



A computational tool for the simulation and optimization of microbial strains accounting integrated metabolic/regulatory information

Paulo Vilaça^{a,b,*}, Isabel Rocha^a, Miguel Rocha^b

^a IBB-Institute for Biotechnology and Bioengineering/Centre of Biological Engineering, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal

^b Department of Informatics/CCTC, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal

ARTICLE INFO

Article history:

Received 11 August 2010

Received in revised form

14 November 2010

Accepted 26 November 2010

Keywords:

Metabolic engineering

Integrated models

Metabolic models

Regulatory models

Software tool

Open-source

ABSTRACT

Background and scope: Recently, a number of methods and tools have been proposed to allow the use of genome-scale metabolic models for the phenotype simulation and optimization of microbial strains, within the field of Metabolic Engineering (ME). One of the limitations of most of these algorithms and tools is the fact that only metabolic information is taken into account, disregarding knowledge on regulatory events.

Implementation and performances: This work proposes a novel software tool that implements methods for the phenotype simulation and optimization of microbial strains using integrated models, encompassing both metabolic and regulatory information. This tool is developed as a plug-in that runs over OptFlux, a computational platform that aims to be a reference tool for the ME community.

Availability: The plug-in is made available in the OptFlux web site (www.optflux.org) together with examples and documentation.

© 2010 Elsevier Ireland Ltd. All rights reserved.

1. Introduction

Over the last few years, the combined efforts of Metabolic Engineering (ME) and Systems Biology (SB) allowed the reconstruction of genome-scale metabolic and regulatory models for some organisms with an industrial interest. These models, together with appropriate simulation methods, allowed the development of computational approaches for finding suitable genetic modifications for specific applications. The aim is to make the microorganisms fit to comply with industrial purposes, i.e. to be able to synthesize some desired compounds in significant amounts, rather than to follow their natural aims.

The majority of the mathematical and computational approaches to perform simulations with metabolic models are based on steady-state approximations, assuming that concentrations of internal metabolites do not change over time. In this scenario, it is possible to determine the space of possible flux distributions (i.e. the set of feasible flux vectors), using constraint-based metabolic models. These models are constructed based on the known stoichiometry of the metabolic reactions, thermodynamic constraints and flux capacities. The most popular constraint-based

approach is Flux Balance Analysis (FBA) (Kauffman et al., 2003) that uses Linear Programming (LP) to reach an optimal distribution of flux values for all the reactions. In its most widely used formulation, FBA predicts cellular behavior assuming the organism maximizes its biomass production rate, an assumption that was shown to successfully predict various metabolic phenotypes (Ibarra et al., 2002). Some alternative constraint-based approaches have been developed for the simulation of the phenotype of mutant microorganisms: MOMA (Minimization of Metabolic Adjustment) (Segrè et al., 2002) or ROOM (Regulatory On/Off Minimization of metabolic fluxes) (Shlomi et al., 2005).

On the other hand, Boolean approaches have been used to create regulatory models. Boolean networks approximate the dynamics of the regulatory network by considering that each node in the network (i.e. gene, stimulus) is in a binary state: active or inactive. The value of each node is assigned by a Boolean update rule, depending on the value of some other nodes. Those models were used to understand regulatory interactions (Kauffman et al., 2004) and also to simulate the behavior of the system under given genetic/environmental conditions (Li et al., 2006). Although metabolic and regulatory systems are known to be highly dependent, only a few studies have focused on the analysis of integrated models, for example by incorporating regulatory information as additional constraints of genome-scale metabolic models (Covert et al., 2004).

Some methods were also developed to allow the phenotype simulation with such integrated models. One example is rFBA (Regulatory Flux Balance Analysis), which simulates growth in batch

* Corresponding author at: IBB-Institute for Biotechnology and Bioengineering/Centre of Biological Engineering, University of Minho, 4710-057 Campus de Gualtar, Braga, Portugal.

E-mail addresses: pvilaca@deb.uminho.pt (P. Vilaça), irocha@deb.uminho.pt (I. Rocha), mrocha@di.uminho.pt (M. Rocha).

cultures by predicting dynamic flux profiles in changing environments (Covert et al., 2001). This method works by predicting a regulatory and a metabolic steady state for short successive time intervals. For each step, the method computes a regulatory state that is consistent with the metabolic steady state of the previous interval. Then, FBA is used to find a steady-state flux distribution taking as input the regulatory state of the current time interval and the process is further iterated.

On the other hand, SR-FBA (Shlomi et al., 2007) (Steady-state Regulatory Flux Balance Analysis) is a method for simulating a metabolic-regulatory steady state, assuming that the organism maximizes its biomass production rate as in FBA. The integrated state satisfies both metabolic and regulatory constraints. This method is based on a Mixed Integer Linear Programming (MILP) formulation, since it translates the Boolean logic underlying regulatory constraints and the mapping between genes and reactions to linear equations, integrating them as additional constraints on top of the metabolic model.

The metabolic phenotype simulation methods described above 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. The idea is to force the microorganisms to synthesize a desired product, while keeping it viable. One of the first efforts to tackle this class of optimization problems was the OptKnock algorithm (Burgard et al., 2003), where MILP is used to identify an optimum set of knock-outs under a metabolic steady-state approximation. However, this algorithm does not allow considering nonlinear objective functions and a considerable computation time is required to reach a solution. OptGene (Patil et al., 2005; Rocha et al., 2008) was proposed as an alternative method, which uses Evolutionary Algorithms (EAs) and Simulated Annealing (SA) for this scenario.

A common limitation of OptKnock and OptGene is the fact they only use metabolic information, determining sets of reactions to be eliminated from the metabolic model, instead of sets of genes to knock-out, which is the real purpose. 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, there are many exceptions, such as isoenzymes, protein complexes, or enzymes that catalyze several reactions.

By adding a transcriptional/translational layer into the metabolic models (Reed et al., 2003), using gene-reaction associations, new phenotype simulation and strain optimization approaches can be implemented (Vilaça et al., 2010). Gene-reaction associations, in a similar way to regulatory interactions, can be based on a Boolean logic representation, where the interactions between reactions and encoding genes are defined using logical operations (AND, OR). These associations are used to determine the reaction(s) that will be inactivated in the model, when a given gene (or set of genes) is knocked out.

Recently, OptORF was proposed as a new strain optimization method (Kim and Reed, 2010) that integrates regulatory and metabolic information, transforming the Boolean gene-reaction and regulatory rules in constraints, in a way similar to SR-FBA. Such as OptKnock, it also uses MILP to identify an optimum set of gene knockouts given a production target. Its limitations are therefore, similar to the ones of OptKnock, since it does not allow nonlinear objective functions and needs a considerable time to compute a solution. Using meta-heuristics such as EAs or SA to perform this task can provide near optimal solutions within a reasonable time and also allows the optimization of nonlinear objective functions. However, given their stochastic nature, these methods do not guarantee to reach optimal solutions and can potentially provide different results in each run.

Some of the methods mentioned above are available in a few software tools that provide some support to the ME community. CellNetAnalyser (Klamt et al., 2007), Cobra toolbox (Becker et al., 2007) and OptFlux (Rocha et al., 2010) are examples of platforms that provide subsets of such tools. Both Cobra and CellNetAnalyser can be used to perform the phenotype simulation of both wild type and mutant strains, under distinct environmental conditions. However, OptFlux, an open-source software platform recently launched by our research group, is the only one able to perform strain optimization using both OptGene and OptKnock methods. It provides a simple user's interface together with a powerful plug-in engine/API, enabling its easy extension. Both OptFlux and COBRA toolbox allow the user to load transcriptional and translational information in the form of gene-reaction associations, but they still do not allow the integration of regulatory information. CellNetAnalyser allows the user to load regulatory models, but it does not have the capacity to integrate the regulatory information with the metabolic models in order to provide phenotype simulation.

To address the identified limitations, we hereby present a novel computational tool, developed as a plug-in for the OptFlux platform, that allows users to load and integrate regulatory models with metabolic models, providing state of the art methods for the phenotype simulation of both wild type and mutant strains, as well as strain optimization algorithms able to work with the integrated models. To the best of the authors' knowledge this is the first computational tool to allow this set of functionalities using integrated models, representing an important step in Metabolic Engineering.

The next sections describe in more detail the functionalities of the tool and provide case studies and examples of its usage.

2. Software description

2.1. Integrated models

The regulatory plug-in allows loading a regulatory model and integrating it with a metabolic model previously loaded with OptFlux. The regulatory models are qualitative, using a Boolean representation. Such models consist on a set of genes and a set of additional Boolean variables or conditions. Genes can be either metabolic, meaning that they encode enzymes, or regulatory, meaning that they encode for proteins involved in regulatory events (such as transcription factors). The regulated genes can be either metabolic or other regulatory genes.

Each regulated gene has an associated Boolean rule, which determines its state from a set of inputs, i.e. a function containing Boolean values of other genes (typically regulatory), as well as other conditions, that may influence the gene expression. The conditions can be of two types: variables identifying perturbations to the system (e.g. stress situations) or variables characterizing the environment, including the presence of external compounds in the media (e.g. oxygen, substrates) or other features (e.g. pH value). The Boolean function can use the following operators: AND, OR, NOT.

The connection between regulatory and metabolic models is conferred on two levels:

- Gene connections, linking the genes present in the metabolic model (in the form of gene-reaction associations that define which genes encode the enzymes connected with a particular reaction) with regulated genes present in the regulatory model;
- Connections between environmental variables in the regulatory model and the external metabolites in the metabolic model. These associate environmental Boolean variables in the regulatory model to the presence/absence of a metabolite in the environment as given by the metabolic model. These variables

can be used in the right hand side of the rules for each gene (metabolic or regulatory).

It is assumed that the identifiers in both models are coincident, in order to allow the previous matches to take place.

Three representation levels define the integrated model:

- Regulatory level, including the Boolean rules in the regulatory model.
- Transcriptional/translational level, including the gene-reaction associations.
- Metabolic level, including reactions, metabolites, reaction stoichiometry, reversibility and flux bounds.

2.2. Methods for phenotype simulation

The proposed tool implements two different phenotype simulation algorithms for regulatory/metabolic models: an approach based on a steady state approximation to rFBA and the previously mentioned SR-FBA. The first algorithm consists of two distinct steps: (i) the regulatory one, which simulates the regulatory component of the biological system, reaching as a result the set of genes that can be expressed, and (ii) the metabolic step, which simulates the constraint-based metabolic model to reach the flux values for all the reactions. This approach is much faster than the SR-FBA simulation, since it relies on two computationally light processes (Boolean network simulation and LP), when compared to MILP, required by SR-FBA.

Given the huge number of possible intermediate states that would result from the Boolean network simulation using asynchronous methods, and since here the interest is in calculating a steady-state approximation, the Boolean simulation is carried out by a synchronous and deterministic Boolean network simulation method (Naldi et al., 2009). Here, the values of all variables are updated simultaneously in every step. Since the method is deterministic, the state will always be the same, if the same initialization is provided. The user can define the initial state for each particular gene by choosing the values manually or loading those from a file. The default option is to consider all gene variables initially to be true, except for the genes that are knocked out (if a mutant strain is being used in the simulation). This default approach is used in the *E. coli* case study shown in the final section, since it has been shown that in this case the final attractor only depends on the values of external stimuli and not on the initial states of the genes (Samal and Jain, 2008).

The variables linked to the metabolic compounds in the environment are initialized using information from the flux bounds of the drain reactions in the metabolic model, i.e. if the bounds defined in the model for the respective exchange reaction allow the uptake of the metabolite, the variable is considered to be true. The regulatory network state is then iterated until it reaches an attractor, which represents a “steady-state” of the regulatory model. In some cases, the simulation can reach “cyclic attractors”, i.e. the system oscillates between two (or more) states. Here, a conservative approach is taken, considering as “off” only the genes that have an “off” value in all states within the cyclic attractors.

In a second step, the set of genes that are set to “off” in the previous step are used to reach a set of reactions that cannot carry a flux, using the transcriptional/translational information contained in the gene-reaction rules. Finally, a simulation method (FBA, MOMA, ROOM) is executed, considering the metabolic model constraints, together with the ones that come from the previous steps. The final result is a steady-state flux distribution consistent with the regulatory state. In Fig. 1 the two steps of the algorithm are schematically shown for a small network.

Section (A) shows a simple regulatory model composed of two transcription factors/regulatory genes (TF_1 and TF_2). TF_1 is activated when both the compounds S and O_2 are present in the environment, activating the expression of the metabolic gene g_1 . TF_2 is activated when the compound O_2 is present in the environment, activating the expression of the metabolic gene g_2 . The transition table shows the next state for all possible initial states. This information is also shown in the graph form, where the three resulting attractors are shown. Section (B) demonstrates how the 2-step integrated simulation works. The integrated model is composed by the regulatory model from (A), a metabolic model with five reactions and seven metabolites, as well as gene-reaction associations for all reactions including six metabolic genes, two of which are also present in the regulatory model (g_1 and g_2). In the first step, the steady-state of the regulatory model is determined, i.e. an attractor is calculated as shown in section A. In this example, there are three possible regulatory steady-states depending on the presence or absence of O_2 and S in the environment. If S is absent, the two genes will not be expressed. If it is present and O_2 is absent, g_2 will be expressed and g_1 will be “off”. If the two compounds are present in the environment both genes (g_1 and g_2) will be expressed. To explain the second step of the method, it is assumed that O_2 is absent and S is present in the environment. Using the gene-reaction associations and assuming that the metabolic genes not included in the regulatory model can be expressed, the inactive reactions are calculated (in this example R_1 is inactive). The flux of this reaction is constrained to be 0 in the simulation of the metabolic model (using FBA, MOMA or ROOM), used to reach the steady-state flux distribution.

The plug-in also implements the aforementioned SR-FBA, an integrated simulation method that employs MILP to identify a consistent regulatory and metabolic steady-state. The regulatory model and the gene-reaction rules are transformed from Boolean rules to linear constraints on the MILP formulation, resorting to the following transformations:

- Boolean expression $a=b$ AND c is formulated as $-1 \leq 2b + 2c - 4a \leq 3$;
- Boolean expression $a = \text{NOT}(b)$ is formulated as $a + b = 1$.

These constraints are added to the stoichiometric, reversibility and flux bound constraints present on the metabolic model.

The output of both simulation methods is a set of Boolean values that represent the expression state of the genes, reached by the regulatory simulation, as well as the flux values for all reactions.

2.3. Algorithms for strain optimization

Regarding the strain optimization tasks, the proposed tool is able to run two optimization meta-heuristics: Simulating Annealing (SA) and Evolutionary Algorithms (EA). These optimization algorithms are similar to the ones previously proposed by the authors (Rocha et al., 2008) and included in OptFlux (Rocha et al., 2010), available for strain optimization using metabolic models. Here, instead of selecting a set of reactions to inactivate, the optimization problem consists in selecting a set of genes to knock out. This will be a subset of all genes in the integrated model (i.e. both metabolic and regulatory genes), that when deleted maximize the production of a desired compound.

Both optimization algorithms can use either of the phenotype simulation methods explained above in the evaluation of the potential solutions, using its output (the flux values) to compute the fitness value, given by an appropriate objective function. It is possible to choose between two alternatives: the Biomass-Product Couple Yield (BPCY) and the Product Yield with Minimum Biomass (PYMB) (Patil et al., 2005; Rocha et al., 2008).

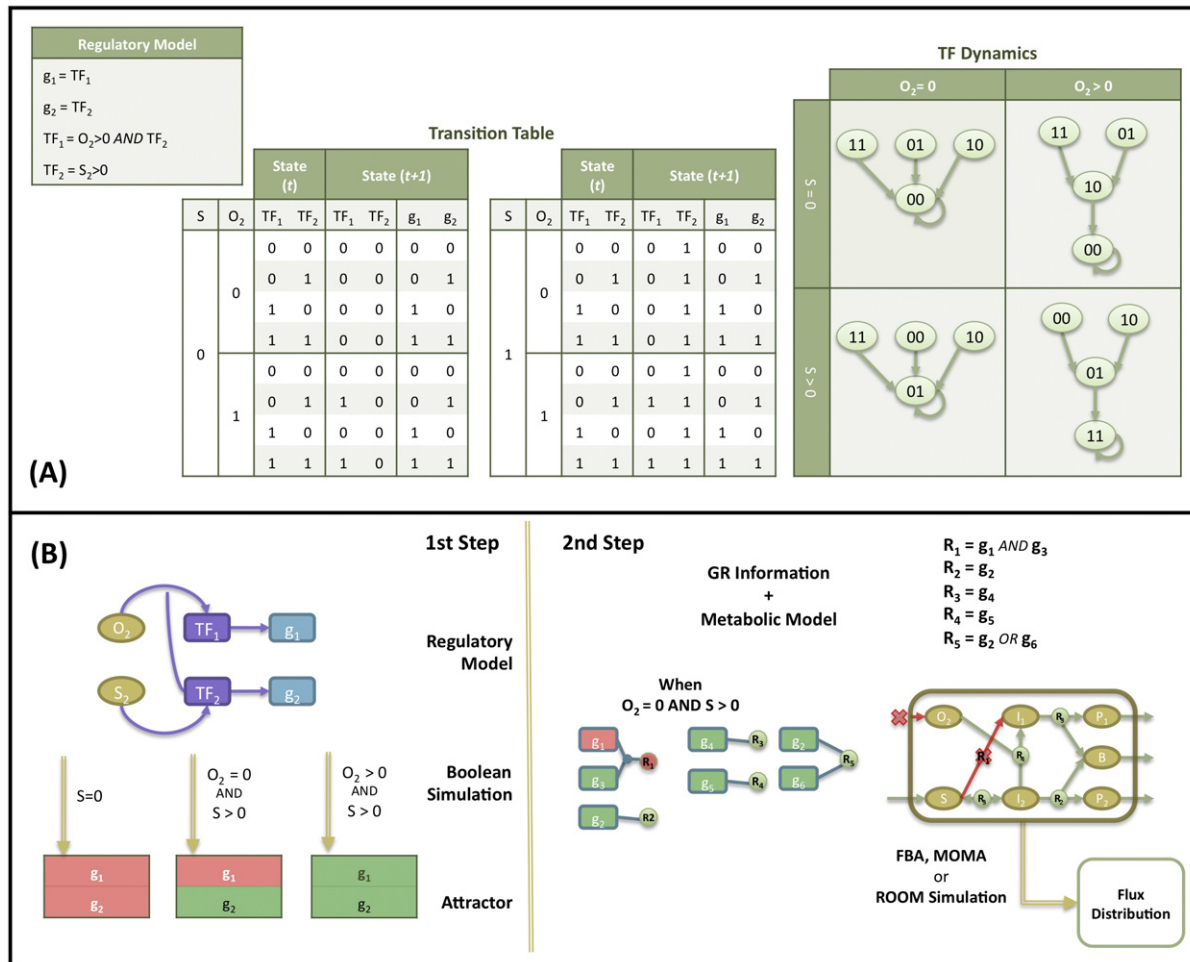


Fig. 1. Phenotype simulation method for integrated models.

In PYMB, the fitness is the value of the flux representing the excretion of the desired product divided by the substrate flux, if the biomass flux value is greater than a user-defined threshold value (a percentage of the wild type value). On the other hand, the fitness value returned by BPCY is given by:

$$BPCY = \frac{P \times B}{S}$$

where P stands for the value of the flux representing the excretion of the desired product; B for the value of the 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.

Since the models account for a significant number of genes, the optimization task is computationally hard. The plug-in, therefore, provides a method for the discovery of essential genes, defined as those that when knocked out, make the organism non-viable (biomass flux near zero), also allowing to manual edit this information and load it from a text file. These data can be used in optimization tasks, not allowing essential genes to be targets for optimization, since they would unnecessarily increase the number of decision variables and therefore the size of the search space. In the proposed software, it is also possible to choose the maximum length of the set of gene knockouts in a solution and also to allow this number to vary during the optimization process or not. An illustration of the structure of both algorithms is given in Fig. 2.

The figure illustrates the main steps of the two strain optimization algorithms EA and SA. Details can be found in Rocha et al. (2008).

2.4. Implementation

OptFlux is a recent open-source platform that aims to be the reference software for the ME community. It was developed to facilitate its use by Biologists/Biotechnologists with no background in programming or command line interfaces. It offers straightforward graphical user interfaces, based on a simplistic workflow (Data–Operation–Data) conferred by being entirely built on top of the *AlBench* framework (Glez-Peña et al., 2010). Its plug-in nature makes easier the development of new components and improves modularity. It integrates three types of objects: *operations*, *datatypes* and *datatype views*, following the MVC (Model–View–Controller) software design pattern, which allows separating the functionalities from the interfaces in the software development process.

This plug-in adds five new functionalities to *OptFlux*, all related to the integration of regulatory and metabolic models:

- **Loading a regulatory network:** loads a regulatory model from a csv file that contains all genes and their regulatory boolean rules, to be integrated with a metabolic model previously loaded;

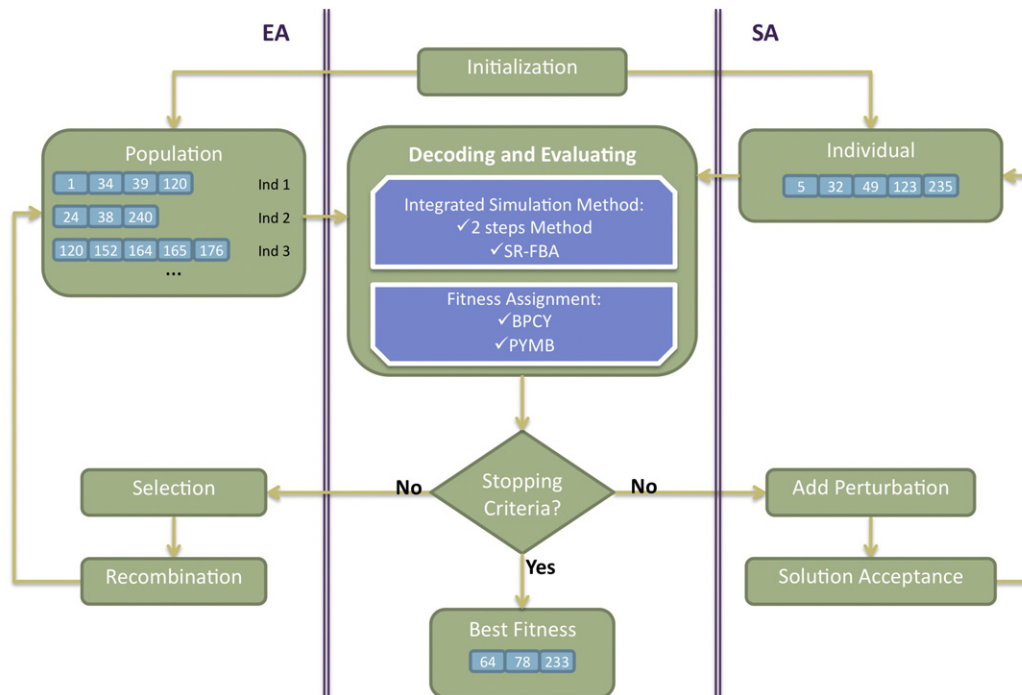


Fig. 2. Structure of the strain optimization algorithms.

- **Simulation:** simulates a wild type or a mutant strain, computing the state of for each gene (i.e., if it is active or not) and for each reaction its flux value. It allows the user to define a set of gene knockouts, the environmental conditions, the simulation method and the values of all user defined variables for the regulatory model (e.g. stress situations, pH value);
- **Load critical genes:** loads a set of essential genes from a file;
- **Compute critical genes:** Computes the set of essential genes for the model;
- **Optimization:** Runs a strain optimization algorithm to identify a set of genes to knockout in order to maximize a given objective function. It allows the user to choose the optimization algorithm (EA or SA) and its parameters, the objective function, the desired product flux, the phenotype simulation method, values of user defined variables in the regulatory model and the environmental conditions.

Several views of these operations are shown in Fig. 3 (the model described in the next section is used in the example).

More information and documentation about the plug-in can be found on OptFlux's website (www.optflux.org/), where there are also examples and a set of relevant How To's.

3. Applications

3.1. Case study: prototype network

To demonstrate the capabilities of this new tool, an example of an integrated metabolic and transcriptional regulatory network is shown in Fig. 4. This is based on an example from Kim and Reed (2010), where the regulatory model is composed by a transcription factor (TF1) that is activated when metabolite S is present, activating the expression of two genes (G3, G5) and repressing the expression of gene G1A. All the other genes are considered expressed by default. In the transcriptional layer, all reactions have a gene-reaction rule associated. R1 is catalyzed when genes G1A and G1B are expressed; R2 is catalyzed by the enzyme encoded by

gene G2; R3 and R4 are catalyzed by the enzymes encoded by genes G3 and G4, respectively; finally, R5 can be catalyzed by enzymes encoded either by genes G5 or G6.

In the metabolic level, the substrate (S) is utilized to produce biomass (B) and by-products P1 and P2. The cellular objective is to maximize biomass production (B) and the engineering objective is the production of P1. Reactions R2 and R5 are the only reactions that can produce biomass. Reaction R2 converts the internal metabolite I1 into product P1 and 0.08 biomass (B), whereas reaction R5 converts the internal metabolite I2 into product P2 and 0.12 Biomass. Without any modification, the model produces preferentially the product P2 and no P1, because reaction R5 produces more biomass than reaction R2, which is the only reaction capable of producing P1. Analyzing the metabolic network, it is possible to conclude that removing reactions R3 and R4 or, in alternative, R5 allows the production of P1.

Using the proposed tool and selecting any of the available simulation and optimization methods to maximize the production of P1, it finds two possible sets of knock-outs (TF1, G4 or TF1, G6) obtained in two independent runs. These are the same sets as the ones calculated using the OptORF approach. Inactivating TF1 and G4, reactions R3 and R4 are removed, while in the case of TF1 and G6, reactions R3 and R5 are eliminated. Both produce the desired result.

3.2. Ethanol production using *E. coli*

To further test the proposed software with a larger case study, the production of ethanol using *E. coli* as the host microorganism was considered. In this case, genome-scale metabolic and regulatory models for *E. coli* are used and the results are compared to the ones reported in Kim and Reed (2010).

The results are given in detail in a file provided as **supplementary material** containing a detailed description of all experiments and tables with the results. The analysis of these results shows that the software is able not only to find strategies that are similar to the ones previously published, but also to find other solutions that perform well both in terms of growth and ethanol production.

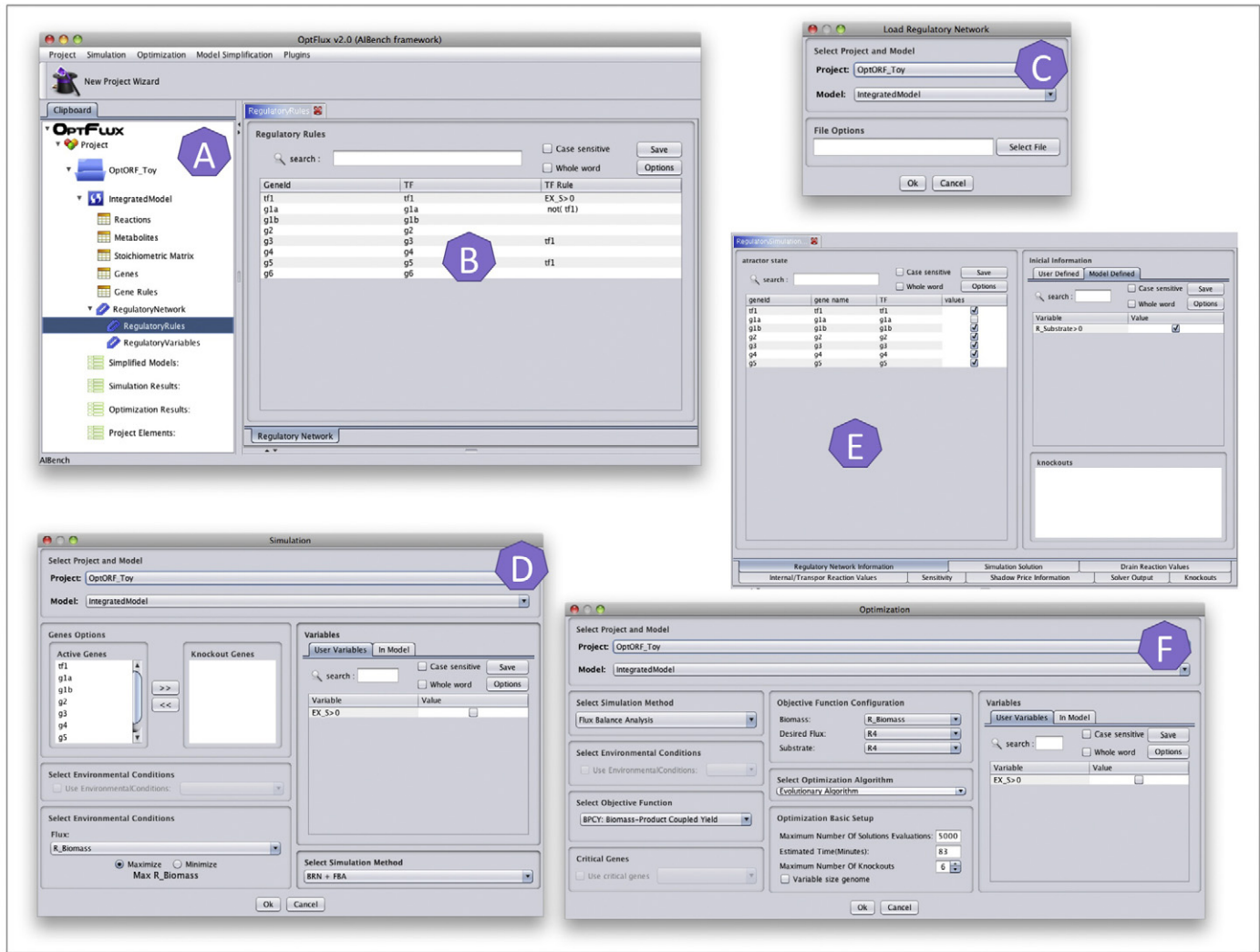


Fig. 3. Screenshots of OptFlux with the proposed plug-in. (A) Clipboard containing the main datatypes; (B) one of the views of the regulatory model; (C) load regulatory model operation interface; (D) mutant simulation operation interface; (E) view that shows an attractor from the simulation of a regulatory model; and (F) strain optimization operation interface.

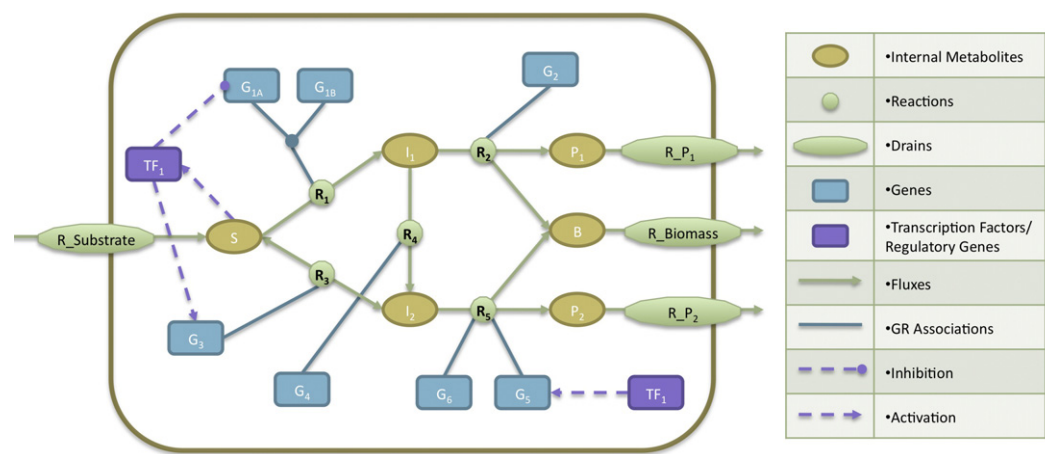


Fig. 4. Prototype network – “toy” integrated model used in the examples.

4. Conclusions and further work

To the best of the authors' knowledge, the proposed plug-in makes the OptFlux platform the first available software that integrates regulatory with metabolic models, allowing both phenotype simulation and strain optimization operations. The software is available for the use of the ME community and it has been validated in two case studies, one including a genome-scale model for *E. coli*.

As further work, the authors intend to pursue the validation of the tool with other case studies and also to enlarge its functionalities. The implementation of the OptORF method proposed in Kim and Reed (2010) is one of the priorities.

Also, the authors are aware of the limitations of this work regarding the approach followed for regulatory network simulation, namely the assumption related to synchrony and the related problem of the possible existence of multiple attractors. Although this does not represent accurately the biological phenomena, it is a reasonable assumption in some of the most used networks currently available (Samal and Jain, 2008). In future work, the authors intend to improve the capabilities of the software, by taking advantage of previous work in the search of attractors without simulation of the Boolean network (Devloo et al., 2003) and asynchronous network simulation (Naldi et al., 2009).

Acknowledgements

This work was funded by Portuguese FCT (MIT-Portugal Program) through the project MIT-PT/BS-BB/0082/2008.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.biosystems.2010.11.012.

References

- Becker, S., Feist, A., Mo, M., Hannum, G., Palsson, B., Herrgard, M., 2007. Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox. *Nature Protocols* 2 (3), 727–738.
- Burgard, A., Pharkya, P., Maranas, C., 2003. OptKnock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnology and Bioengineering* 84 (6), 647–657.
- Covert, M., Schilling, C., Palsson, B., 2001. Regulation of gene expression in flux balance models of metabolism. *Journal of Theoretical Biology* 213 (1), 73–88.
- Covert, M., Knight, E., Reed, J., Herrgard, M., Palsson, B., 2004. Integrating high-throughput and computational data elucidates bacterial networks. *Nature* 429 (6987), 92–96.
- Devloo, V., Hansen, P., Labbé, M., 2003. Identification of all steady states in large networks by logical analysis. *Bulletin of Mathematical Biology* 65, 1025–1051.
- Glez-Peña, D., Reboiro-Jato, M., Maia, P., Rocha, M., Díaz, F., Fdez-Riverola, F., 2010. AIBench: a rapid application development framework for translational research in biomedicine. *Computer Methods and Programs in Biomedicine* 98 (2), 191–203.
- Ibarra, R., Edwards, J., Palsson, B., 2002. *Escherichia coli* K-12 undergoes adaptive evolution to achieve in silico predicted optimal growth. *Nature* 420 (6912), 186–189.
- Kauffman, K., Prakash, P., Edwards, J., 2003. Advances in flux balance analysis. *Current Opinion in Biotechnology* 14 (5), 491–496.
- Kauffman, S., Peterson, C., Samuelsson, B., Troein, C., 2004. Genetic networks with canalizing Boolean rules are always stable. *Proceedings of the National Academy of Sciences of the United States of America* 101 (49), 17102–17107.
- Kim, J., Reed, J., 2010. OptORF: optimal metabolic and regulatory perturbations for metabolic engineering of microbial strains. *BMC Systems Biology* 4 (1), 53.
- Klamt, S., Saez-Rodriguez, J., Gilles, E., 2007. Structural and functional analysis of cellular networks with CellNetAnalyzer. *BMC Systems Biology* 1, 2.
- Li, S., Assmann, S., Albert, R., 2006. Predicting essential components of signal transduction networks: a dynamic model of guard cell abscisic acid signaling. *PLoS Biology* 4 (10), e312.
- Naldi, A., Berenguier, D., Fauré, A., Lopez, F., Thieffry, D., Chaouiya, C., 2009. Logical modelling of regulatory networks with GINsim 2.3. *Bio Systems* 97 (2), 134–139.
- Patil, K., Rocha, I., Förster, J., Nielsen, J., 2005. Evolutionary programming as a platform for in silico metabolic engineering. *BMC Bioinformatics* 6, 308.
- Reed, J., Vo, T., Schilling, C., Palsson, B., 2003. An expanded genome-scale model of *Escherichia coli* K-12 (iJR904 GSM/GPR). *Genome Biology* 4 (9), R54.
- Rocha, M., Maia, P., Mendes, R., Pinto, J., Ferreira, E., Nielsen, J., et al., 2008. Natural computation meta-heuristics for the in silico optimization of microbial strains. *BMC Bioinformatics* 9, 499.
- Rocha, I., Maia, P., Evangelista, P., Vilaça, P., Soares, S., Pinto, J., et al., 2010. OptFlux: an open-source software platform for in silico metabolic engineering. *BMC Systems Biology* 4, 45.
- Samal, A., Jain, S., 2008. The regulatory network of *E. coli* metabolism as a boolean dynamical system exhibits both homeostasis and flexibility of response. *BMC Systems Biology* 2, 21.
- Segrè, D., Vitkup, D., Church, G., 2002. Analysis of optimality in natural and perturbed metabolic networks. *Proceedings of the National Academy of Sciences of the United States of America* 99 (23), 15112–15117.
- Shlomi, T., Berkman, O., Ruppin, E., 2005. Regulatory on/off minimization of metabolic flux changes after genetic perturbations. *Proceedings of the National Academy of Sciences of the United States of America* 102 (21), 7695–7700.
- Shlomi, T., Eisenberg, Y., Sharan, R., Ruppin, E., 2007. A genome-scale computational study of the interplay between transcriptional regulation and metabolism. *Molecular Systems Biology* 3, 101.
- Vilaça, P., Maia, P., Rocha, I., Rocha, M., 2010. Metaheuristics for strain optimization using transcriptional information enriched metabolic models. *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, 205–216.