BIOINFORMATICS ORIGINAL PAPER

Vol. 28 no. 23 2012, pages 3097–3104
doi:10.1093/bioinformatics/bts590

Systems biology

Advance Access publication October 7, 2012

Robust design of microbial strains

Jole Costanza[1], Giovanni Carapezza[1], Claudio Angione[2], Pietro Lió[2,*] and Giuseppe Nicosia[1,*]

[1]Department of Mathematics and Computer Science, University of Catania, Viale A. Doria 6, 95125 Catania, Italy and [2]Computer Laboratory, University of Cambridge, William Gates Building, 15 JJ Thomson Avenue, Cambridge CB3 0FD, UK

Associate Editor: Trey Ideker

ABSTRACT

Motivation: Metabolic engineering algorithms provide means to optimize a biological process leading to the improvement of a biotechnological interesting molecule. Therefore, it is important to understand how to act in a metabolic pathway in order to have the best results in terms of productions. In this work, we present a computational framework that searches for optimal and robust microbial strains that are able to produce target molecules. Our framework performs three tasks: it evaluates the parameter sensitivity of the microbial model, searches for the optimal genetic or fluxes design and finally calculates the robustness of the microbial strains. We are capable to combine the exploration of species, reactions, pathways and knockout parameter spaces with the Pareto-optimality principle.

Results: Our framework provides also theoretical and practical guidelines for design automation. The statistical cross comparison of our new optimization procedures, performed with respect to currently widely used algorithms for bacteria (e.g. *Escherichia coli*) over different multiple functions, reveals good performances over a variety of biotechnological products.

Availability: http://www.dmi.unict.it/nicosia/pathDesign.html.

Contact: nicosia@dmi.unict.it or pl219@cam.ac.uk

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received and revised on July 11, 2012; accepted on September 12, 2012

1 INTRODUCTION

Metabolic engineering is becoming central in basic and applied biological fields and requires mathematical models for accurate design purposes. The aim is overproducing desired metabolites by operating on genetic manipulations, as well as identifying novel and non-native synthesis pathways. Many organisms are used to analyze the metabolite production potential and identify the metabolic interventions needed to produce the metabolite of interest. Thus, strains have been systematically designed through *in silico* analysis to overproduce target metabolites, such as lycopene (Alper *et al.*, 2005), ethanol (Jarboe *et al.*, 2010) and isobutanol (Atsumi *et al.*, 2010). The efforts are particularly focused on predicting flux distributions and network capabilities, most notably flux balance analysis (FBA) (Orth *et al.*, 2010). Recent FBA models incorporate also information on enzymes and genome, and integrate the relationships among genes, enzymes and reactions. This makes it well suited to studies that characterize many different perturbations such as different substrates or genetic manipulations (knockouts).

In the past years, a variety of methods has been implemented to search for the genetic manipulations that optimize a cellular function of interest. These methods, such as OptKnock (Burgard *et al.*, 2003), OptFlux (Patil *et al.*, 2005), OptGene (Rocha *et al.*, 2008) and GDLS (Lun *et al.*, 2009), have been tested in FBA organism models. However, all these methods require high computational efforts: the execution times grow exponentially (Burgard *et al.*, 2003; Patil *et al.*, 2005; Rocha *et al.*, 2008) or linearly (Lun *et al.*, 2009) as the number of manipulations allowed in the final designs increases. Because of the large number of reactions occurring in the cellular metabolism, the dimension of the solution space is very large and finding genetic manipulations is quite expensive.

In this work, we use a multi-objective optimization algorithm to seek the genetic manipulations that optimize multiple cellular functions. The algorithm implements a global search with a heuristic and combinatorial method called genetic design through multi-objective optimization (GDMO). The idea is to use and improve the Pareto-optimal solutions. Pareto optimality is important to obtain not only a wide range of Pareto-optimal solutions but also the *best trade-off design*, as reported by Cutello *et al.* (2006) for the protein structure prediction problem. Moreover, the multi-objective optimization turns out to aid in the automatic design in several biological problems (Stracquadanio *et al.*, 2010).

The area underlying the Pareto curve and the first derivative, and in particular the presence of jumps (i.e. quick variations in the objective functions during the optimization procedure), carry valuable biotechnological information. For the first time, we use the *ε-dominance analysis* so as to consider all the solutions obtained by GDMO that are dominated with a tolerance ε by the Pareto-optimal solutions. We report that multi-objective optimization provides more insights than single-objective optimization on the capability of these organisms to adapt to the simultaneous presence of different conditions and constraints. We combine multiple-target optimization with knockout parameter space to investigate the most complete available metabolic data and search for the optimal nutrients in strains that allow the maximization or minimization of metabolic targets, namely *Escherichia coli* (Feist *et al.*, 2007), *Geobacter sulfurreducens* (Sun *et al.*, 2009) and many others.

*To whom correspondence should be addressed.

Additionally, we relate pathways to sensitivity analysis (SA). In modeling, SA is a method used to discover the main inputs, that is the inputs that have a substantial influence on the outputs of the model. In the last years, SA indices have been adopted in systems biology interrogating the reactions space (RoSA—reactions oriented SA) (Stracquadanio *et al.*, 2010) and species space (SoSA—species oriented SA) to find their influence on the outputs of the system (Zhang and Goutsias, 2010). In this work, we perform SA to find the most sensitive pathways in the FBA model of *E.coli*. In particular, we present the novel pathway-oriented SA (PoSA), to find the genetic manipulations that have the largest influence on the output of the model. Unlike other SA methods applied in biological modeling, whose inputs (reactions or species) are valued in a *real* region of interest, PoSA is applied when inputs are valued in a *binary* region of interest. PoSA investigates the knockout solution space and determines the influence of the pathways on the outputs of an FBA model. Since our search-and-optimize algorithm provides a set of feasible solutions with different genetic manipulations, it is worth seeking a relationship between the sensitivity indices and the proposed manipulations. In this way, we are able to select only the best manipulations. In particular, thanks to the information provided by PoSA, we can choose the GDMO knockout strategies that affect genes belonging to *insensitive pathways*.

Each point of the Pareto front represents a strain, i.e. an *E.coli* with specific genetic manipulations, and it is also associated with three robustness analysis (RA) indices that we compute. The *robustness* estimates how robust is a strain obtained by GDMO when it undergoes small perturbations, which can be *external* (changes in the nutrients) or *internal* (changes in the metabolism). Among the strains proposed by GDMO, we are able to choose the most robust one. In particular, we use three robustness methods to evaluate different components of the model.

2 SYSTEM AND METHODS

Here, we focus on the development of a new computational framework to design optimal and robust bacteria able to perform particular tasks.

In Figure 1, we show the layout of our computational framework applied to organisms modeled through FBA. The framework is composed of three blocks. The first is constituted by the SA, able to find the most sensitive parameters of the model. We can investigate the reactions and the species in the metabolic model in terms of sensitivity, using the RoSA and PoSA methods. Furthermore, the novel PoSA is able to identify the most sensitive metabolic pathways by ranking them according to knockouts. We consider a single pathway as an input of the system. Each pathway (that is a set of reactions converting particular substrates in specific final products) is perturbed by mutating genes that control its biochemical reactions. PoSA ranks the pathways according to their influence on the outputs of the model. Pathways with important influence have large sensitivity index (μ^* and σ^*), as reported in the pre-processing part of Figure 1. Each pathway in the graph is represented by a circle and its size indicates the number of genes belonging to it.

The multi-objective optimization algorithm searches both for the genetic manipulations (through gene deletions) and for nutrients with respect to defined target functions. Hence, we perform both the genetic design and the flux design in microbial strains. The result of the multi-objective optimization is a set of non-dominated points, called Pareto front (or Pareto surface). The non-dominated points are shown in red in Figure 1, while all the dominated points are shown in blue. All the dominated points and the non-dominated points, which satisfy all of the inequality and equality constraints, and all of the variable bounds, constitute the observed feasible region.

In the genetic design, each strain (a particular phenotype) is identified by a binary 'knockout vector' (which represents the

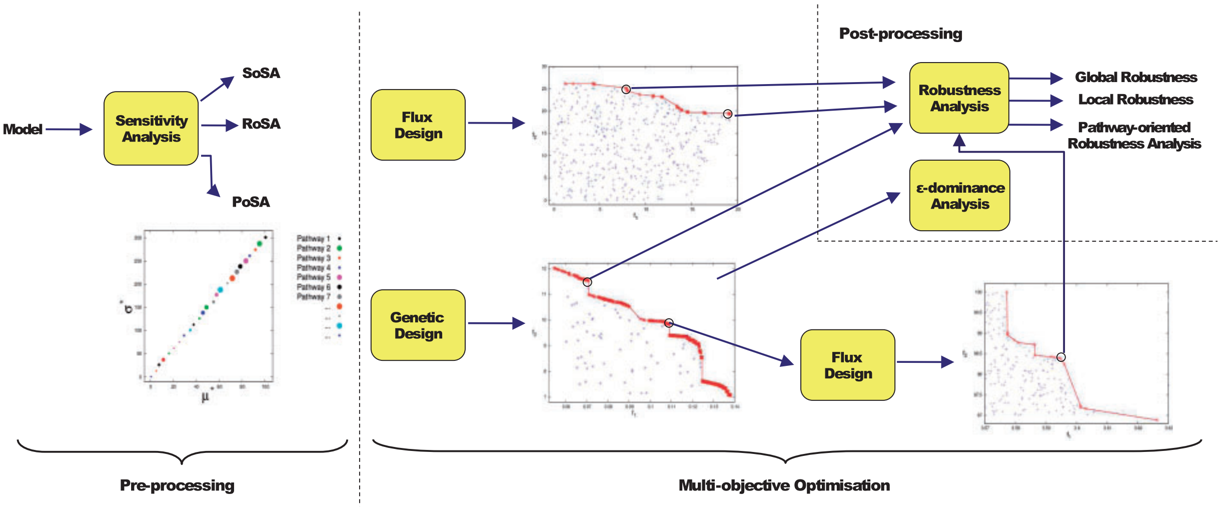

Fig. 1. A schematic representation of our automatic framework for optimal bacterial metabolism. In the pre-processing step, the species (SoSA), reaction (RoSA) and pathway-oriented SA (PoSA) are applied to the metabolic model. Then, the multi-objective optimization allows genetic and flux design. In the post-processing step, suitable solutions (selected from the Pareto front) are subjected to global, local and PoRA analysis. The ε-dominance analysis is performed to investigate the neighborhood of the suitable genetic designs

genotype), whose elements are 1 when the corresponding gene set is turned off. The importance of the knocked out genes can be evaluated by means of the ranking provided by PoSA. A gene set can be composed of a single gene, when it synthesizes for an enzyme, or can be associated with more genes, which synthesize for enzymes that form enzyme complexes and enzyme subunits. The relation between genes in a gene set is regulated by means of a Boolean relationship. When all the genes are necessary to catalyze the corresponding reactions (a single gene set can regulate more reactions), genes are linked by the 'AND' operator; otherwise, if one gene is sufficient, genes are linked by the 'OR' operator. In addition, through the multi-objective optimization, we are also able to find the favorable nutrients set (flux design) to optimize the wild-type/strains yield and evaluate the over/under investment of nutrients (uptake rate of fluxes). Pareto optimality is very useful for the analysis of metabolism, as reported in the previous works by Sendin *et al.* (2009) and Schuetz *et al.* (2012), where the authors used deterministic multi-objective approaches to evaluate the fluxes distributions in the *E.coli* wild-type network. In our work, we remark the usefulness of Pareto optimality and adopt an effective and state-of-the-art algorithm to investigate the knockout space. After the optimization, we perform an ε-dominance analysis to search accurately near the edge of the Pareto-optimal region.

The RA is the third task of our computational framework. For each phenotype (strain or wild type), in a post-processing step, we process the fragility of the metabolic network when it is subjected to small perturbations, which can be hexogen or endogen. From the Pareto fronts, we select interesting solutions using decision-making methods; for instance, we select the solutions near to the ideal solution, the knee points, the end points or points with suitable features. For each solution, we calculate the global robustness (GR), the local robustness (LR) and the pathway-oriented robustness (PoRA) values, indicating, respectively, the robustness of the whole network, of each single reaction and of each metabolic pathway.

We test the algorithm in the genome-metabolic network of *E.coli* (Feist *et al.*, 2007), composed of 2382 reactions, in order to maximize a metabolite of interest and simultaneously ensure the biomass formation, with the minimum knockout cost. The knockout cost is defined according to the Boolean relationship between genes. For example, if a gene set is composed of two genes linked by 'AND', the cost to ensure the turning off of the corresponding reactions (knockout cost) is 1. Instead, the cost to ensure the catalysis of the corresponding reactions is 2, since both genes are necessary to turn on the reactions associated with that gene set.

2.1 FBA modeling framework

FBA is a mathematical approach for analyzing the flow of metabolites through a metabolic network (e.g. their formation and degradation, transport and cellular utilization) composed of n biochemical reactions. For every metabolite $X_i, i = 1, \ldots, m$ a material balance is $\mathrm{d}X_i/\mathrm{d}t = \sum_{j=1}^{n} S_{ij}v_j$, where S_{ij} is the stoichiometric coefficient associated with each reaction flux $v_j, j = 1, \ldots, n$. If we consider this material balance under steady-state conditions, we have $\sum_{j=1}^{n} S_{ij}v_j = 0$. By considering all the intermediates simultaneously at steady state, the balance equation can be written in matrix form $Sv = 0$, where S is the stoichiometric matrix of m rows and n columns, and v is the vector of the fluxes (metabolic and transport fluxes). The matrix S is not square and $n > m$, so we have a plurality of solutions. Each solution is a flux distribution representing a particular metabolic state, depending on the genotype and on the transport fluxes. The FBA approach finds the metabolic state in order to optimize a particular objective function, such as the maximization of growth rate or ATP production. Consequently, the problem can be formulated as a linear programming problem:

$$\begin{aligned} &\text{maximize (or minimize)} \quad f'v, \text{ such that } Sv = 0 \\ &\qquad\qquad v_j^{\mathrm{L}} \leq v_j \leq v_j^{\mathrm{U}}, \quad j = 1, \ldots, n, \end{aligned} \tag{1}$$

where f is a vector of weights (n dimensional). All the elements in f are either 0 or 1. f_i is equal to 1 if v_i is the objective we want to optimize. There may be more elements in f equal to 1, when there are several natural objectives to optimize. v_j^{L} and v_j^{U} are the lower- and upper-bound values (thermodynamic constraints) of the flux v_j. (In our analysis, we consider $v_j^{\mathrm{U}} = 100$ and $v_j^{\mathrm{L}} = -100$ for the fluxes that represent reversible reactions.) The output of FBA is a particular distribution of fluxes, denoted by v, that optimizes the objective functions. Remarkably, FBA does not describe how a certain flux distribution is realized (by kinetics or enzyme regulation), but which flux distribution is optimal for the cell; for instance, it provides the highest rate of biomass production at a limited inflow of external nutrients. Biomass can be defined in terms of the biosynthetic requirement for the cell and is represented by a dummy reaction formulated according to experiments found in the literature.

2.2 Pathway-oriented SA

In order to allow our algorithm to work at the genetic level, we use the gene–protein–reaction (GPR) mappings. GPR mappings provide the links between each gene and the reactions v_j that depend on it and define how certain genetic manipulations affect reactions in the metabolic network. For a set of L genetic manipulations, the GPR mappings are represented by a $L \times n$ matrix G, where the (l,j)th element is 1 if the lth genetic manipulation maps onto the reaction j, and is 0 otherwise. GPR associations distinguish between single- and multi-functional enzymes, isoenzymes, enzyme complexes, enzyme subunits, so that they capture the complexity and diversity of the biological relationships through a Boolean approach. We used the approach implemented in OptKnock (Burgard *et al.*, 2003) to find the fluxes distribution that reproduces the desired productions (synthetic objectives) and achieves the maximal growth. The bi-level problem is represented as

$$\begin{array}{lll} \max & g'v & \\ \text{such that} & \displaystyle\sum_{l=1}^{L} y_l \leq C & \\ & y_l \in \{0, 1\} & \\ & \max & f'v \\ & \text{such that} & Sv = 0 \\ & & (1-y)'G_j v_j^{\mathrm{L}} \leq v_j \leq (1-y)'G_j v_j^{\mathrm{U}}, \\ & & j = 1, \ldots, n, \end{array} \tag{2}$$

where g is a vector of weights (n dimensional) associated with the synthetic objectives and $g\prime$ is its transpose. For example, when the synthetic objectives v_j and v_h have to be maximized, the weights g_j and g_h are equal to 1. y is the knockout vector of L bits. If there are no impaired reactions in the metabolic network, y contains only zeros. Conversely, when $y_l = 1$, the gene set embroiled in the manipulation l is turned off, and the corresponding reactions are in the absent status (the lower and upper bounds are set to zero, resulting in a modified metabolic network). C is an integer representing the maximum number of knockout allowed. The bi-level problem can be converted to a MILP problem (for a detailed description, see the original work by Burgard *et al.*, 2003). We implemented and solved the problem using the GLPK solver.

In PoSA, the knockout vector y used to represent the genetic manipulations is partitioned in p subsets of bits $\{b_1, b_2, \ldots, b_s, \ldots, b_p\}$. Each subset b_s includes the genetic manipulations linked to the reactions involved in the sth metabolic pathway of the network. Each subset b_s has a cardinality W_s, where $W_s < L$, $\forall s = 1, \ldots, p$. The genes are clustered in metabolic pathways as reported by Feist *et al.* (2007). Each pathway performs a particular task in the metabolism, e.g. the *citric acid cycle*, the *oxidative phosphorylation*, the *pentose phosphate pathway* and so on. PoSA takes also into account the eventuality that a reaction belongs to different pathways: when the gene responsible for that reaction is knocked out, the reaction is impaired in all its pathways. We generated the gene-pathway (GP) mappings, defined by the $L \times p$ matrix P, where the (l, s)th element of P is 1 if the lth genetic manipulation is linked to the reactions involved in the sth functional pathway, and 0 otherwise. We also adopted the reaction-pathway (RP) mappings, mathematically described by the $n \times p$ matrix R, where the (j, s)th element of R is 1 if the jth reaction is part of the sth functional pathway, and 0 otherwise. The *E.coli* model used for our analysis has $p = 36$ functional pathways.

For the combinatorial problem described in (2), we define the 'elementary effect' (Morris, 1991) for the input b_s as

$$\mathrm{EE}_s = \frac{F(b_1, b_2, \ldots, b_{s-1}, \tilde{b}_s, b_{s+1}, \ldots, b_p) - F(\tilde{y})}{\Delta_s}, \qquad (3)$$

where $\tilde{b}_s$ is the mutation on the input b_s and consists of the *switching* of bits chosen randomly in b_s: if a bit is equal to 0 (or 1), the permutation turns it into 1 (or 0). Δ_s is a scale factor defined as:

$$\Delta_s = \frac{1}{W_s} \sum_{i=1}^{W_s} \tilde{b}_s(i), \forall s = 1, \ldots, p. \qquad (4)$$

The output $F(y)$ considered in PoSA is the vector v of fluxes. $\tilde{y}$ is the mutation performed on the knockout vector y defined in the Boolean region of interest $\Omega = \{0, 1\}^{\mathrm{L}} = \left\{(y_1, \ldots, y_l, \ldots, y_{\mathrm{L}}) | y_l \in \{0, 1\}\right\}$.

The distribution of effects EE_s is obtained by permuting y through a random sampling of KQ points from Ω and permuting b_s by randomly sampling KQN points from Ω. If the procedure was performed for each input, the result would be a random sample at a total cost of KQ for calculating $F(\tilde{y})$ and KQN for $F(b_1, b_2, \ldots, \tilde{b}_s, \ldots, b_p)$, with a total cost of pKQ$(N + 1)$ evaluates of function. As regards the details, in the Supplementary Material we report the code and pseudo-code of the algorithm.

The estimation of the mean μ^* and the standard deviation σ^* of the distribution of the elementary effects will be used to detect those inputs that should be considered influent in the model. A high μ^* indicates an input with an important 'overall' influence on the output, while a large σ^* indicates an input whose influence is highly dependent on the values of the inputs (Morris, 1991).

2.3 Optimization through genetic manipulation

GDMO is a combinatorial global search method that finds the genetic manipulation strategies to simultaneously optimize multiple cellular functions. The simultaneous optimization of multiple objectives differs from the single-objective optimization because the solution is not unique when the objectives are in conflict with each other. For instance, the knockout strategy able to ensure the production of a metabolite alters the biomass formation and the ability of the organism to reproduce itself. Therefore, metabolite production and biomass formation are strongly in conflict. The solution of a multi-objective problem is a potentially infinite set of points, called Pareto-optimal solutions or *Pareto front*. A solution is said to be Pareto optimal if there exists no feasible solution for which an improvement in one objective leads to a simultaneous improvement in one (or more) of the other objectives. Formally, a point y^* in the solution space is said to be Pareto optimal if there does not exist a point y such that $F(y)$ dominates $F(y^*)$, where F is the vector of Z objective functions. In our case, the space of variables (i.e. the domain of y) is discrete.

Our method implements a genetic algorithm inspired by NSGA-II (Deb *et al.*, 2002) and is composed of four key steps. We start with the initialization of the population Pop and the computation of the fitness score. The population can be initialized in different ways: randomly or assigning present status to all genes or selecting a set of knocked out genes. The population Pop is represented by a $I \times (L + Z + 2)$ matrix, where I is the number of individuals, L is the number of decision variables and Z is the number of objective functions. The last two columns are used to store two parameters of the algorithm linked to each individual: the rank and the crowding distance (Deb *et al.*, 2002). The values of the objective functions are calculated by solving the combinatorial problem (2). Each individual represents a feasible solution, composed of the proposed knockout strategy $\tilde{y}$. The fitness score is computed after sorting according to the level of non-domination. Each individual is assigned a rank, and between two solutions with different non-domination ranks, we prefer the solution with the lowest rank (Deb *et al.*, 2002).

Successively, three steps are iteratively performed. In a *binary tournament selection* process, two individuals are selected at random, and their fitness is compared. The individual with the best fitness is selected as a parent. The algorithm selects a number of parents (i.e. the best individuals) equal to the half of the population. Parents are mutated using a *combinatorial mutation operator* to create an offspring population. A mutation represents a switch, from 0 to 1 or from 1 to 0. The process is randomly executed; for each parent individual, we create ten offspring, but only the best is chosen. Mutations can achieve the maximum knockout number equal to the parameter C (fixed at 50 by default). A new population of I individuals is formed selecting the best individuals from the parents of the previous generation and the current offspring. The new population undergoes a new round of evaluation. Finally, a *selection* operator is performed in order to reach the last front. For each generation of the algorithm, the Pareto-optimal solutions are provided.

This cycle is repeated until the number of generations reaches its upper bound. The number of generations D and individuals I are parameters chosen by the user. After calculating the Pareto-optimal solutions, we perform a post-processing filtering, in order to eliminate redundant knockouts that are not, in fact, necessary for the achievement of the selected production and biomass level.

The time complexity of the genetic algorithm is $O(ZDI^2)$, where Z is the number of the objectives, D the number of generations and I the population size.

2.4 FBA using experimental conditions

The gene expression data provide several information on the activation of genes when the organism undergoes specific external stimuli. In a first approximation, we may transform microarray data matrix in a Boolean matrix, where 0 represents the knockout condition for a gene and 1 represents the activation. Our framework is able to read gene expression data, transfer them to a metabolic model and evaluate *in silico* the metabolic fluxes distribution using FBA. In this way, it is possible to investigate the behavior of an organism as well as to compare different experimental conditions. It could be interesting, in future developments, extending the exploration analysis from a binary domain to a quantitative domain, evaluating the gene expression in the metabolic network. In addition, through our optimization method, we can deduce how the growth of the organism improves in a given experimental condition, when additional genes are turned off (or on).

2.5 Pareto front ε-dominance

Another analysis that we perform is inspired by the idea of Laumanns *et al.* (2002), namely to use a condition of approximated dominance for their evolutionary multi-objective algorithm with the aim of improving the diversity of solutions and the convergence. However, we use this idea to perform a post-processing analysis in order to calculate an approximated Pareto front. This way, we search for solutions that may have been discarded because they are dominated by a small amount ε that, for our purposes, can be considered negligible. Therefore, once the optimization routine has been performed, all the sampled points are revisited. Then, a new set of solutions is built, called 'ε-non-dominated' points set, by applying a 'relaxed' condition of dominance, called ε-dominance. In a formal way, assuming that all the objective functions are positive and must be maximized, given $\varepsilon > 0$, we seek all the points (solutions) w belonging to the set: $\{w : F_r(w) - \epsilon \geq F_r(u),\ \forall\, r = 1, ..., Z\}$, where F is the vector of the Z objective functions and u represents all the others sampled points. This set will contain both the new 'ε-non-dominated' solutions and the old non-dominated ones. The results are shown in Figure S8 of the Supplementary Material.

2.6 Robustness analysis

After the optimization, the validity of the biological strain, designed *in silico*, must be tested, and this is performed by the *RA*. In this way, we assess the ability of a strain to adapt to small perturbations that can occur at any stage of the biochemical processes, either within the bacterium or caused by the environment in which it reproduces itself.

The basic principle of this analysis is the following. First, we define the perturbation as a function $\tau = \gamma(\Psi, \sigma)$, where γ applies a stochastic noise σ to the system Ψ and generates a trial sample τ. The γ-function is called γ-perturbation. Without loss of generality, we assume that the noise is defined by a random distribution. We generate a set T_τ of trial samples τ in order to render the calculation of robustness statistically meaningful. Each element τ of the set T_τ is considered robust to the perturbation for the stochastic noise σ and the given property φ if the following condition is verified:

$$\rho(\Psi, \tau, \varphi, \delta) = \begin{cases} 1, & \text{if } |\varphi(\Psi) - \varphi(\tau)| \leq \delta \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where Ψ is the *reference system*, φ is a *metric* (or property), τ is a *trial sample* of the set T_τ and δ is a *robustness threshold*. The definition of this condition makes no assumptions about the function φ.

The robustness of a system Ψ is defined as the number of robust trials of T_τ with respect to the total number of trials $|T_\tau|$. The robustness index is a function of δ, so the choice of this parameter is crucial. Since we are interested in the behavior of strain when subjected to small perturbations, and since the behavior is acceptable when the deviations from the original value is as small as possible, we chose the values of ε equal to 1% of the metric and σ equal to 1% of the perturbed variable.

According to the principle above reported, we implement the global, local and PoRA methods. In our analysis, we perturb the upper v_j^{U} and lower v_j^{L} bounds, $j = 1, \ldots, n$ of the metabolic fluxes. In particular, in GR the perturbation is performed simultaneously for all the fluxes of the network to evaluate the fragility of the complete organism. In LR, the perturbation is performed for each flux (hence we have a robustness index for each flux), whereas in PoRA, the perturbation is performed simultaneously for all the fluxes clustered in a metabolic pathway obtaining a robustness index for each pathway. The results are reported in Table 1 and in the Supplementary Material (Tables S1 and S2).

2.6 Glocal analysis

We also implement the analysis described by Hafner *et al.* (2009) to compare the results obtained by the GR and LR analyses. In the glocal analysis, the authors implement a procedure that calculates the volume occupied by those parameters such that the system maintains the desired characteristics. The volume is computed in the $2n$-dimensional parameter space. In our case, the volume is such that Equation (5) holds. Since this research requires a huge computational effort, given the high number of dimensions (R^{2n}, where $2n$ is the number of parameters), it is guided by an iterative procedure that involves the principal component analysis. Then, they calculate local coefficients and use them to derive the key parameters of the robustness (through the

Table 1. Comparison between GDMO and previous genetic design methods

	Wild type	OptFlux	OptGene	GDLS	GDLS	OptKnock	OptKnock	GDMO	GDMO	GDMO
Acetate	8.30	15.129 (+82.3%)	15.138 (+82.4%)	15.914 (+91.7%)	n.a. n.a.	n.a. n.a.	12.565 (+51.4%)	13.791 (+66.13%)	19.150 (+130.7%)	n.a. n.a.
Succinate	0.077	10.007 (+12877%)	9.874 (+12704%)	n.a. n.a.	9.727 (+12514%)	9.069 (+12362%)	n.a. n.a.	n.a. n.a.	n.a. n.a.	10.610 (+13659%)
Biomass	0.23	n.a. n.a.	n.a. n.a.	0.0500 (−78.4%)	0.0500 (−78.4%)	0.1181 (−77.9%)	0.1165 (−49.6%)	0.130 (−43.72%)	0.053 (−77.10%)	0.087 (−62%)
kc	n.a.	n.a.	n.a.	14	26	54	53	**3**	10	**8**
GR (%)	54.76/53.68	n.a.	n.a.	13.76	16.6	43.24	43.08	45.32	27.6	40.40
LR (%)	54.0/54.67	n.a.	n.a.	16.0	21.33	40.0	40.60	39.33	24.0	46.0
R	1.30/1.34	n.a.	n.a.	1.45	1.45	1.18	1.02	0.78	0.44	1.32
PoRA (%)	100.0/99.33	n.a.	n.a.	19.33	28.67	87.33	76.67	81.33	43.33	83.33

We compare OptFlux (Rocha *et al.*, 2008), OptGene (Patil *et al.*, 2005), GDLS (Lun *et al.*, 2009), OptKnock (Burgard *et al.*, 2003) and our multi-objective optimization algorithm (GDMO) to maximize acetate (Ac) and succinate (Suc) productions [mmolh^{-1} gdW^{-1}]. The second column reports the amounts of acetate, succinate and biomass when all the genes are turned on. The third and fourth rows show the biomass [h^{-1}] and the knockout cost (kc). The last four rows show a comparison between the RA methods. The two values of robustness reported for wild type are referred, respectively, to the productions of Ac and Suc. *R*-values (Hafner *et al.*, 2009) and GR-values are GR indices. The strain is more robust when R and GR detect high values. For LR and PoRA, we report the minimum value found, which is associated with the less robust flux (glucose uptake rate) and the less robust pathway (energy metabolism). 'n.a.' stands for *not applicable*.

Spearman's partial correlation coefficient). In the global part, we found that the results are comparable in most cases with our global metrics. The results of the analysis are reported in Table 1 (R versus GR). In the local part, we found that the most influential flux is also the one that obtains the minimum LR value.

2.7 Quantitative and qualitative knockout analysis

Pareto optimality gives information about the trend of organisms in their ability to produce particular metabolites, as reported in the previous section. In addition, we color the Pareto points in order to obtain a map of the knockout cost dispersion (Figure S8B–D of the Supplementary Material). In this way, each point characterizes both the phenotype of an organism (for instance, the amount of acetate and biomass) and the genotype (in terms of how many genes are knocked out). Nevertheless, it is also important to give a qualitative score for each knockout strategy. We have two sensitivity measures: μ^* (mean) and σ^* (standard deviation). Large μ^* indicates high overall influence, high linear effect, while large σ^* indicates that either the specific input is involved with other inputs, or its effect is non-linear or non-additive. According to the μ^* sensitive index obtained by PoSA, we assign a quality score (QS) for each strain. Strains that have genetic manipulations involved in pathways with low μ^* values are preferred, and thus get a high score. The score ranges from 0 to 1; 0 when the genetic strategy involves gene sets linked to the pathway with the largest μ index, that is the most sensitive and 1 when the genetic strategy involves gene sets linked to the pathway with the lowest μ index. The score is normalized by the square root of the number of samples, since manipulations involve different knocked out genes. Consequently, if we find two Pareto solutions through GDMO that have the same phenotype and different knockout strategies, we are able to choose the best solution in terms of knockout, according to the QS calculated using PoSA. For the strains reported in Table 1, we obtained from left to right the QS equal to 0.285, 0.063 and 0.223.

2.8 Implementation

We implement GDMO, SA and RA using MATLAB and GLPK (GNU Linear Programming Kit). We illustrate the capabilities of GDMO by applying it to several overproduction problems in iAF1260 *E.coli* (Feist *et al.*, 2007). In a pre-processing analysis, we perform a reduction of the FBA network to remove duplicate and dead-end reactions as described by Burgard *et al.* (2001), Pharkya and Maranas (2006) and Mahadevan and Schilling (2003). After the reduction, the resulting metabolic network is mathematically identical to the original network. Initially, in *i*AF1260, there are $n = 2382$ reactions, $m = 1668$ metabolites and $L = 913$ gene sets; after the reduction, we obtain $n = 959$, $m = 483$ and $L = 632$ in anaerobic conditions and $n = 1019$, $m = 506$ and $L = 663$ in aerobic conditions. In particular, for acetate and succinate production, we performed experiments in both anaerobic and aerobic conditions, with 10 and 5 mmolh^{-1} gDW^{-1} of available glucose.

3 RESULTS AND DISCUSSION

Taking into account that each gene is assigned to at least one of the 36 different pathways in the metabolic network, PoSA evaluates the importance of a pathway on the basis of the knockouts that are involved in its metabolism and indicates a ranking of the metabolic pathways in the (μ^*, σ^*) space reported in Figure 2.

The study of the variance-to-mean ratio (VMR) is a good measure of the degree of randomness of a given phenomenon. In the *E.coli* analysis, PoSA-sensitive μ^* and σ^* indices are linked with a linear relationship and the VMR is >1; thus, the elementary effects set is said to be over-dispersed, highlighting the presence of great variability. We can deduce that the elementary effects of the 36 pathways are sampled from a negative binomial distribution. The VMR is linked to the Pareto front and can be harnessed to explore the solution space, since it describes the probability distribution of the phenomenon. In general, highly networked cell components (such as those for nucleic

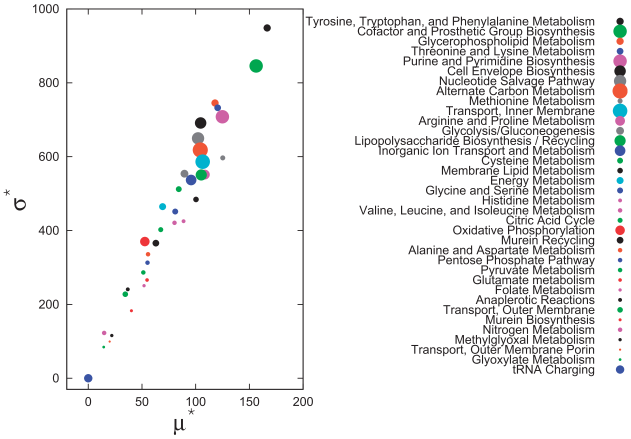

Fig. 2. PoSA for model of iAF1260 *E. coli*. The model is composed of 36 pathways whose reactions and genes are clustered according to the functionality of the pathway to which they belong. The size of a sign is proportional to the number of genes involved in the pathway

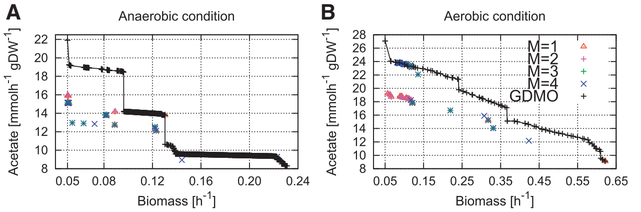

Fig. 3. Performance of GDMO. Maximization of biomass and acetate production in anaerobic (**A**) and aerobic (**B**) conditions, with glucose uptake rate 10 mmolh^{-1} gDW^{-1} in iAF1260 *E.coli*. The Pareto fronts obtained by GDMO are in black, and the results obtained by GDLS in red, purple, green and blue, set with $M = 1$, 2, 3, 4 and $k = 1$, 2, 3, 4, respectively. M and k are parameters of GDLS and define, respectively, the number of solutions proposed and the maximal number of neighborhood genes to knock out for each iteration of the algorithm. For a detailed description, see the original work (Lun *et al.*, 2009)

acids, amino acids, cofactors and energetic metabolism) are in the top right corner, while specific, often single reaction very abundant components (such as those for bacterial walls, nitrogen, glutamic and carbohydrates) are in a bottom left position. Results in Figure 2 were obtained by evaluating more than 3 million calls to the function F of Equation (3).

In Table 1, we report the genetic strategies obtained by GDLS (Lun *et al.*, 2009), OptFlux (Rocha *et al.*, 2008), OptGene (Patil *et al.*, 2005), OptKnock (Burgard *et al.*, 2003) and GDMO, where the goal is to optimize succinate and acetate productions in the *E.coli* metabolic network. To compare these methods, we run OptKnock, while the GDLS, OptGene and OptFlux solutions are extracted by published data by Lun *et al.* (2009).

We run GDMO initializing the *E.coli* network with an empty set of knockout, i.e. in wild-type configuration, and setting the population size $I = 1000$ and the number of generations $D = 1500$. Table 1 reports the best solutions in terms of acetate and succinate obtained by the previous methods, along with our proposed solutions. Details of the genetic strategies are reported in Tables S1 and S2 of the Supplementary Material. We used the multi-objective optimization method to maximize the production of metabolites of interest and biomass, minimizing simultaneously the knockout cost. Table 1 also reports the robustness indices for the wild-type organism and strains. The effect of the knockouts on the robustness of the network can be noticed by comparing the GR, LR and PORA values of strains with those of the wild type. The values of GR and LR are of the same order of magnitude, probably because the robustness of the network is strongly linked to the glucose uptake rate.

In Figure 3, we show the comparison between the results obtained by the method proposed by Lun *et al.* (2009) and our Pareto solutions for optimizing acetate production (in the Supplementary Material, we report also the results for succinate production). The solutions provided by GDLS do not outperform Pareto fronts, since they occupy positions in the area under the Pareto curves. In the best cases, they lie on the Pareto fronts. Some suggested solutions and several optimization experiments have been reported as Supplementary Material (Tables S1 and S2 and Figs S2 and S3). In addition, the ε-dominance analysis reveals other interesting points. For instance, we find 14.05 mmolh^{-1}gDW^{-1} of acetate with a knockout cost equal to 5 and 9.175 mmolh^{-1} gDW^{-1} of succinate with a knockout cost equal to 5.

In order to study the favorable environmental conditions (flux design), i.e. nutrients for *E.coli*, we performed the simultaneous optimization of acetate, succinate and biomass on the complete network, i.e. without knockouts. We considered the anaerobic and aerobic condition (O_2 uptake rate = 10 mmolh^{-1} gDW^{-1}) and maintained fixed the glucose uptake rate at 10 mmolh^{-1} gDW^{-1}. We used NSGA-II (Deb *et al.*, 2002) to perform the optimization by exploring the continuous space of exchange fluxes. In our analysis, we perturbed the thermodynamics constrains v_j^L, $j = 1, \ldots, n^{ex}$, where n^{ex} is the number of the exchange fluxes. The decision variables are real values from 0 to −100 (0 when the uptake is not allowed and −100 when the potential uptake rate is 100 mmolh^{-1} gDW^{-1} h^{-1}). Only glucose and oxygen were kept constant. Setting the population size at 100, we ran NSGA-II for 500 generations. In Figure 4, we show the results of the optimization in aerobic and anaerobic conditions (the observed Pareto front and the observed feasible points). In anaerobic condition, we found 100 mmolh^{-1} gDW^{-1} h^{-1} of acetate, 42.918 mmolh^{-1} gDW^{-1} of succinate and 3.6204 h^{-1} of biomass (the trade-off). In this condition, we noticed a significant increment in the L-aspartate, citrate, lactose, fumarate and malate uptake rates. In aerobic condition, we found 100 mmolh^{-1} gDW^{-1} h^{-1} of acetate, 21.889 mmolh^{-1} gDW^{-1} of succinate, 4.16 h^{-1} of biomass and a significant increment in the L-asparagine, 1, 4-alpha-D-glucan, Fe(III)dicitrate, 2-oxoglutarate uptake rates. In our analysis, we perturbed simultaneously almost all the exchange fluxes, but it is possible to select a smaller set of nutrients according to experimental requirements.

Pareto fronts provide significant information in metabolic design automation. The size of non-dominated solutions, the first derivative and the area under the curve are important markers for the best design within the same organism or between different organisms. Jumps correspond to sudden decreases in the availability of entire pathways and sub-networks when a crucial hub is eliminated, for instance the elimination of Krebs cycle or other key biosynthetic hubs. The area under the Pareto

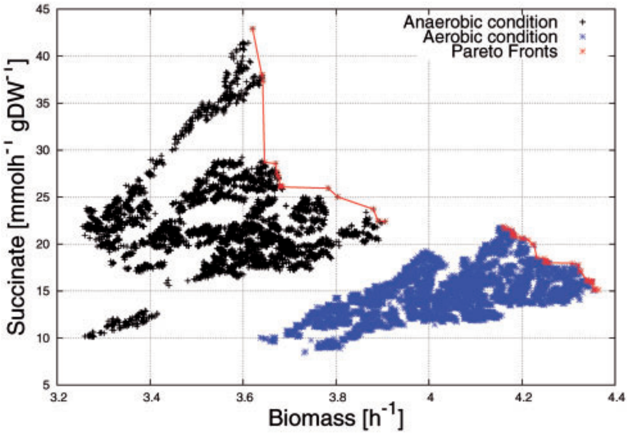

Fig. 4. Optimization of acetate production, succinate production (y-axis) and biomass formation (x-axis). We consider the wild-type bacteria (i.e. knockout zero) and perform the maximization in aerobic (blue signs, $O_2 = 10\,\mathrm{mmolh^{-1}\,gDW^{-1}}$) and anaerobic conditions (black signs) on a basis of $10\,\mathrm{mmolh^{-1}\,gDW^{-1}}$ glucose fed to identify favorable nutrients (input fluxes). The algorithm reaches the maximum production of acetate ($100\,\mathrm{mmolh^{-1}\,gDW^{-1}}$). In red we show the Pareto fronts

front provides an estimate of the number of intermediates which may be exploited for biotechnology purposes (optimization of an additional objective) or to build synthetic pathways (synthetic biology). Given two bacteria or two conditions for the same bacterium, the highest Pareto front would probably represent the best conditions for adding or optimizing pathways leading to new biotechnology products. Pareto optimality is useful to compare the ability of different organisms for optimizing specific metabolites (Supplementary Figs S9 and S10).

Through our framework we are able to program bacteria in order to obtain desired outputs, thus framing them as living computers (Angione *et al*., 2012). The goal is to provide a simple tool to search and propose to the biotechnologist the best and suitable solutions *in silico*, so as to reproduce them *in vivo*. Our framework investigates nutrients, reactions, metabolic pathways and knockouts for bacteria and other organisms in an efficient automatic design. We are able to present several proposals and indicate the best in terms of environmental conditions, knockout cost, robustness and sensitivity. Knockout strategies are useful in synthetic biology, while simulating the FBA in a particular experimental condition, using the gene expression values is important for providing an optimization of bacteria in a given environment and biotechnological/medical condition.

Conflict of Interest: none declared.

REFERENCES

Alper,H. *et al.* (2005) Construction of lycopene-overproducing *E. coli* strains by combining systematic and combinatorial gene knockout targets. *Nat. Biotechnol.*, **23**, 612–616.

Angione,C. *et al.* (2012) Computing with metabolic machines. In Voronkov,A. (ed.) *Turing-100*. Volume 10 of EPiC Series. pp. 1–15.

Atsumi,S. *et al.* (2010) Engineering the isobutanol biosynthetic pathway in *Escherichia coli* by comparison of three aldehyde reductase/alcohol dehydrogenase genes. *Appl. Microbiol. Biotechnol.*, **85**, 651–657.

Burgard,A. *et al.* (2003) Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnol. Bioeng.*, **84**, 647–657.

Burgard,A.P. *et al.* (2001) Minimal reaction sets for *Escherichia coli* metabolism under different growth requirements and uptake environments. *Biotechnol. Prog.*, **17**, 791–797.

Cutello,V. *et al.* (2006) A multi-objective evolutionary approach to the protein structure prediction problem. *J. R. Soc. Interf.*, **3**, 139–151.

Deb,K.D. *et al.* (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*, **6**, 182–197.

Feist,A.M. *et al.* (2007) A genome-scale metabolic reconstruction for *Escherichia coli* k-12 mg1655 that accounts for 1260 orfs and thermodynamic information. *Mol. Syst. Biol.*, **3**, 291–301.

Hafner,M. *et al.* (2009) 'Glocal' robustness analysis and model discrimination for circadian oscillators. *PLoS Comput. Biol.*, **5**, 1–10.

Jarboe,L.R. *et al.* (2010) Metabolic engineering for production of biorenewable fuels and chemicals: contributions of synthetic biology. *J. Biomed. Biotechnol.*, **2010**, 1–18.

Laumanns,M. *et al.* (2002) Combining convergence and diversity in evolutionary multi-objective optimization. *Evol. Comput.*, **10**, 263–282.

Lun,D.S. *et al.* (2009) Large-scale identification of genetic design strategies using local search. *Mol. Syst. Biol.*, **5**, 296.

Mahadevan,R. and Schilling,C. (2003) The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metab. Eng.*, **5**, 264–276.

Morris,M. (1991) Factorial sampling plans for preliminary computational experiments. *Technometrics*, **33**, 161–175.

Orth,J.D. *et al.* (2010) What is flux balance analysis? *Nat Biotechnol.*, **28**, 245–248.

Patil,K. *et al.* (2005) Evolutionary programming as a platform for in silico metabolic engineering. *BMC Bioinformatics*, **6**, 308.

Pharkya,P. and Maranas,C. (2006) An optimization framework for identifying reaction activation/inhibition or elimination candidates for overproduction in microbial systems. *Metab. Eng.*, **8**, 1–13.

Rocha,M. *et al.* (2008) Natural computation meta-heuristics for the in silico optimization of microbial strains. *BMC Bioinformatics*, **9**, 499.

Schuetz,R. *et al.* (2012) Multidimensional optimality of microbial metabolism. *Science*, **336**, 601–604.

Sendin,J.O. *et al.* (2009) Multi-objective optimization of biological networks for prediction of intracellular fluxes. In Corchado,J. *et al.* (ed.) *IWPACBB 2008*. Advances in Soft Computing. Springer, Berlin, pp. 197–205.

Stracquadanio,G. *et al.* (2010) Analysis and optimization of c3 photosynthetic carbon metabolism. In Rigoutsos,I. and Floudas,C.A. (eds.) *IEEE BIBE, Philadelphia, PA, USA, May 31–June 3*. IEEE Computer Society, pp. 44–51.

Sun,J. *et al.* (2009) Genome-scale constraint-based modeling of *Geobacter metallireducens*. *BMC Syst. Biol.*, **3**, 15.

Zhangs,H.-X. and Goutsias,J. (2010) A comparison of approximation techniques for variance-based SA of biochemical reaction systems. *BMC Bioinformatics*, **11**, 246.