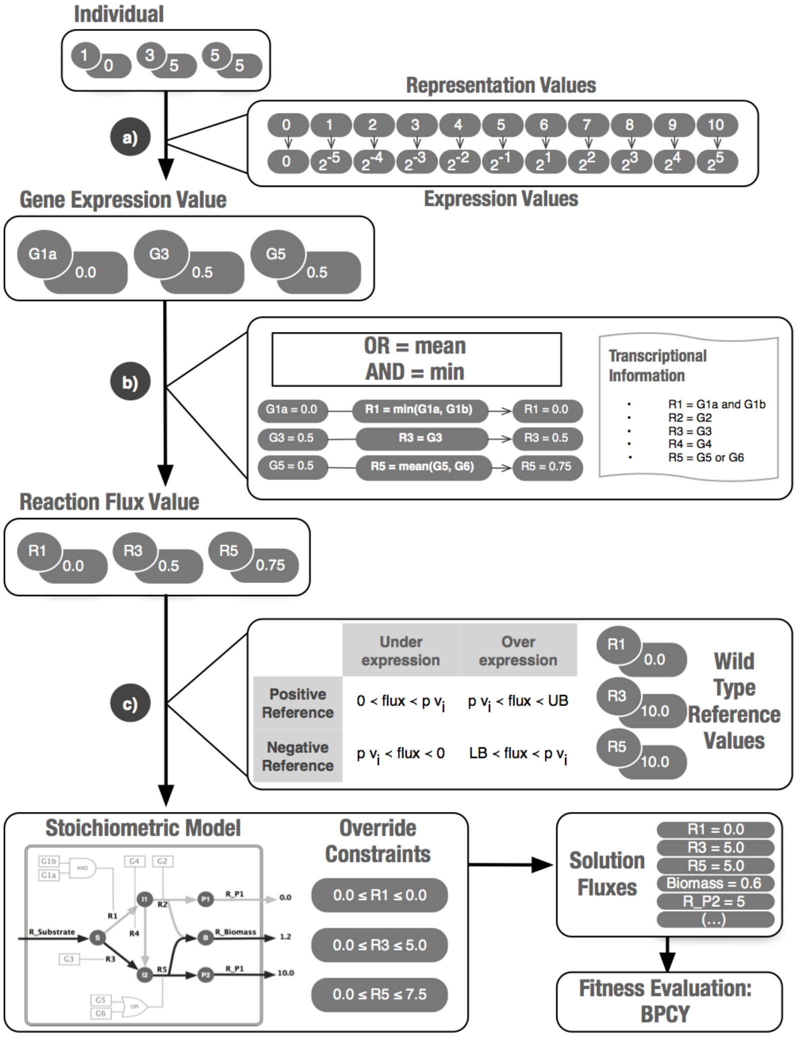
The simulation of the mutant has as input a list of genes and associated expression values, relating the gene expression to its level in the wild type. If the expression value is 0, it encodes a gene deletion, if it is less than 1, it deems a gene underexpression; a value higher than 1 is an overexpression. Since these are gene expression values, they need to be propagated to reaction flux values. The simulation process is made in two steps: *(i)* using the transcriptional/translational information, the gene expression values are decoded into constraints over reactions fluxes (the process of decoding is depicted in Figure 1b); *(ii)* these constraints are then added to the pFBA optimization problem and solved as usual (Figure 1c).

Regarding *(i)*, transcriptional/translational information is given in the form of gene-reaction rules represented as Boolean expressions. Since we are working with numerical values of expression, a strategy to convert the Boolean operators to numerical values is needed. The AND operator is transformed into the minimum function and the OR operator is the average. To generate the constraints to add to the linear programming problem in pFBA, a reference set of flux values is provided by the simulation of the wild type strain (also with pFBA). If the relative value of a given reaction is equal to one, no constraints are added. On the other hand, if the value is larger (smaller) than 1, a constraint is added forcing the flux to be higher (lower) than the wild type reference value (see Figure 1 c).

2.2 *Strain optimization*

In the representation adopted, each solution is composed by a list of pairs, each made by a gene index and the respective expression level (two integers). The expression level encodes a series of possible expression values. A discrete representation is used, greatly reducing the solution space and, therefore, the complexity of the optimization task.

To fairly distribute the discrete values of expression between under and overexpression and not disregarding knockouts, we defined a logarithmic scale containing the values: $0, 2^{-n}, 2^{-n+1}, \ldots, 2^{-1}, 2, 2^2, \ldots, 2^{n-1}, 2^n$. The parameter n will define the maximum (minimum) range of overexpression (underexpression). As depicted in Figure 1, after decoding the gene expression values to reaction flux levels and applying the constraints to the model, a pFBA simulation is run and returns the fluxes



	Under expression	Over expression
Positive Reference	$0 < flux < p\,v_i$	$p\,v_i < flux < UB$
Negative Reference	$p\,v_i < flux < 0$	$LB < flux < p\,v_i$

Fig. 1: Global scheme of the solution's decoding and evaluation: a) shows the decoding of the expression index to real values; b) shows the transformation of gene expression values to reaction flux values, using the transcriptional information available in the model; c) the under and overexpression constraints are calculated based on the reference values, and then applied as override constraints to the model; the simulation of the mutant is run and the reaction fluxes are sent to the BPCY evaluation function to measure the quality of the solution.

of the reactions in the model. To measure the quality of the solution, we use an objective function termed as Biomass-Product Coupled Yield (BPCY) [6].

In this work, EAs and SA were used as the optimization methods. These are similar to the ones previously proposed by the authors research group for knockout optimization [7]. The major change in these algorithms was the novel representation presented above. Six reproduction operators were used: *(i)* an operator based on the uniform crossover, which works as follows: the genes regulated on both parent genomes are sent to one of the offspring; the others are randomly sent to the offspring; *(ii)* two random mutation operators that randomly modify the gene and the expression value indexes; *(iii)* a mutation operator which changes the expression value index of a pair by the next or previous value; *(iv)* to generate solutions with different sizes two mutations operators: the shrink mutation operator removes a selected pair and the grow mutation operator adds a new pair.

3 Implementation

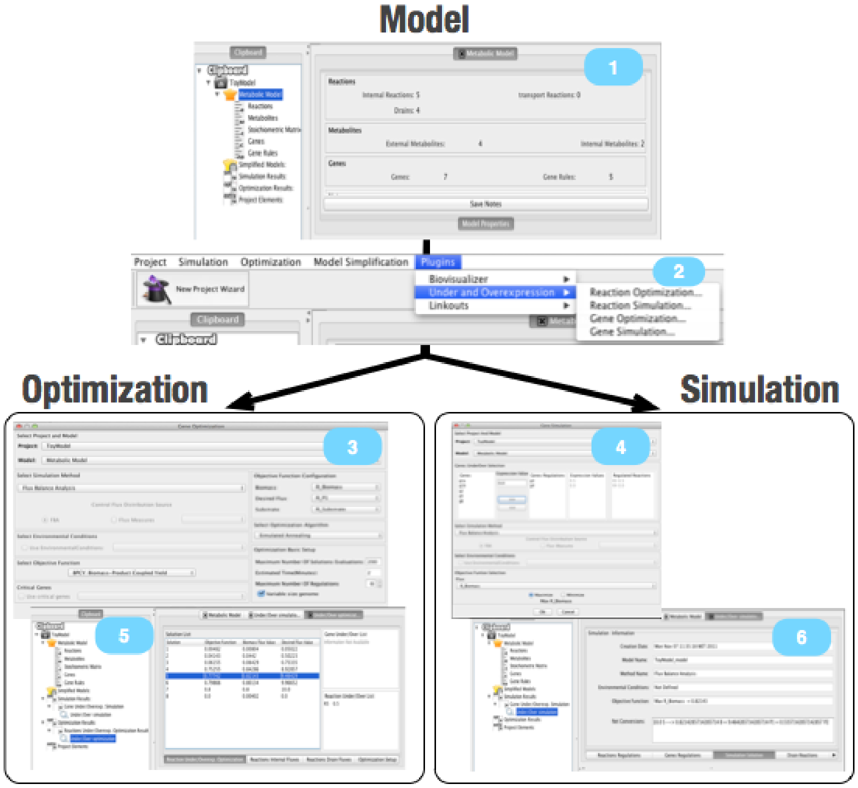
The framework proposed in this work was developed in JAVA within the OptFlux platform (`http://www.optflux.org`) [10]. OptFlux is an open source ME platform with an user-friendly interface. It has a modular structure, enabling the possibility to easily develop plug-ins adding new features to the application. OptFlux includes operations to run *in silico* phenotype simulations of the wild type or mutant strains using several methods. Users can also perform strain optimization selecting gene knockouts or reaction deletions. The software supports importing/exporting GSMs in several file formats, such as flat files and standard SBML.

OptFlux interface is built on top of AIBench (`http://www.aibench.org`) [11], using it to help producing GUI components such as the clipboard, the panels and views. AIBench is a Java framework that eases the development of applications based in the Input-Processing-Ouput paradigm, being developed according to the MVC (Model-View-Controller) strategy.

The plug-in proposed enables OptFlux users to perform operations involving mutants with a set of genes that are over/underexpressed. The user can perform the phenotype simulation of mutant strains specifying a set of genes and their relative expression levels or a set of reactions with relative flux levels. The methods available for the simulation are the same provided for knockouts, including the FBA and pFBA described above, among others.

Users have the possibility to perform strain optimization to find both sets of genes or reactions and their relative levels of expression. Both EA and SA are available to be selected as optimization methods and the user can configure the objective function, the termination criterion (e.g. maximum number of solutions evaluated) and other parameters. Also, the target compound and the biomass reaction need to be defined.

The plug-in is available for download through the OptFlux web site. Documentation is also available regarding the plug-in, with a detailed How-to where the main steps to configure and perform the operations are depicted. Figure 2 shows the configuration panels of the simulation and optimization operations, providing also an example of the views used to present the output solutions.



Reactions
Internal Reactions: 5 transport Reactions: 0
Drains: 4
Metabolites
External Metabolites: 4 Internal Metabolites: 2
Genes
Genes: 7 Gene Rules: 5

Fig. 2: A summary of the panels and menus of the proposed plug-in. Panel 1 shows the model loaded on OptFlux; through the menu bar in 2 the user can chose to simulate the mutant phenotype or to run strain optimization; 3 depicts the menu for configuration of the optimization, while 5 shows the results; simulation is made through the menu in 4, where the user can choose each gene and the expression level (the framework previews the constraints to be applied); the simulation results appear in a panel shown in 6.

4 Case studies

The validation process for this software, in a first step, has been done using a small model containing transcriptional/translational information [12]. The model is portrayed in Figure 3 a), showing the fluxes of the wild type with an FBA simulation, where it produces P2 and does not produce P1. Therefore, in order to validate our framework we defined as objective function the maximization of the production of P1. Regarding this problem, it is easily perceptible that two solutions may be found: the knockout of G3 and G4 and therefore the deletion of R3 and R4; or the deletion of G5 and G6 that leads to the deletion of R5 (both gene deletions are needed since they regulate R5 by an OR operator). The drain reactions (R_P1, R_P2, R_Biomass and R_Substrate) were set to be critical and by that means impossible to be regulated. We ran the framework for this problem with a number of function evaluations enough to assure that a good solution was found. Easily, the framework found a solution producing the maximum level of P1, as desired, suggesting the first solution mentioned: the deletion of G3 and G4 (Figure 3 b).

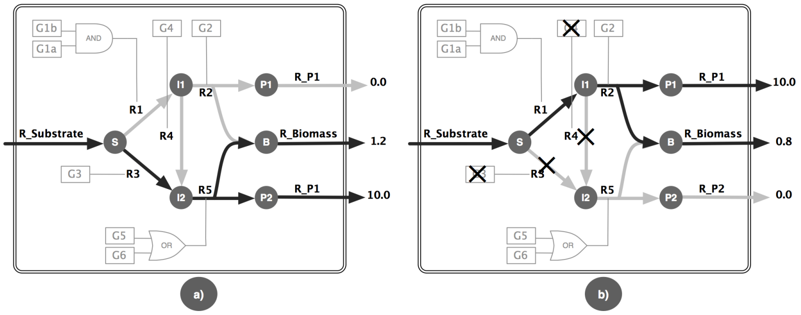


Fig. 3: Toy model from [12] showing the wild type flux distribution (flux of P2 = 10 and biomass flux = 1.2), a). The second picture, b), shows the mutant simulation with the best solution suggested by the framework optimizing the production of P1. R1 and R2 are now active leading to the maximum production of P1 (10) and to a flux 0.8 in biomass.

This framework was tested with a larger case study, involving the production of succinic and lactic acid in *Escherichia coli*. The model used is from [2] and it contains 1075 reactions, 761 metabolites and 904 genes. The obtained results are presented elsewhere [13] and are considerably better when compared to the strain optimization based on gene knockouts, being the difference substantially larger with lactic acid production. Another important result is the fact of the under and over-expression solutions are, in general, quite small, increasing the feasibility of the genetic manipulations suggested.

5 Conclusion

Strain optimization based on gene knockouts is widely used in Biotechnology, since it can greatly reduce production costs. Here, a framework is proposed based on gene/reaction under/overexpression that overcomes the knockout optimization limitations. Thus, this work expands the available set of computational tools available for ME experts, being well integrated within a broad platform such as OptFlux.

Future work will focus on further validation of the framework with other case studies. An improvement to the handling of constraints regarding reactions with zero flux in the wild type will be considered. Also, existing tools to analyse the solutions *in silico* will be improved.

Acknowledgements

Support of FCT and Programa COMPETE (ref. PTDC/EIA-EIA/115176/2009)

References

1. G. Stephanopoulos, A. Aristidou, and J. Nielsen, Metabolic engineering. Acad. Press, 1998.
2. J. Reed, T. Vo, C. Schilling, and B. Palsson, “An expanded genome-scale model of Escherichia coli K-12 (iJR904 GSM/GPR).,” Genome biology, vol. 4, p. R54, Jan. 2003.
3. J. Edwards and M. Covert, “Minireview Metabolic modelling of microbes : the flux-balance approach,” Environmental Microbiology, vol. 4, pp. 133–140, 2002.
4. N. Lewis, K. Hixson, Conrad, et al., “Omic data from evolved E. coli are consistent with computed optimal growth from genome-scale models.,” Molec Syst Biol, vol. 6, no. 390, 2010.
5. A. Burgard, P. Pharkya, and C. Maranas, “Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization.,” Biotechnology and bioengineering, vol. 84, pp. 647–57, Dec. 2003.
6. K. Patil, I. Rocha, J. Förster, and J. Nielsen, “Evolutionary programming as a platform for in silico metabolic engineering.,” BMC Bioinformatics, vol. 6, no. 308, 2005.
7. M. Rocha, P. Maia, R. Mendes, et al., “Natural computation meta-heuristics for the in silico optimization of microbial strains.,” BMC bioinformatics, vol. 9, no. 499, 2008.
8. P. Vilaça, P. Maia, I. Rocha, and M. Rocha, “Metaheuristics for Strain Optimization Using Transcriptional Information Enriches Metabolic Models,” Proc. EvoBio 2010, LNCS, 2010.
9. P. Pharkya and C. Maranas, “An optimization framework for identifying reaction activation/inhibition or elimination candidates for overproduction in microbial systems.,” Metabolic engineering, vol. 8, pp. 1–13, Jan. 2006.
10. I. Rocha, P. Maia, P. Evangelista, P. Vilaça, S. Soares, et al., “OptFlux : an open-source software platform for in silico metabolic engineering,” BMC Systems Biology, 2010.
11. D. Glez-Peña, M. Reboiro-Jato, P. Maia, M. Rocha, F. Dìaz, and F. Fdez-Riverola, “AIBench: a rapid application development framework for translational research in biomedicine,” Computer Methods and Programs in Biomedicine, vol. 98, pp. 191–203, 2010.
12. J. Kim and J. Reed, “OptORF: Optimal metabolic and regulatory perturbations for metabolic engineering of microbial strains.,” BMC systems biology, vol. 4, p. 53, Jan. 2010.
13. E. Gonçalves, R. Pereira, I. Rocha, et al., “Optimization approaches for the in silico discovery of optimal targets for gene over/underexpression,” J Computational Biology, in press.