In silico strain optimization by adding reactions to metabolic models

Sara Correia[1] and Miguel Rocha[1*]

[1]CCTC, University of Minho, Campus de Gualtar, Braga, Portugal
{scorreia|mrocha}@di.uminho.pt

Summary

Nowadays, the concerns about the environment and the needs to increase the productivity at low costs, demand for the search of new ways to produce compounds with industrial interest. Based on the increasing knowledge of biological processes, through genome sequencing projects, and high-throughput experimental techniques as well as the available computational tools, the use of microorganisms has been considered as an approach to produce desirable compounds. However, this usually requires to manipulate these organisms by genetic engineering and/ or changing the enviromental conditions to make the production of these compounds possible. In many cases, it is necessary to enrich the genetic material of those microbes with hereologous pathways from other species and consequently adding the potential to produce novel compounds.
This paper introduces a new plug-in for the OptFlux Metabolic Engineering platform, aimed at finding suitable sets of reactions to add to the genomes of selected microbes (wild type strain), as well as finding complementary sets of deletions, so that the mutant becomes able to overproduce compounds with industrial interest, while preserving their viability. The necessity of adding reactions to the metabolic model arises from existing gaps in the original model or motivated by the productions of new compounds by the organism. The optimization methods used are metaheuristics such as Evolutionary Algorithms and Simulated Annealing. The usefulness of this plug-in is demonstrated by a case study, regarding the production of vanillin by the bacterium *E. coli*.

1 Introduction

In a world where the markets pressure, globalization and the shortage of some natural resources have a large impact on the world economy, it is urgent to find alternative processes to produce compounds of interest, as well as to increase the productivity of existing processes. Biotechnological processes, involving the use of selected microorganisms to produce substances with industrial interest, are becoming an alternative to tradicional processes. To make these processes competivite, it becomes crucial to have a deep knowledge of the biological processes ocurring in these microbes, a task made possible by the recent genome sequencing projects and other experimental techniques, together with the boost of the Bioinformatics field to handle these new data. One interesting approach involves making use of computational tools to simulate the microbe's behavior *in silico*, before the application of genetic or enviromental manipulations *in vivo* [1].

An important challenge in Metabolic Engineering (ME) [2] consists in the identification of genetic manipulations to be applied to an organism, with the aim of constructing a mutant strain

*To whom correspondence should be addressed.

able to produce compounds of industrial interest. Based on the knowledge about the biological system and, more specifically, its metabolic network, we can manipulate the environment in which it develops, or alter it genetically, to maximize the production of a given compound [3].

Recently, advances have been achieved concerning the available knowledge of some biological organisms, for instance from the sequencing of their genomes and also from various types of high-throughput experimental data (e.g. gene expression, proteomics). However, the lack of tools to perform the analysis and interpretation of biological data still limits the use and interconnection of that knowledge [4].

In this context, OptFlux (http://www.optflux.org) [5] is an open-source and modular platform for ME, incorporating strain optimization tasks, using Evolutionary Algorithms (EAs) [6] and Simulated Annealing (SA) [7]. OptFlux also allows the use of stoichiometric metabolic models for phenotype simulation of both wild-type and mutant organisms, *Metabolic Flux Analysis* and pathway analysis using Elementary Flux Modes, among other features.

When performing strain optimization, some limitations arise from the fact that metabolic models are incomplete [8] or the desired product can not be produced by the organism. In both cases, it will be necessary to find reactions to add to the metabolic model. In this paper, we present a new plug-in for OptFlux that allows to incorporate a set of reactions from an external database into an existing metabolic model performing phenotype simulation using those added reactions. Also, optimization methods will be put forward to allow the selection of the best set of reactions to add to the model, according to a given objective function (e.g. maximizing the production of a compound or filling gaps in the model).

The major concern, during the optimization task, is the large amount of reactions in the external database. If the number of reactions is high, the number of possible combinations increases exponentially and therefore it is crucial to reduce the number of reactions before starting the optimization process. So, methods to reduce and filter the database given a desired product are also developed in this work.

2 Methods for phenotype simulation and strain optimization

The simulation process allows the prediction of the organism phenotype, using methods based on fundamental restrictions to the biological system. One of these methods is Flux Balance Analysis (FBA) [9, 10], that calculates the flux distribution making it possible to predict the growth rate of an organism or the rate of production of a metabolite, based on stoichiometric, reversibility and fluxes constraints. FBA assumes that metabolic networks will reach a steady state constrained by the stoichiometry.

Predicting the metabolic state of an organism after a genetic manipulation (e.g. gene knockout) is a challenging task, because mutants are generally not subjected to the same evolutionary pressure that shaped the wild type. In these cases, other methods such *Minimization of Metabolic Adjustment* (MOMA) [11] and *Regulatory On/Off Minimization of metabolic fluxes* (ROOM) [12] are proposed to find a flux distribution for mutant strains.

Based on these methods, a question arises: how to find the ideal set of genes to be deleted to reach the desired phenotype? To try answer this question, the *OptGene* algorithm proposed by Patil et al [13] and its extensions made by Rocha et al [14] were proposed. In this last work,

the authors' research group proposed a set-based representation that considered variable-sized solutions, allowing for solutions with different numbers of knockouts during the optimization process. Two optimization algorithms were developed: SA and Set-based EAs (SEAs). Both search for the optimum set size in parallel with the search for the optimum set of gene deletions.

This work aims to enlarge the set of possible genetic modifications by addressing gene additions. In this case, using SEAs or SA approaches, the optimization process finds a set of new reactions to be added to the model. Optionally, a complementary set of reactions to remove can also be optimized. Optimization methods are the same that were used previously. The main difference lies in the representation of the solutions. Although still using a representation based on sets, it is necessary to integrate information regarding the reactions to be added. Thus, a new way of representing solutions including two independent sets (knockouts and added reactions) was created. In Figure 1, the representation of one solution is depicted.

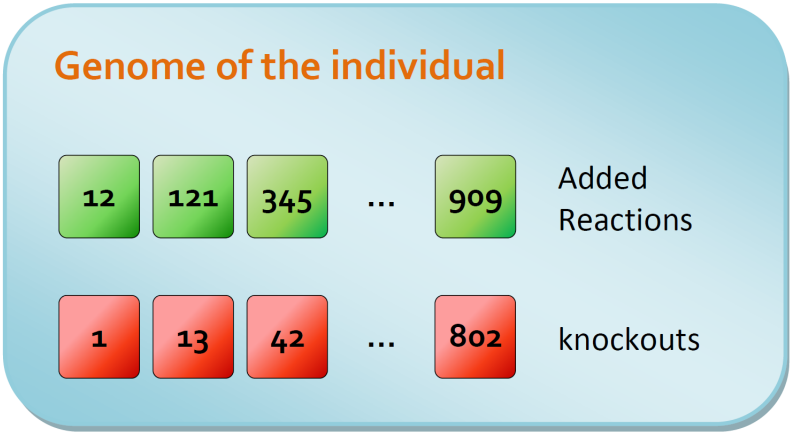

Figure 1: Representation of the genome of an individual. Green squares represent reactions that will be added to the model (numbers are the reactions indexes in the external database). The knockouts are represented by red squares (numbers are indexes of reactions in the model).

For the new solution representation used in SA and EAs algorithms, it is necessary to develop new reproduction operators: crossover and mutation. These operators only modify one of the sets (knockouts or added reactions) in the genome, randomly selected. The new used operators are:

- **Random Mutation**: replaces an element of the set by another;
- **Grow Mutation**: introduces a number of new elements into the set, whose values are randomly generated;
- **Shrink Mutation**: removes a number of randomly selected elements from the set;
- **Crossover**: 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.

3 OptFlux plug-in for adding reactions

A new plug-in was developed for OptFlux to allow the addition of external reactions to a metabolic model. The addition of new reactions can be made for phenotype simulation or

to conduct a strain optimization process. Methods to import, filter and visualize the external database of reactions are also available. In the OptFlux platform the new functionalities can be accessed by the "Plugins/ Add Reactions" menu.

3.1 Import database of reactions

Importing an external database of reactions into OptFlux can be made using the same methods used for creating metabolic models (SBML [15] and flat text files). Also, a new format of text files is defined (details are in the site documentation) to allow a more flexible scheme. When using this format, the user can filter the input data files to select only reactions that satisfy some restrictions. This is useful for readability and to reduce the search space in the optimization tasks. In Figure 2, the application of two filters to a database is shown. After applying filters, the user obtains a set of reactions that will be imported to the OptFlux platform.

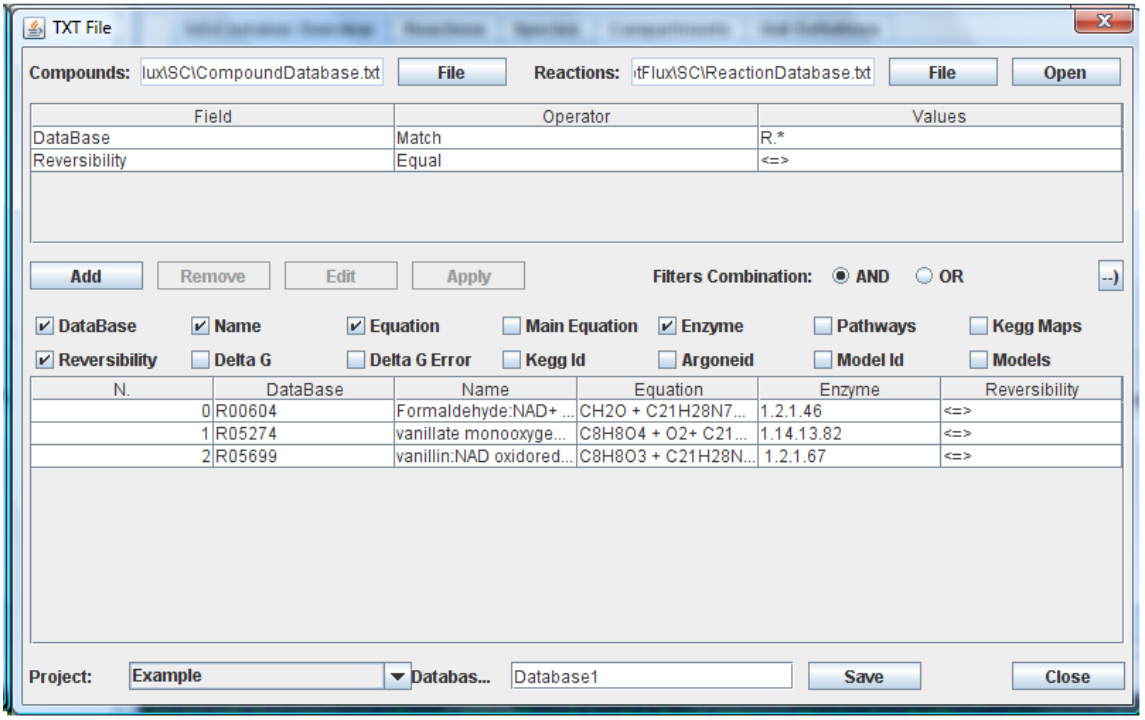

Figure 2: Interface for selecting reactions and importing them to OptFlux. In this example, the user chooses only the reactions where ids start with "R" and that are reversible.

The reactions database size can be reduced by applying methods to keep only the reactions that may lead to the production of the target compound.

3.2 Filter database by target compound

As stated before, the major problem in finding the reactions set to add to the metabolic model is related with the search space size, i.e. the number of reactions contained in the external database. To overcome this obstacle, methods were developed to filter the reactions database. Given the target product, the algorithm selects, from the database, only the reactions present in pathways between metabolites present in the metabolic model and the target product. The user can manipulate the number of selected reactions through a tolerance parameter.

The main steps of the algorithm are the following:

- construct a structure that represents the dependencies between the target product and metabolites belonging to the metabolic model. All metabolites that belong to the metabolic model have the $cost = 1$, and the the reactions of external database and intermediate metabolites have an empty cost, before start runing the algorithm.
- calculate the cost to obtain each intermediate metabolite. The cost is the lower value from the reactions costs that produce it. The cost of a reaction is given by the sum of its reagents costs, when all reagents have an associated cost. In each algorithm iteration a set of new metabolites and reactions costs are calculated. When during an iteration is impossible to calculate any missing cost, because is no way to obtain a specific metabolite, the algorithm stops.
- insert into the database the reactions that belong to pathways with lower cost. Through the tolerance parameter, the higher accepted cost is given by the rule $min_c ost + (max\_cost - min\_cost) * tolerance/100$. In Figure 3 we can observe one example of a set of selected reactions, if the cost for obtaining the $MetaboliteX$ varies from 2 to 10, and the user chooses 30% for tolerance value, the reactions that will be picked have costs between 2 and 4.4.

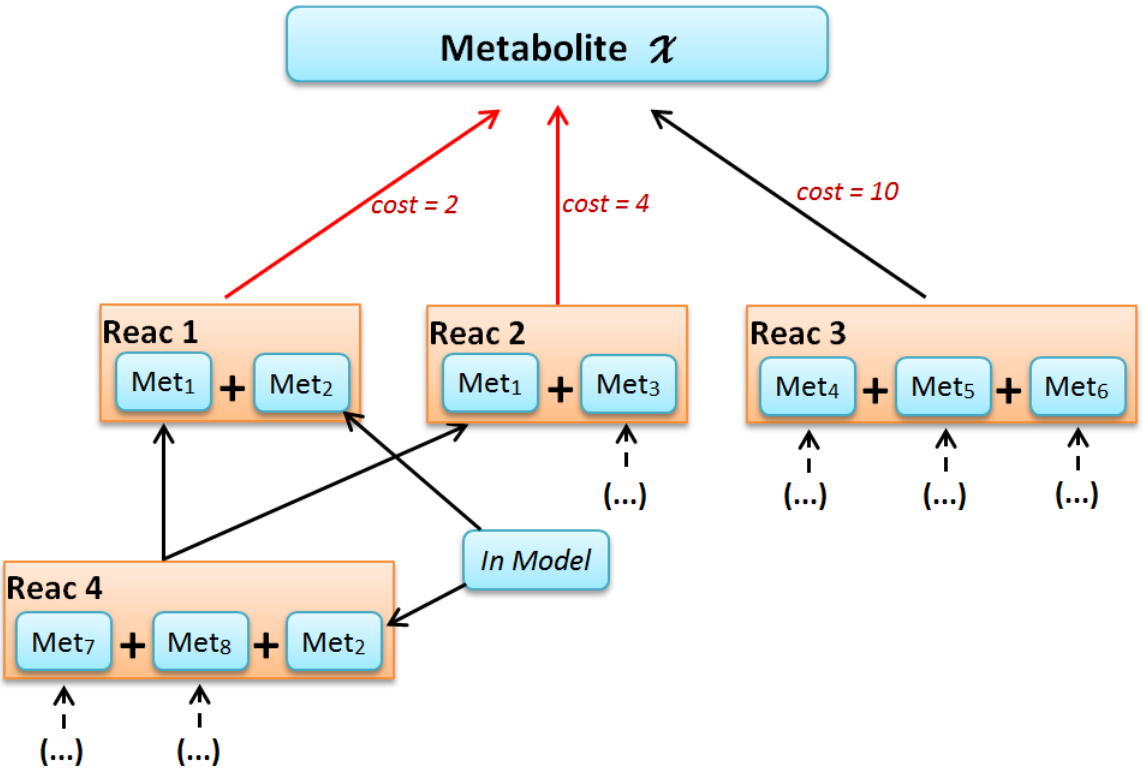

Figure 3: This scheme represents the reactions that will be chosen for the metabolite X production. In this example we can observe that the metabolite x can be produced through the reactions: Reac1, Reac2 and Reac3, with costs 2, 4 and 10 respectively. If the user chooses 30% for tolerance value, only the reactions Reac1 and Reac2 will be inserted in the database. This process is repeated for all metabolites from reactions Reac1 and Reac2. Note that the Met2 is already in the original metabolic model.

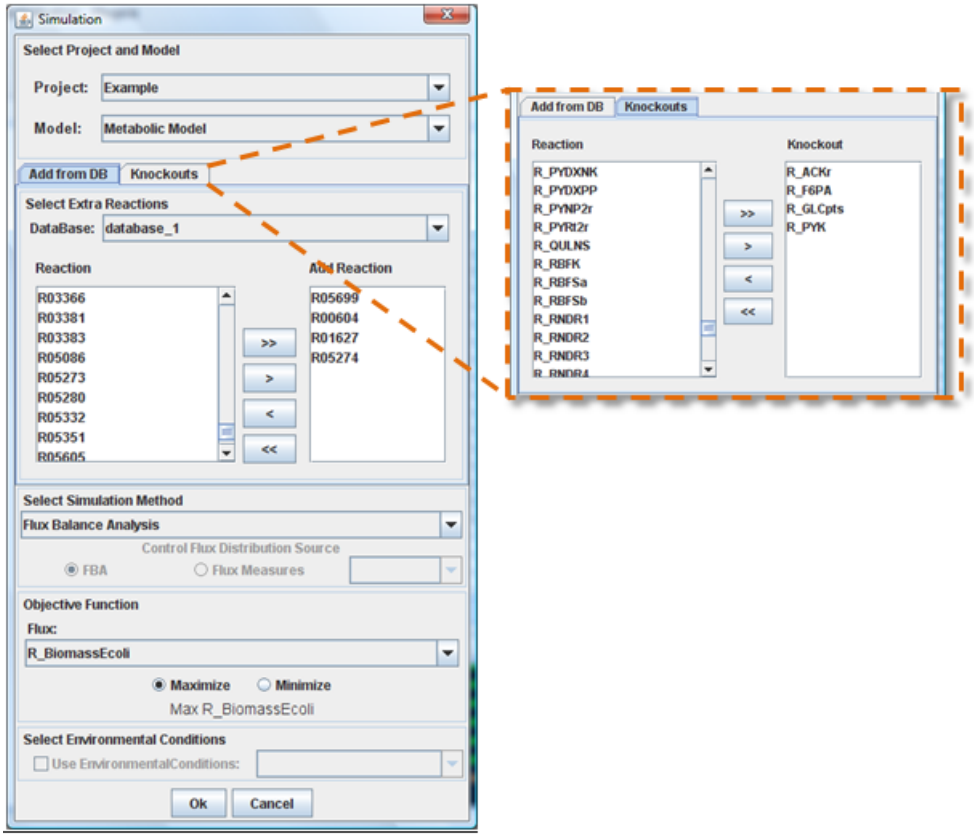

Figure 4: Interface for mutant simulation. The case study of vanillin production is shown here (see below). In the example, 4 knockouts and 4 added reactions are selected.

This filter, based on the desired product, can be very useful to create a new database with reactions that belong to new pathways for producing the target compound. However, this filter may discard reactions that when added to the metabolic model could increase the yield of target compound or biomass production.

3.3 Mutant simulation by adding reactions

The phenotype simulation functionality allows mutant simulation by adding new reactions and optionally removing others from the model. After selecting the model to use, a previously loaded database is selected and the set of reactions to be added is chosen. Also, a set of knockouts can be selected. In Figure 4, the simulation interface is presented. During the configuration process, the user selects the simulation methods (FBA, MOMA or ROOM), the environmental conditions (the rates at which external metabolites can be consumed/ produced), and the objective function (e.g. the maximization/ minimization of a selected flux).

The result of mutant simulation can be observed in a specific interface (Figure 5), where the user can check the main results of the simulation: the list of added reactions, list of knockouts and values for all fluxes in the model.

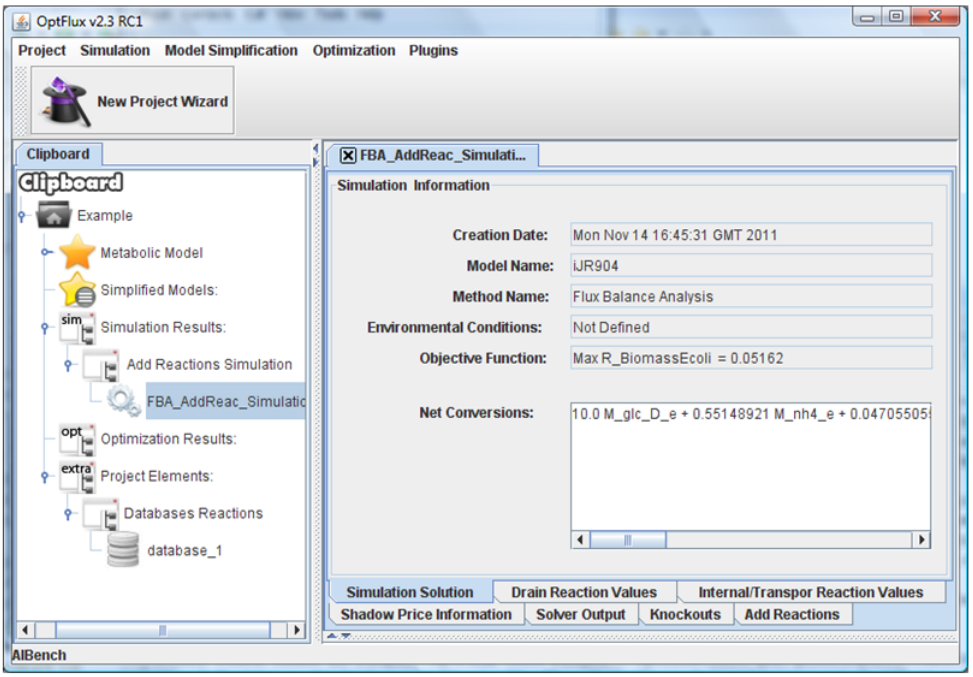

Figure 5: Interface showing simulation results: in the left the clipboard shows the main objects and in the right side the visualization of the main results of mutant simulation are shown in distinct tabs.

3.4 Strain optimization by adding reactions

The strain optimization process tries to find a set of reactions from a database to be added to the model to improve a given objective function (e.g. the production of specific product). The search can be for only a set of reactions to be added or the combination of added reactions and knockouts. In the interface (Figure 6), the user selects:

- **algorithm**: available optimization algorithms are EAs and SA;
- **simulation methods**: to be used in the simulation of each solution evaluated (FBA, MOMA or ROOM);
- **objective function**: used to calculate the fitness value of each solution; options are the Biomass-Product Coupled Yield (BPCY) [13] and Product Yield. The BPCY function is given by $BPCY = (PG)/S$, 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;
- **optimization basic setup**: configure the maximum number of solution evaluations, the maximum number of knockouts and added reactions and if the genome size should be fixed or have a variable size;
- **environmental conditions**: as defined for the simulation;
- **essential information**: define if it is possible to knockout some special type of reactions like drains, transport and critical reactions.

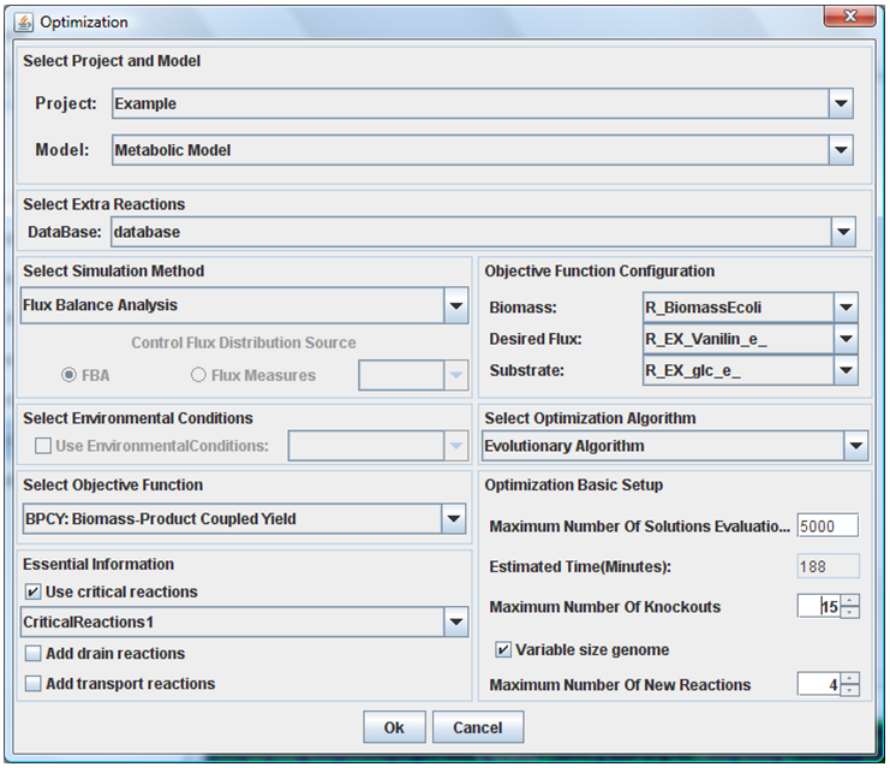

Figure 6: Interface for strain optimization processes. In this example, an EA is configured, the simulation method is FBA, the objective function is BPCY, essential information uses the critical reactions, a maximum of 15 knockouts and 4 new reactions are permitted in variable sized sets.

Furthermore, there are available interfaces to view the solutions found during the optimization task. For each, the list of knockouts, the list of added reactions and the flux values for all reactions of the metabolic model are displayed.

4 Results

4.1 Rebuilding gaps in the metabolic model

In this case study, used for validation purposes, OptFlux simplification methods were used to identify reactions constrained to a flux value of zero in the *E.coli* model. The model is reduced eliminating those reactions and a database is created with the removed reactions (407 reactions). In each run, three randomly selected reactions are further removed from the new reduced model and inserted into the database. The optimization methods must find these reactions and re-integrate them in the model to maximize biomass production. This process was repeated 10 times for SA and EA. The number of evaluations needed to find the optimal solution in each run are given in Table 1.

4.2 Vanillin case study

This case study aims to identify new pathways for the production of vanillin from glucose in *E. coli* and validate the implemented simulation method. To demonstrate the validity of the

Table 1: Number of function evaluations to find the optimal solution using SA and EA algorithms.

Test reactions	EA	SA
TPI,TKT1 e TKT2	500	300
IGPS, IDOND e ENO	2060	1120
MDH, ICDHyr e CBMK	2700	2930
IPPS, HSST e GSNK	9550	1735
PANTS, P5CR e ORPT	8240	4270
ADCL, IMPD e PSERT	6750	11103
RPI, TALA e ACLS	1215	1680
ACOTA, DDPA e PFL	2020	998
PRPPS, SPMS e TRDR	1035	5302
A5PISO, RPI e TYRTA	9065	7650

simulation process, we used the previous study with the OptStrain framework [16]. To proceed with the test it was required to build the reactions database to add to the metabolic model. The new pathway added to the metabolic model can be observed in Figure 7.

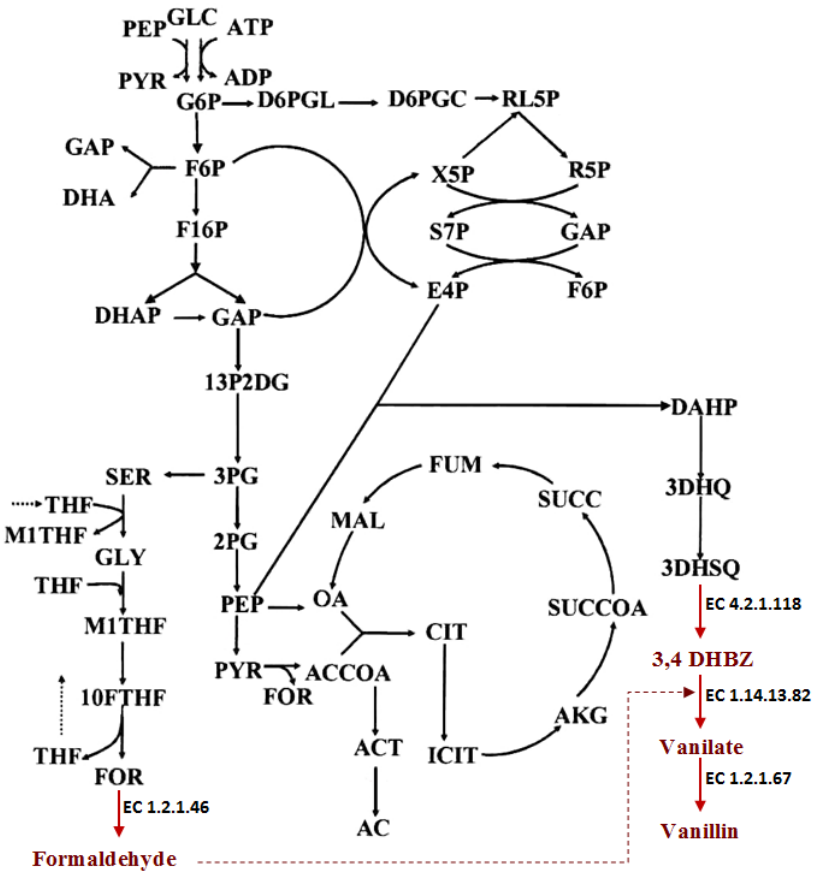

Figure 7: The added pathway for vanillin production. The figure is partially adapted from Maranas et al. (2004)[16]

The simulation was performed for each of three sets of knockouts in the paper, considering the substrate flux of 10 $mmol/gDWh^{-1}$ and the objective function the maximization of biomass production. The FBA was used as the simulation method in phenotype simulation. The obtained results agree with the ones from the previous work [16], thus validating our implementation.

The next step was to run the strain optimization processes to find a set of added reactions and knockouts, that maximizes vanillin production coupled with the organism growth.

The process was run 30 times for each EA and SA algorithms using as objective function the Biomass-Product Coupled Yield (BPCY). This function assigns the same importance for the biomass and target product production, multiplying their values. Before starting the optimization process, it was necessary to change the ids of the metabolites from the metabolic model to those used in the database. Otherwise, it would not be possible to integrate new reactions in the metabolic model.

The process parameters were configured taking EAs and SA encoding sets with a variable size. An important parameter is concerned with the maximum size allowed for both sets, containing respectrively the list of knockouts and the list of added reactions. Thus, four distinct configurations were tested with different combinations for the maximum number of knockouts and of added reactions as follows:

config1 $\Leftarrow$ number of knockouts: 20 and number of added reactions 10;
config2 $\Leftarrow$ number of knockouts: 10 and number of added reactions 10;
config3 $\Leftarrow$ number of knockouts: 7 and number of added reactions 7;
config4 $\Leftarrow$ number of knockouts: 5 and number of added reactions 5.

Figure 8 shows the distribution of obtained values for the biomass and vanilin production using each configuration for the EA and SA algorithms.

Table 2 shows the 95% confidence interval of results obtained in the optimization process, considering the best solution from each run.

Table 2: The 95% confidence interval of results obtained in the optimization process.

		Config 1	Config 2	Config 3	Config 4
EA	Fitness (BPCY)	[0.170;0.181]	[0.169;0.182]	[0.165;0.172]	[0.165;0.175]
	Biomass	[0.309;0.351]	[0.304;0.431]	[0.289;0.418]	[0.299;0.443]
	Product	[5.264;5.478]	[4.850;5.499]	[4.861;5.515]	[4.719;5.443]
	Knockouts	[8.780;11.420]	[6.840;8.227]	[6.101;6.699]	[4.651;4.949]
	Added reactions	[8.741;9.259]	[8.763;9.370]	[6.225;6.642]	[4.382;4.752]
SA	Fitness (BPCY)	[0.177;0.189]	[0.175;0.188]	[0.162;0.169]	[0.168;0.169]
	Biomass	[0.323;0.437]	[0.320;0.392]	[0.276;0.363]	[0.309;0.310]
	Product	[4.822;5.407]	[5.054;5.423]	[5.171;5.622]	[5.405;5.471]
	Knockouts	[11.134;14.332]	[6.727;8.140]	[5.161;5.972]	[4.568;4.898]
	Added reactions	[8.409;9.058]	[8.430;9.037]	[5.317;5.883]	[4.0;4.0]

Comparing these results with the ones obtained in the previous study [16], we see that the BPCY value of their solution was 0.035 ($BPCY = (6.787 \times 0.052)/10 = 0.035$). Although the vanillin production is lower in our case, the BPCY value increased significantly given that the biomass is much higher, which mean that our strain has a larger growth rate.

Afterwards, we focused in increasing the production of vanillin, without treating the biomass formation as a priority. Considering this, the tests were repeated with a new objective function, by maximizing the flux of the product, ensuring a minimum limit of biomass production (5% of the wild type value). The results shown in Table 3 contain the best solution for each configuration, where the production of vanillin is higher than the obtained in [16].

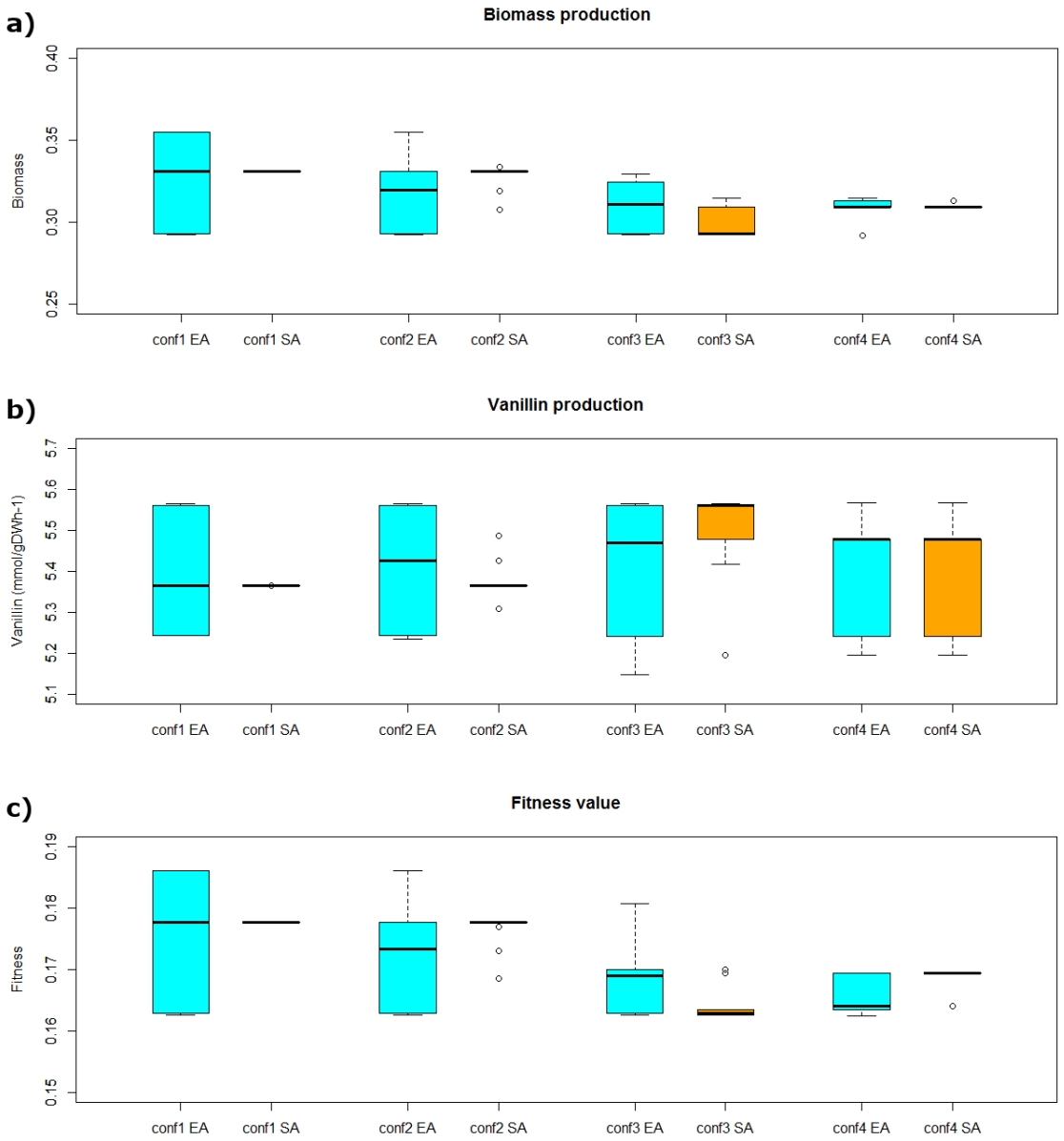

Figure 8: This figure shows the results obtained by the EA and SA algorithms in the 30 runs for each of the configurations showing boxplots for the: a) the values of biomass fluxes; b) the values for the desired product flux (vanillin production) and c) the fitness values (BPCY).

The smaller set of added reactions suggested by the optimization process include the reactions with KEGG [17] (http://www.genome.jp/kegg) ids:

- R01216 - 2-dehydropantoate formaldehyde-lyase;
- R01627 - 3-dehydroshikimate hydro-lyase;
- R05273 - oxygen oxidoreductase;
- R05274 - vanillate:oxygen oxidoreductase.

A supplementary file containing the full results obtained in the experiments summarized here is given in `http://darwin.di.uminho.pt/jib/`.

Table 3: Best results of strain optimization for vanillin production using the Yield objective function for each algorithm (EA and SA).

		Product	Biomass	No. Knockouts	No. added reactions
EA	config 1	6.948	0.022	20	4
	config 2	7.003	0.011	9	7
	config 3	6.978	0.016	7	6
	config 4	6.533	0.103	5	5
SA	config 1	6.948	0.022	17	6
	config 2	6.985	0.014	10	4
	config 3	6.934	0.024	7	6
	config 4	6.466	0.116	5	4

5 Conclusion

This paper presents methods for the simulation of strains by adding external reactions to the metabolic model, aiming to produce a target product with industrial interest or to fill gaps in metabolic model. In this approach, information is added to the stoichiometric model regarding new reactions, thus making an extension to the initial model.

Users can create their own database using files containing information on reactions and metabolites. After importing the database to the OptFlux platform it is possible to create new databases with a subset of reactions by applying some filters. Thus, the search space of possible solutions can be reduced this way.

Methods for strain optimization were developed, using EAs and SA, to find a sets of external reactions to be added and the necessary knockouts to maximize an objective function, typically related to the production of a compound of interest.

To provide these features to the scientific community, a plug-in has been developed for the OptFlux ME platform that allows simple and intuitive phenotype simulation and strain optimization with the addition of external reactions to the metabolic model. Thus, the tool set available for ME experts has been enlarged with useful techniques.

Future work will be devoted to the validation of these methods with other real world case studies.

Acknowledgements

This work is supported by project PTDC/EIA-EIA/115176/2009, funded by Portuguese FCT and Programa COMPETE.

References

[1] 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 Biochem Sci*,

26(3):179–86, 2001.

[2] H. Kitano. Systems biology: a brief overview. *Science*, 295:1662–1664, 2002.

[3] J. Nielsen. Metabolic engineering. *Applied Microbiology and Biotechnology*, 55(3):263–283, 2001.

[4] J. S. Edwards, M. Covert, and B. Palsson. Metabolic modelling of microbes: the flux-balance approach. *Environ Microbiol*, 4(3):133–40, 2002.

[5] I. Rocha, P. Maia, P. Evangelista, P. Vilaca, S. Soares, J. P. Pinto, J. Nielsen, K. R. Patil, E. C. Ferreira, and M. Rocha. OptFlux: an open-source software platform for in silico metabolic engineering. *BMC Syst Biol*, 4:45, 2010.

[6] T. Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Dortmund, Germany, 1996.

[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

[8] V. S. Kumar, M. S. Dasika, and C. D. Maranas. Optimization based automated curation of metabolic reconstructions. *BMC Bioinformatics*, 8:212, 2007.

[9] J. D. Orth, I. Thiele, and B. . Palsson. What is flux balance analysis? *Nature Biotechnology*, 2010.

[10] K. J. Kauffman, P. Prakash, and J. S. Edwards. Advances in flux balance analysis. *Curr Opin Biotechnol*, 14(5):491–6, 2003.

[11] D. Segrè, D. Vitkup, and G. M. Church. Analysis of optimality in natural and perturbed metabolic networks. *Proc Natl Acad Sci U S A*, 99(23):15112–7, 2002.

[12] T. Shlomi, O. Berkman, and E. Ruppin. Regulatory on/off minimization of metabolic flux changes after genetic perturbations. *Proc Natl Acad Sci U S A*, 102(21):7695–700, 2005.

[13] K. R. Patil, I. Rocha, J. Förster, and J. Nielsen. Evolutionary programming as a platform for in silico metabolic engineering. *BMC Bioinformatics*, 6:308, 2005.

[14] 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*, 9:499, 2008.

[15] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, et al. The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–31, 2003.

[16] P. Pharkya, A. P. Burgard, and C. D. Maranas. Optstrain: a computational framework for redesign of microbial production systems. *Genome Res*, 14(11):2367–76, 2004.

[17] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, and Y. Yamanishi. Kegg for linking genomes to life and the environment. *Nucleic Acids Research*, 36(Database issue):D480–D484, 2008.