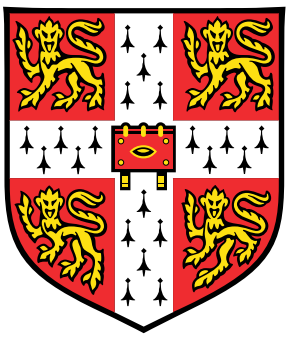
Computational methods for multi-omic models of cell metabolism and their importance for theoretical computer science



Claudio Angione

Computer Laboratory

University of Cambridge

This dissertation is submitted for the degree of

Doctor of Philosophy

King's College October 2015

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text. This dissertation contains fewer than 60,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Claudio Angione
October 2015

Acknowledgements

First, I would like to thank my supervisor, Dr Pietro Lió, not only for his guidance through all these years, but also for allowing me to grow as a research scientist.

A very special thanks to Prof Giuseppe Nicosia, whose mentoring activity and fruitful collaboration has strongly shaped my career.

I would also like to thank all the past and present members of the Computational Biology research group at the Computer Laboratory, without whom many interesting lunchtime discussions would not have taken place. Also, I am grateful to my funding bodies: Computer Laboratory, CHESS, King's college, Cambridge Philosophical Society.

Finally, a special thanks to my family because they certainly provided all the support I needed to pursue my studies abroad. This thesis is dedicated to them.

Abstract

To paraphrase Stan Ulam, a Polish mathematician who became a leading figure in the Manhattan Project, in this dissertation I focus not only on how computer science can help biologists, but also on how biology can inspire computer scientists.

On one hand, computer science provides powerful abstraction tools for metabolic networks. Cell metabolism is the set of chemical reactions taking place in a cell, with the aim of maintaining the living state of the cell. Due to the intrinsic complexity of metabolic networks, predicting the phenotypic traits resulting from a given genotype and metabolic structure is a challenging task. To this end, mathematical models of metabolic networks, called *genome-scale metabolic models*, contain all known metabolic reactions in an organism and can be analyzed with computational methods. In this dissertation, I propose a set of methods to investigate models of metabolic networks. These include multi-objective optimization, sensitivity, robustness and identifiability analysis, and are applied to a set of genome-scale models.

Then, I augment the framework to predict metabolic adaptation to a changing environment. The adaptation of a microorganism to new environmental conditions involves shifts in its biochemical network and in the gene expression level. However, gene expression profiles do not provide a comprehensive understanding of the cellular behavior. Examples are the cases in which similar profiles may cause different phenotypic outcomes, while different profiles may give rise to similar behaviors. In fact, my idea is to study the metabolic response to diverse environmental conditions by predicting and analyzing changes in the internal molecular environment and in the underlying multi-omic networks. I also adapt statistical and mathematical methods (including principal component analysis and hypervolume) to evaluate short term metabolic evolution and perform comparative analysis of metabolic conditions.

On the other hand, my vision is that a biomolecular system can be cast as a "biological computer", therefore providing insights into computational processes. I therefore study how computation can be performed in a biological system by proposing a map between a biological organism and the von Neumann architecture, where metabolism executes reactions mapped to instructions of a Turing machine. A Boolean string represents the

genetic knockout strategy and also the executable program stored in the "memory" of the organism. I use this framework to investigate scenarios of communication among cells, gene duplication, and lateral gene transfer. Remarkably, this mapping allows estimating the computational capability of an organism, taking into account also transmission events and communication outcomes.

Table of contents

List of figures

List of tables

List of Algorithms

Glossary

Metabolism Set of chemical reactions taking place in a living cell

Metabolite Reactant or product of a metabolic reaction

Genome-scale metabolic model Mathematical model that contains all known metabolic reactions of an organism

FBA Flux balance analysis, a method for large-scale steady-state analysis of cell metabolism

Stoichiometric matrix Matrix S whose entries are stoichiometric coefficients S_{ij} representing the coefficient of metabolite i in reaction j

Biomass Chemical reaction in an FBA model that represents the growth rate of the organism

Steady-state Cellular state of equilibrium where metabolite concentrations are constant and a set of nutrients are being constantly converted to generate biomass

Linear programming Method for the optimization of a linear objective function, subject to linear constraints

Gene Region of DNA that codes for a specific protein or RNA

Gene set Set of genes coding for a protein

Gene expression Process that uses the information encoded in a gene to assemble the corresponding protein

Amino acids Building blocks in the creation of proteins

Codon Triplet of DNA nucleotides that codes for a specific amino acid

Protein Molecule composed of amino acids and responsible for many cellular functions (including catalyzing metabolic reactions)

Transcription Process that copies genetic information from DNA to mRNA

Translation Process that decodes mRNA to create proteins

Flux Rate of production of substrates through metabolic reactions

Omics Various levels of biological organization, e.g., proteomic, fluxomic, transcriptomic

Gene knockout The process of making a gene inoperative

Organelle Part of the cell specialized in performing a specific function (e.g., the mitochondrion produces energy)

Affymetrix Antisense2 Type of chip used to obtain microarray profiles of gene expression

Chapter 1

Introduction

Understanding the role of individual components in a biological system is an important step to elucidate or predict its behavior. However, a major theme in the discipline called *systems biology* is investigating these systems with an integrated approach. In fact, studying the interactions between different components also gives insights into the functioning and behavior of the components taken independently [90].

1.1 Systems biology and metabolic networks

In systems biology, the value of modelling a biological system is not merely explanatory of a biological process, but also predictive. A model can be used to suggest hypotheses that can be tested, or to pinpoint unexpected behaviors that can be further investigated *in vitro*. Systems biology was pioneered by Tomita et al. [161], who provided the first quantitative model for the simulation of a full cell metabolism. More recently, a research effort by Karr et al. [85] provided, for the first time, a whole-cell computational model of the life cycle of a small pathogenic bacterium, *Mycoplasma genitalium*. The model includes metabolism, replication of the genome, and cell division.

The term *metabolism* refers to the set of chemical reactions taking place in living cells, with the aim of maintaining cellular functions. Once considered only a passive result of the state of a cell, metabolism is now widely recognized as a main contributor to the cell behavior. In the last 25 years, high-quality genome-scale models (also called *reconstructions*) of metabolic networks have been combined with constraint-based optimization in order to analyze microorganisms at steady state. On one hand, metabolic reconstructions of bacteria have been developed to facilitate the study and manipulation of biochemical processes [124], allowing the bio-production of valuable compounds to be optimized through metabolic engineering [89]. On the other hand, the study of human metabolism is becoming increasingly

important for biomedical applications as an approach for understanding many diseases and aspects of health. This is enabled by the availability of high-quality reconstructions such as Recon2 [159], which integrates extensive metabolic information from various resources.

Systems-based approaches have been successfully applied over the last decade to investigate metabolic networks, composed of a set of chemical reactions and a pool of metabolites (both reactants and products of the chemical reactions). There are different methods to represent a metabolic system, for instance by using a set of linear equations or ordinary differential equations (ODEs). Each variable represents the variation of a metabolite concentration, in a dynamic or steady state, where the metabolite concentration depends on the rates of the reactions that produce and consume that metabolite. Each flux can also be modeled by using kinetic parameters [87]. However, ODE-based systems often contain a large number of equations (differential or algebraic), and solving the problem analytically is often very hard. This bottleneck recently led to the increasing use of steady-state models analyzed with linear methods. Flux balance analysis (FBA) is the most widely used method of this class.

FBA is in fact the most widely used constraint-based technique to predict flux distributions and network capabilities in large biochemical networks [24]. FBA has proved useful thanks to its ability to handle large networks: it requires information about biochemical reactions and stoichiometric coefficients, but does not involve kinetic parameters. This makes it well suited to studies that enumerate and characterize perturbations such as different substrates or genetic interventions (e.g., knockouts) leading to obligatory coupling between the growth rate and the overproduction of a desired metabolite [167]. In general, FBA is a very powerful tool for predictions of cell behavior under different metabolic conditions.

More specifically, FBA is a linear programming technique that models the steady-state condition in a chemical reaction network (see Section 1.2) [120]. The combination of flux balance constraints and capacity constraints on reaction fluxes is a system of linear homogeneous equations and inequalities; thus, its solution space is a convex polyhedral cone representing the feasible flux distributions. Intuitively, the steady-state constraints used in FBA can be thought of as Kirchoff's laws applied to any node representing a metabolite in the network: the flux through each metabolite in the network must be constant, namely the input flux must equal the output flux.

Although FBA provides a steady-state representation of metabolism, as biologists would agree, there is no biology except in the light of evolution [55]. Much of the uncertainty about the behavior of a microorganism is due to the lack of statistical bioinformatics methodologies for accurate measure of adaptability to different environmental conditions and over time [65, 137]. Approaches involving both mathematics and bioinformatics would

benefit from the study of the molecular response to adaptation. In turn, this would enable to discover the relation between environmental ("external") conditions and changes in the metabolic-phenotypic networks (the "internal" environment). At the same time, it would elucidate the genotype-phenotype relationship, which is still an open problem in biology.

As a result of many recent research efforts to elucidate the relation between genotype and phenotype, we currently have models for a better understanding of the individual components, but arguably a less clear picture of the interactions between the biological components that result in a given phenotype [24]. We still have, moreover, limited knowledge about how to use these models to predict a phenotypic response to a changing environmental condition, due to the lack of comprehensive data across different conditions and accurate training processes performed on the models [35].

Measurements of gene expression level are able to generate transcriptional profiles of microorganisms across a diverse set of environmental conditions. Databases of environmental conditions have been recently produced for several organisms, including *Escherichia coli* [58], *Clostridium* [98], *Salmonella* [80], and fission yeast [96]. Although such resources, coupled with statistical analysis, remain key to the interpretation of measured data, they do not provide a comprehensive understanding of the resulting cellular behavior. For instance, similar gene expression profiles may cause different phenotypic outcomes, while different environmental conditions may give rise to similar behaviors. Additionally, the actual response to a given condition is highly dependent on the multiple cellular objectives that the microorganism is required to meet [13, 63].

In particular, modifications of FBA are able to reduce the problem of determining the metabotype (the fluxes through all reactions in the system) under a given condition and gene expression profile to a tractable linear program under the assumptions of steady-state and optimality. Due to their scalability and precision, these methods have been used widely, for example to predict growth phenotypes in specific environmental conditions [139].

The overall functioning of an organism due to changes in the environmental condition affects various levels of cellular organization. Complex alterations involve also gene expression and metabolism [92]. In both metabolic engineering and disease studies we often seek the gene expression profiles that could lead to some desired metabolic state. This amounts to solving the inverse problem of what is typically addressed through FBA, but computational methods suitable for such studies are still lacking [154].

In this dissertation, we propose a set of methods to analyze and optimize genome-scale metabolic models. The recent availability of high-throughput data regarding multiple layers of biological organization (called *omics*) allows mapping cellular processes at the levels of mRNA, proteins, and metabolites. The aim of this dissertation is also to propose methods

aimed at improving the predicting capability of a metabolic model and elucidating the genotype-phenotype relationship by including multiple biological layers in the model. We specifically focus on gene expression and codon usage.

Genome-scale models. Genome-scale metabolic models contain all known metabolic reactions in an organism. A genome-scale model, which represents the starting point for FBA, is built using the following process. First, a draft reconstruction is obtained using annotated enzyme, reaction and pathway data from databases like KEGG [84] and EcoCyc [86]. Details on which genes control each reaction are also included. Then, a literature review improves the draft reconstruction. At this stage, the FBA representation of the model is built. Finally, the model is run and validated by comparing its predictions with existing experimental results, and new experiments are performed to further improve and validate the model.

1.2 Flux balance analysis

A central role in systems biology is played by mathematical models, possibly allowing for a comprehensive analysis of complex biological systems. The idea of flux balance analysis (FBA) is to start from a system of ODEs – one for each chemical in the system – and then supposing a steady state. This assumption enables the use of linear systems and linear programming. As well as permitting the simulation of large, usually genome-scale, systems in a few seconds of CPU time, this approximation has the additional advantage of facilitating the introduction of additional layers of experimental data that can be added to the model. These layers, e.g., proteome, fluxome or transcriptome data, are usually referred to as omics (this topic will be extensively covered in Chapter 3).

FBA is widely used in systems biology to quantify the entire metabolic steady-state of a cell and calculate its flux distribution. In FBA, all known metabolic reactions in a given cell are considered, and they are mathematically described in a way that allows simulation of various states and configurations of the chemical reaction network.

As introduced above, FBA is based on two main assumptions:

- **Homeostatic assumption**: the organism has reached a steady state where the metabolite concentrations are constant and a set of nutrients are being constantly converted to generate biomass;

- **(Multi-level) optimality**: in each state, the organism tends to maximize one or multiple objectives, usually related to growth, biotechnologically-relevant compounds (e.g., acetate exchange) and important energy-carrying molecules (e.g., ATP).

FBA is usually applied to genome-scale models, built as described in Section 1.1. From the beginning, FBA was widely used for metabolic engineering of microorganisms to enhance the production of biotechnologically or industrially relevant chemicals. Recently, predictions of flux distributions, combined with experimental methods, have been successfully used to formulate novel biological hypotheses. For instance, FBA has been used to generate tissue-specific models of human metabolism [118], and to identify novel therapeutic targets against infections [98].

1.2.1 Mathematical formulation of FBA

FBA is suitable for analyzing the flow of metabolites through a metabolic network (e.g., their formation and degradation, transport and cellular utilization). Mathematically, in a metabolic network, the derivative of a metabolite concentration can be computed through a linear combination of the input and output reaction fluxes, which produce and consume (respectively) that metabolite. Let the network be composed of m metabolites with concentration x_i, $i = 1,\ldots,m$ and n reactions with flux rates v_j, $j = 1,\ldots,n$. The balance that metabolite concentrations x_i must satisfy is

$$\frac{dx_i}{dt} = \sum_{j=1}^{n} S_{ij} v_j, \quad i = 1,\ldots,m, \tag{1.1}$$

where S_{ij} is the stoichiometric coefficient of the ith metabolite in the jth reaction. A negative (positive) coefficient S_{ij} in the stoichiometric matrix indicates that the ith metabolite is consumed (produced) in the jth reaction. A metabolite that is not part of the reaction, has a zero entry for the corresponding column. Since each metabolite is part of a very limited number of reactions in the network, S is a sparse matrix.

Under steady state conditions $\frac{dx_i}{dt} = 0$, $\forall i$, the balance for the ith metabolite is $\sum_{j=1}^{n} S_{ij} v_j = 0$ (homeostatic assumption). Therefore, at steady state, the balance equation is $Sv = 0$, where S is the stoichiometric matrix (m rows and n columns), and v is the vector of the flux rates (metabolic and transport fluxes).

In a metabolic network, metabolites/nodes are linked by reactions/edges. In FBA each node (i.e., metabolite) of the metabolic network represents a constraint, while the reaction rates v_j represent the variables. Since the matrix S is not square and $n > m$, we have more variables than constraints, and therefore a plurality of solutions. Each solution is a

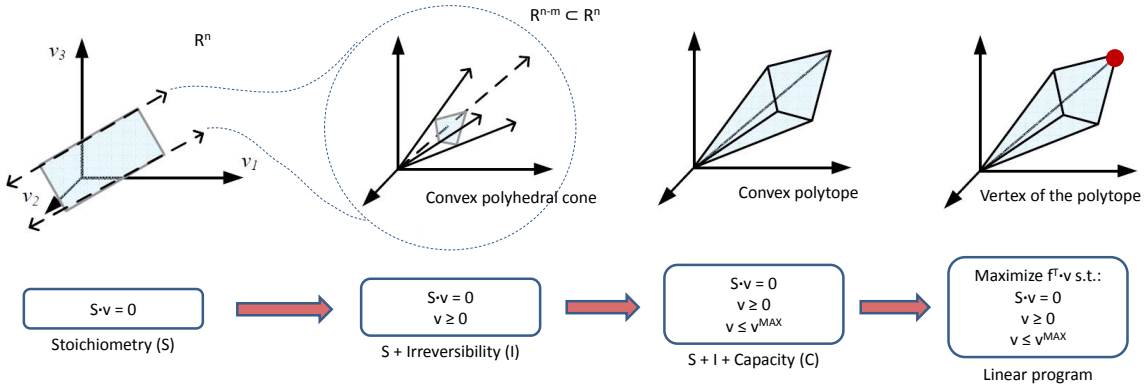


Fig. 1.1 **Mathematical formulation of flux balance analysis.** The stoichiometric matrix S of n reactions and m metabolites restricts the search of possible flux distributions to the hyperplane $\mathbb{R}^{n-m}$. Thermodynamic constraints (irreversibility of reactions) limit the space of feasible solutions, which becomes a polyhedral cone. Capacity constraints (enzyme or transport capacities) constitute an upper bound for the flux rates. If this is available for every flux in the network (more specifically, it is sufficient that the upper bound is available for the edges of the cone), the feasible space reduces to a convex polytope. Once an objective function has been defined, the final linear program finds the final flux distribution as a vertex of the convex polytope, and then reconstructs the solution in the initial space $\mathbb{R}^n$.

distribution of fluxes v and represents a particular metabolic steady state. The FBA approach seeks the metabolic state that optimizes a linear combination of flux rates, considered as objective functions (optimality assumption):

$$\begin{aligned} &\text{maximize (or minimize)} && f^{\mathsf{T}} v, \\ &\text{such that} && Sv = 0, \\ & && v_j^L \leq\ v_j \leq\ v_j^U, \quad j = 1, \ldots, n, \end{aligned} \tag{1.2}$$

where f is a n-dimensional column vector of weights, and f^{T} is its transpose. There may be more than one element in f equal to 1, when there are several flux rates to optimize; this is equivalent to optimizing a linear combination of them. The bounds v_j^L and v_j^U are the lower and upper bound values for the flux v_j, and represent thermodynamic constraints. In our analysis, we consider $v_j^U = 1000$ and $v_j^L = -1000$ for the fluxes of reversible reactions, and $v_j^U = 1000$ and $v_j^L = 0$ for the fluxes of irreversible reactions.

As shown in Fig. 1.1, the constraints in FBA consist of a set of linear equations representing flux balance constraints, plus a set of inequalities representing thermodynamic and enzyme capacity constraints. The system is solved using linear programming. This homogeneous system of m equations and n variables is usually highly underdetermined, since the number of reactions is greater than the rank of the stoichiometric matrix, unless

the value of some reaction fluxes is already known [91, 101]. The null space matrix V is defined as the matrix whose columns constitute a complete set of linearly independent solutions of $Sv = 0$. Since every linear combination of solutions of $Sv = 0$ is also a solution, the columns of V span the kernel (null space) of S.

The output of FBA is a particular distribution of fluxes, denoted by $v \in \mathbb{R}^n$, that optimizes the objective functions. We remark that FBA does not describe how a certain flux distribution is obtained (by kinetics or enzyme regulation), but finds which flux distribution is optimal for the organism when the maximization of specific objectives is required.

1.2.2 Selection of FBA objectives

The selection of a reaction flux as an objective to maximize in FBA is done through the vector f in (1.2). The most commonly chosen is the metabolic reaction representing biomass formation, which is scaled to represent the exponential growth rate of the bacterial culture. This choice is motivated by the assumption that cells maximize the growth rate as a priority, to ensure survival. The biomass reaction is often used also as an indicator of the role of any other reaction in the model. For instance, the impact of a gene knockout, which may result in a reaction turned off, is measured through the change of flux rate of the biomass reaction.

Unless otherwise stated, we will consider the biomass as the first objective in every analysis performed on genome-scale models. However, biomass is not the only objective considered in FBA. The maximization of ATP, a coenzyme used as an energy source in the cell, is also common. In metabolic engineering, the interest is towards the overproduction of a "synthetic" objective, e.g., acetate or succinate exchange flux, or even a nonnative metabolite such as 1,4-butanediol, with nonnative pathways added to the chemical reaction network [177].

1.2.3 Limitations of FBA

FBA has its shortcomings. First, an ideal model should include different biological processes at different scales (e.g., transcription, translation, signaling, and metabolism). Therefore, modeling the metabolism on its own using FBA ignores the remaining processes and their regulation of cellular function. Furthermore, kinetic ODE models are more suitable than FBA when the focus is on modeling regulatory mechanisms or metabolite concentrations. However, techniques based on ODEs are suitable only for small subsystems where experimental data is available, and do not scale well for large or genome-scale system due to the high computational cost and to the large number of parameters that need to be estimated. There is also little availability of kinetic data for the simulation of chemical reaction net-

works. These issues significantly limit the ability to study genome-scale systems using an ODE-based approach.

A further challenge in metabolic modeling is the combination of extracellular dynamics and intracellular steady state. FBA is a steady-state analysis and therefore does not natively support dynamics. To approximate metabolic behavior under dynamic conditions, a step-wise FBA, called "Dynamic FBA" was introduced (see Section 1.3.1). As the focus of the present work is to analyze large genome-scale metabolic models with a systems-based approach, throughout this dissertation the FBA steady-state approach will be used.

1.3 Applications and challenges of flux balance analysis

Recently, more than 1000 prokaryotic genomes have been fully sequenced, thus allowing FBA models to incorporate also information on enzymes and genome, including the relationships among genes, proteins and reactions (*GPR mapping*). To date, more than 90 genome-wide metabolic reconstructions have been published [39]. FBA-based approaches have proved more efficient than other mathematical modeling techniques, such as those based on ordinary differential equations, at tackling genome-scale metabolic networks [47]. As a result, they have been extensively used to characterize energy production in cells [8], and to design synthetic pathways *in silico* (e.g., for the production of biofuels [177]). The main toolbox for performing FBA in MATLAB is the COBRA toolbox by Palsson's group [143].

1.3.1 Dynamic FBA

In order to combine extracellular dynamics and intracellular steady states, and therefore model metabolism under dynamic conditions, a step-wise FBA approach, commonly referred to as Dynamic FBA, has been developed [104]. The idea is that a steady-state FBA simulation is performed at each time step, and the uptake rates of given nutrients for the next simulation are reduced according to how much nutrient has been consumed in the previous FBA simulation.

More specifically, the assumption is that the bacterium has a limited availability of extracellular nutrients (e.g., glucose), encoded as a constraint (maximum uptake rate) in the associated exchange reaction. Then, after an FBA simulation is run, the amount of nutrient which has been actually taken up by the organism is used to reduce the nutrient availability (maximum uptake rate) for the FBA simulation at the next time step.

Dynamic FBA allows investigating genome-scale networks under transient conditions. It can be thought of as a compromise between fully dynamic models, which cannot be simulated at large scale, and steady-state models, which do not involve kinetics.

1.3.2 Regulatory FBA

Regulatory FBA, proposed by Covert et al. [46], is a framework where the regulatory mechanisms affecting genes are used in conjunction with FBA. Since genes control reactions in the genome-scale model, the presence of an enzyme or regulatory protein may cause given genes to be inactive. Therefore, the corresponding reaction is temporarily removed from the model (e.g., by constraining its lower and upper bounds to be 0). Similarly to Dynamic FBA, the regulatory constraints and the uptake flux bounds are updated in a step-wise manner to allow dynamic modelling of growth. While this approach provides an effective way of modelling regulatory or gene expression-based constraints on FBA models, only Boolean regulation of reactions (on/off) is allowed.

1.3.3 MONGOOSE: solving FBA using exact rational arithmetic

As mentioned above, COBRA [143] is arguably the most widely used toolbox for FBA. Chindelevitch and co-workers recently developed MONGOOSE [39], a pipeline to perform structural analysis and reduction of a genome-scale metabolic network using exact arithmetic (rational instead of floating point). The main claim of MONGOOSE is that, when using exact arithmetic, the biomass reaction is blocked (i.e., not able to carry any flux under any condition) in 44 out of the 89 models examined. Remarkably, they claim that the biomass of three out of ten curated models in the BiGG database from Palsson's group [142] were reported as blocked by MONGOOSE but not by COBRA [1].

Since MONGOOSE was released, an interesting debate has grown in the metabolic modeling community. This was mainly due to the strong claim that those three models curated by Palsson's group (Escherichia coli iAF1260, Helicobacter pylori iIT341 and Mycobacterium tuberculosis iNJ661), were able to carry a nonzero biomass using COBRA's floating point operations, but were blocked to zero biomass if using exact arithmetic in MONGOOSE. In the original MONGOOSE paper, the authors claimed that this difference was due to "COBRA not enforcing the flux balance constraints on internal metabolites exactly" (i.e., using rational arithmetic). The proposed explanation was that, even if the deviations from exact flux balance are small, they can make the difference between a feasible and an unfeasible problem.

[1] The MONGOOSE name is indeed referred to the fact that the mongooses are able to fight and kill cobras.

However, after a commentary by Palsson's group [40], Chindelevith at al. revisited their claim in a recent commentary [114]. In fact, the claim on the three BiGG models being blocked was incorrect, and was due to an incorrect parsing (for two of them) and to a MONGOOSE bug (for the third model). The blockage of five additional models was also due to different parsing between COBRA and MONGOOSE. Although the MONGOOSE bug is reported as a "minor bug", it affects the biomass coefficients and occasionally other coefficients; as a result, four additional models previously classified as blocked were indeed working correctly. Therefore, the original MONGOOSE claims and the initial claim of strong discrepancies between MONGOOSE and COBRA have been significantly weakened.

1.3.4 Incorporating transcriptomic data in FBA models

The integration of various levels of biological organization (omics) in FBA models enables the generation of better computational models with improved predictive capabilities. For instance, the transcriptomic level consists of gene expression levels quantifying the amount of mRNA concentration found in a cell after transcription of genes from the DNA. The mRNA is then translated to proteins, which directly affect the rate at which a chemical reaction takes place in the metabolism. The idea underlying gene expression-based approaches in FBA is therefore that the amount of mRNA, which indicates the expression of a given gene, can serve as an indicator of the protein abundance, and consequently of the activity of the corresponding reactions in the metabolism.

Methods for integrating transcriptomic data in genome-scale metabolic models have been recently reviewed and evaluated [88, 103]. They can be divided into two subsets: *discrete methods* and *continuous methods*. Discrete methods perform binarization (or discretization if more than two levels are used) of gene expression levels in order to set the active and inactive states of the corresponding reactions. The most common approach consists of turning off a reaction if the corresponding gene expression level is below a given threshold. The first on/off techniques were proposed by Covert et al. [46] and Åkesson et al. [2]; similar techniques were then proposed: GIMME [20], iMAT [184] (which uses tri-valued "low/mid/high" expression states), PROM [36], MADE [81], and EXAMO [138]. The main drawback of these methods is the introduction of a threshold that needs to be determined. Additionally, small changes in gene expression level are not captured (unless they cross the threshold level) and therefore different gene expression profiles may yield identical metabolic outcomes.

Continuous methods are instead based on a real-valued map linking gene expression levels and flux bounds of the corresponding reactions. The map is difficult to determine, as the problem of finding a genotype-to-phenotype map is still unsolved in biology. However,

this approach ensures full control on the constraints of the linear program, and even slight modifications in the gene expression level are captured and mapped to the metabolic model, yielding a different metabolic state. Methods belonging to this category are: E-Flux [42] (reactions controlled by overexpressed genes have weaker constraints for their flux rates), Lee et al. [99] (transcriptomic data incorporated in the objective function), and FALCON [18].

Whether gene expression only is a good estimate for protein abundance is still a matter of debate. Therefore, relying solely on transcriptomic data to estimate flux rates in a metabolic network has its disadvantages. For instance, degradation rates, post-translational modifications, protein regulation and degradation are all involved in the biological steps between gene activity and reaction activity, but they are not taken into account by these approaches.

In Chapter 3, FBA is extended to account for environmental conditions by incorporating gene expression levels (transcriptomic) into the model with a continuous approach. Once gene expression has been mapped to the metabolic network, gene expression levels become variables of an optimization algorithm, and multi-objective optimization is performed. Towards addressing the lack of intermediate steps modelled between gene expression and protein abundance, we extend our method with codon usage information, which modifies the translation of mRNA into proteins. Applying multi-objective optimization to the improved model allows us to estimate the gene expression profiles that are likely to maximize the growth rate of the cell while ensuring an optimal amount of an additional flux rate.

In this dissertation, we link the gene expression profiles with the FBA fluxes of the associated reactions in *E. coli* defining a real-valued adjustable map. Protein synthesis is an outcome of the expression of genes coding for protein segments. Hence, we link the gene expression values to the flux of the reactions controlled by the proteins coded by those genes. The approach adopted here, namely using the gene expression level as a proxy for protein abundance, is supported by recent evidences on the correlation between mRNA and protein levels [49, 71, 108]. More recently, Li et al. [100] and Jovanovic et al. [82] showed that mRNA levels are the main contributors to the overall protein expression level in mammals. We note that the availability of proteomic data, compared to gene expression data only, would give direct information on the reaction rates, and would allow a more precise definition of the constraints in the FBA model. However, while being more informative in light of a metabolic model, protein concentration can be currently measured with a lower precision and higher cost if compared to mRNA measurements.

1.4 Multi-objective optimization: Pareto fronts in genome-scale models

When a bacterial metabolic network needs to perform tasks that are in conflict with each other, a multi-objective optimization algorithm is a useful tool to seek trade-off solutions. The aim of multi-objective optimization in biological models is to optimize the secretion or uptake of a set of target metabolites. The two or more contrasting goals of a multi-objective algorithm are usually referred to as *objective functions*. The *Pareto front*, obtained as a result of the multi-objective optimization pipeline, is defined as the set of points x such that there does not exist any other point dominating x at all tasks (objective functions).

Let $f_1, ..., f_r$ be the r objective functions that the organisms aims at optimizing simultaneously. Optimizing simultaneously means, formally, that the organism would need to maximize the following vector function:

$$\begin{aligned} f(x) &= (f_1(x), f_2(x), ..., f_r(x)), \\ x &\in \text{multidimensional search space}, \\ f(x) &\in \text{multidimensional objective space}, \end{aligned} \tag{1.3}$$

where x is the variable (vector) to be optimized in the search space, and $f(x)$ is the corresponding vector in the objective space. Without loss of generality, we assume that all functions have to be maximized, since minimizing a function f_i is equivalent to maximizing $-f_i$.

The output of a multi-objective routine is a set of points in the objective space. This set constitutes the *Pareto front*. It is achieved through the search for all the *Pareto optimal* vectors x^*, namely all those x^* in the search space for which

$$\nexists\, x \in \text{multidimensional search space} \quad \text{s.t.} \quad [f_i(x) > f_i(x^*), \forall i = 1, ..., r] \tag{1.4}$$

(or $f_i(x) < f_i(x^*), \forall i = 1, ..., r$, for the minimization problem). When the objective functions f_i in the organism are in conflict with each other, the term *optimizing* can be thought of as seeking trade-off solutions [41]. In computational biology, the Pareto front can be defined as the set of all the phenotypes that remain after eliminating all the feasible phenotypes dominated in all the tasks required [149].

1.5 Genetic algorithms to approximate the Pareto front

The Pareto front can be approximated using an evolutionary procedure called genetic algorithm (GA). In this dissertation, we propose a genetic algorithm inspired by NSGA-II [52] that performs the search in the space of (input) decision variables, namely Boolean gene knockouts in Chapter 2, and continuous gene expression levels in Chapter 3. In a multi-objective optimization problem, an evolutionary algorithm ensures low computational cost and is able to generate a set of solutions in a single run.

A GA is a robust technique based on the principles of evolution and natural selection. The main features of genetic algorithms are the crossover and mutation operators, which allow the evolution process of the decision variables (inputs) to reach the global optimum for the objective functions (outputs).

The GA is applied to an initial set of input arrays (also called *individuals*) representing possible solutions to the problem. Initially, all individuals are ranked according to their non-domination level (fitness or rank), which is based on the values of the objective functions computed for each individual. After computing the fitness score for all the individuals of the initial population, we employ a *binary tournament selection*, where two individuals are selected at random, and their fitness is compared. The individual with the best fitness is selected as a parent for the next population.

The algorithm then selects a number of parents (i.e., best individuals) equal to half of the population. Parents are mutated using a *combinatorial mutation* and *cross-over operator* to create an offspring, which joins the selected parents in a new population. Finally, the whole process is repeated. As illustrated in Fig. 1.2, the idea behind the GA is that new and possibly better individuals can be obtained by mutating and combining successful existing individuals, and finally keeping only the best individuals with respect to two or more competing objective functions.

Optimizing all the selected objectives simultaneously has advantages over reducing the multi-objective problem to a single-objective optimization. Summarizing two or more objectives in a single objective (e.g., using a linear combination of the objectives) needs a correct estimation of the weights attributed to each objective (coefficients of the linear combination), and most importantly does not permit to correctly approximate non-convex Pareto fronts.

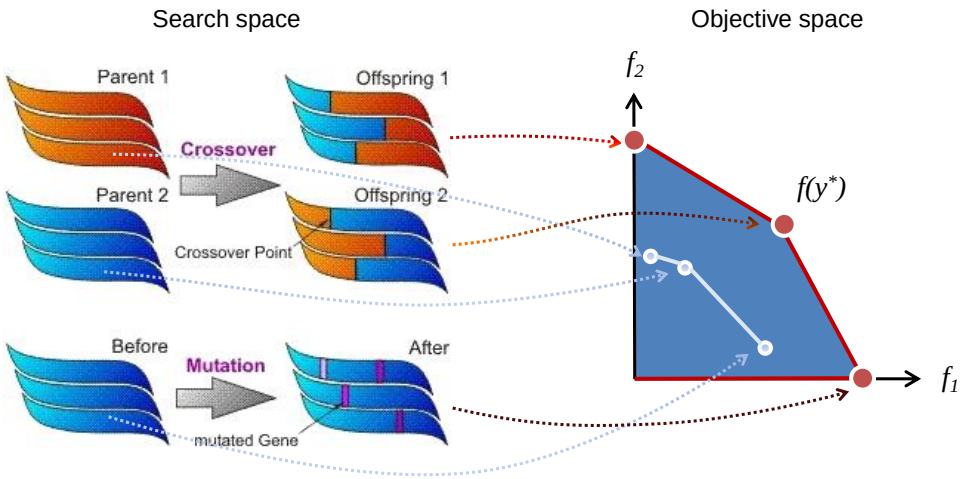


Fig. 1.2 **Genetic algorithms to approximate the Pareto front.** Inspired by the principles of evolution, a genetic algorithm starts from a pool of individuals and evaluates the corresponding output (f_1, f_2) through the model (white front). Only the best individuals are kept in the population and are used to generate new individuals (offspring) through point mutation or cross-over of existing individuals (parents). This process is repeated until no further improvement is detected on the Pareto front (red front in the figure).

1.6 Metabolic engineering

A comprehensive *in silico* analysis is the first step for designing a new synthetic organism. Among the techniques for designing organisms *in silico*, *metabolic engineering* consists of optimizing genetic and regulatory processes within cells to increase the cell production of certain substances.

1.6.1 Metabolic engineering in genome-scale models: the problem of a large search space

Arguably, *Escherichia coli* is the most extensively studied organism in biology, both *in vitro* (synthetic biology) and *in silico* (metabolic engineering). In 2000, Edwards and Palsson [57] published the first genome-scale metabolic network of the K12-MG1655 *E. coli*, composed of 627 reactions, 438 metabolites and 660 genes. Then, the genome-scale reconstruction of *E. coli* was updated to account for 904 genes, 625 metabolites and 931 reactions [133], then 1260 genes, 1039 metabolites and 2077 reactions [60], and finally 1366 genes, 1136 metabolites and 2251 reactions [119].

Metabolic engineering has been applied to *E. coli* and other organisms to find the genetic manipulations, namely gene knockouts, that enable overproduction of a given chemical

("synthetic" objective) in the metabolic network, while keeping a large rate of production of biomass ("natural" objective). We remark that, because of the large number of genes and reactions in the cellular metabolism, the size of the search space is enormous. For instance, in the most recent model of *E. coli* (1366 genes), if one focuses only on the search space of all the possible Boolean gene knockout configurations (as in Chapter 2), where $0/1$ represents gene on/off, there are 2^{1366} knockout configurations to be explored. If we consider the gene expression as a continuous variable (as in Chapter 3), rather than a Boolean variable, the problem becomes even more challenging. This state space explosion renders the problem computationally intractable. The techniques developed in this dissertation, and especially multi-objective optimization, are adopted to effectively search in this large space.

1.6.2 Algorithms for metabolic engineering

Interesting genes predicted through metabolic engineering algorithms are useful targets to guide future experimental studies. The main challenge is that the overproduction of a given chemical in the network usually comes at the expenses of a reduced growth rate. Furthermore, microorganisms have evolved in order to rearrange the flux rates to guarantee maximum production of biomass even after a genetic or environmental perturbation [79]. To reflect this, in any metabolic engineering technique, maximum biomass formation must be ensured after every genetic manipulation [183]. The most common computational methods to perform metabolic engineering are OptKnock, OptFlux, OptGene, and GDLS.

OptKnock [30] uses bilevel linear programming to find and remove reactions whose removal leads to a stronger coupling between biomass and synthetic objective. In large systems, however, this does not necessarily ensure the production of the synthetic chemical that the user needs to overproduce. Specifically, after eliminating the reactions suggested by OptKnock, the model can sometimes avoid the formation of synthetic objective, for particular shapes of the space of feasible objectives [130]. This is due to the presence of a family of alternative solutions that could instead be captured using a multi-objective approach. The main drawback of OptKnock is that it provides only one solution. To find additional solutions, one has to run the algorithm again, by imposing the previous solution as an additional constraint so that the same solution is not found again. This procedure is similar to local search in optimization, and therefore does not allow an effective search in the high-dimensional space of possible gene knockout strategies. Furthermore, it targets reactions directly, and does not implement gene-protein-reaction associations that allow searching knockouts strategies directly on genes.

OptFlux [136] and OptGene [127] implement respectively an evolutionary algorithm and a simulated annealing meta-heuristic, both relying on a local search algorithm. GDLS

[102] starts from an initial Boolean knockout strategy, and then seeks the best strategies that differ from the starting strategy by at most K additional genetic manipulations, where each manipulation consists of changing the state of a single gene (on or off). These strategies are then evaluated using bilevel linear programming, and are used as a starting point for the next round of K additional genetic manipulations. The procedure is then iterated until the round of manipulations does not improve the solutions of the bilevel program. We remark that a limitation of GDLS is not involving multi-objective optimization. In fact, the search is only local and sequential, and each step depends only on the previous one. Furthermore, in order to use GDLS, a model reduction is required.

GDMO, presented in Chapter 2, improves considerably on these aspects by allowing evolution of the proposed Boolean gene knockout strategies. Furthermore, GDMO is a multi-objective optimization evolutionary algorithm that controls a bilevel problem. More specifically, the two objectives of the bilevel linear program become the two objective functions of a multi-objective optimization algorithm, which does not weight them. As a result, both objectives are maximized simultaneously and a Pareto front is produced. Conversely, GDLS only uses bilevel optimization, and therefore intrinsically gives preference to the first objective (biomass), which is the first level of the bilevel problem. Then, it compares the proposed solutions by taking into account only the amount of second objective (acetate or succinate exchange flux). As an optimization algorithm, GDLS can be though ot as a single-objective version of GDMO, where the search is only local and the manipulations on the proposed knockout strategies are only performed by an incremental mutation operator.

By directly comparing the solutions proposed by GDMO and GDLS on the same model and with the same flux rates chosen as objectives, we will show that the solutions provided by GDLS never outperform the Pareto fronts found by GDMO, since they occupy positions in the area under the Pareto curves. In the best case, the GDLS solutions lie on the Pareto front, which however contains additional solutions not found by GDLS.

A common drawback of these methods is that, although genome-scale models contain reactions with information about the genes involved in their catalysis, they only capture qualitative rules (e.g., a given reaction is turned off because it requires either gene A or gene B, which are both off). These methods are therefore not sufficient to study the quantitative effects of gene expression on the metabolism. METRADE, presented in Chapter 3, will address this issue. Both GDMO and METRADE belong to the broad class of metabolic engineering algorithms, as their goal is to maximize the growth rate and the production of a chemical.

1.7 Codon usage affects translation of genes into proteins

Natural selection acts as a driving force at virtually all levels of the genetic information processing and biological organization: from DNA stability, replication and transcription to messenger RNA, life span and efficiency of translation into proteins, to the correct functioning of the metabolic network in the building up and propagation of a living organism. Although, in principle, all these constraints could interact in a very complex way, it is indeed fruitful to try to untangle the role of each element.

Codon usage bias is an important feature to take into account during the process of translating genes into proteins. A *codon* is a triplet of DNA nucleotides that codes for a specific amino acid. The process of translation of the genome allows expressing genes into cellular functions. The translation of coding sequences into proteins starts when the ribosome, a cellular structure where proteins are synthesized, is positioned on the AUG codon (except for some genes using alternative start codons), followed by the polypeptide synthesis in the ribosomal tunnel. The rate of protein synthesis depends on many factors, e.g., the rate of transfer RNA (tRNA) binding and the kinetics of the process. Each tRNA provides the code to assign a codon to a specific amino acid. A tRNA exposes an amino acid and a nucleotide triplet (*anticodon*) that recognizes a specific codon. Specifically, there are 20 amino acids and 4^3 codons, 61 of which actually encode for amino acids. An amino acid can be encoded by one or more codons, while each codon encodes always for the same amino acid [156]. In particular, an amino acid of a growing polypeptide chain can be encoded by up to six codons. Codons coding for the same amino acid are referred to as *synonymous codons*. The different tRNA species exposing the same amino acid are differentially expressed: some tRNAs are more abundant than their synonymous cognates. As a consequence, synonymous codons are not equivalent and are not used randomly.

The codon bias is strong in highly expressed genes, indicating that codon composition has an impact on translation efficiency [121]. Specifically, high-frequency-usage codons allow the quick generation of the polypeptide chain, while low-frequency-usage codons slow down the translation process and allow the nascent protein to fold into a helical structure [76, 171]. In this regard, it has been experimentally proved that replacing rare codons with frequent synonymous codons improves the rate of translation [73]. The usage of each codon reflects the amount of its corresponding tRNA. Differences in codon usage frequency are therefore responsible for rapid or slow translation of genes into proteins, thus affecting the gene expression process [33]. The codon usage in bacteria can modulate the translation to reach the maximum rate of 15 amino acid per seconds on average. A good estimate of the effect of codon usage on the translation process, with realistic assumptions on the *E. coli* behavior, has been produced in [53].

While FBA and flux optimization capture the behavior of an organism at steady state, controlling and optimizing codon usage may also allow capturing the phenotypic noise [141], therefore permitting the adaptation of the organism to a variety of environments. An optimal codon usage also enables fast translation without misincorporations.

In Section 3.4, we will assess the genome-wide transcriptional and translational organization by analyzing the multi-objective optimization of the codon usage distribution in genes, and how this affects the fluxes in the same pathway and the overall metabolism. By modeling the codon bias in the FBA framework, in Chapter 3 we will establish estimators (Pareto optimality and associated measures) of the transcriptional and translational fitness bottlenecks in metabolic pathways. This represents a guide for practical solutions of synthetic biology for gene design in natural strains.

1.8 Why adopting a multi-objective approach?

The question *'what does a particular cell do?'* has often more than one correct answer. A common assumption in systems biology is that microorganisms tend to optimize their metabolic network in order to maximize the growth rate (biomass). However, whether the biomass is the right objective for the analysis of the metabolism is still a matter of debate [22]. There is increasing evidence that bacteria have to cope with multiple, sometimes competing, objectives to optimize simultaneously [145], and a single objective function is not able to capture these. It is also likely that metabolism is not fully optimized for any particular objective [126], and evolution has shaped cells in order to reach an optimal trade-off between all objectives [179].

This suggests that the commonly used single-objective approach, e.g., the maximization of the growth rate, may not be appropriate in many systems biology applications. Therefore, throughout this dissertation, we always take into account multiple objectives. An optimization process that takes into account multiple objectives has also the advantage of ensuring metabolic flexibility for possible reorganizations performed during adaptations to changes in the environmental conditions. Finally, exploring a set of trade-off solutions between competing objectives, rather than a single biomass-maximizing solution, also accounts for suboptimal solutions [174].

Importantly, as detailed in Section 1.4, our multi-objective approach does not require the combination of the objectives into a single objective function. In fact, no prior knowledge is generally available on how two or more particular objectives are balanced in a given cell. Therefore, rather than establishing weights for each objective and then combining them into

a single objective, we optimize all the objectives simultaneously with a GA, providing the final trade-off curve.

1.9 Structure of the dissertation

The remainder of the dissertation is organized as follows. In Chapter 2, we present a multi-objective framework for analysis and optimization of genome-scale metabolic networks. By using a multi-objective optimization method called Genetic Design by Multi-objective Optimization (GDMO), which we published in [44], we maximize several pairs of biological functions, such as acetate production and biomass formation.

In the optimization procedure, we search for the best genetic strategies that maximize selected objectives. The results are represented as a *Pareto front*. The Pareto front is a representation of the ability of the organism to respond to environmental changes by exhibiting plasticity and ability to execute various cellular tasks. The area under the front, the extension and the points of the front, the *knees* and the *jumps* are features that summarize the characteristic phenotype of the organism, and are useful tools to cross-compare models or different organisms. GDMO is then augmented with sensitivity and robustness analyses, which provide a comprehensive assessment of the model under investigation and a post-processing evaluation of the solution found by the optimization process.

To measure adaptability to changing environmental conditions and over time, in Chapter 3 we develop a multi-omic model of *Escherichia coli* that accounts for metabolism, gene expression and codon usage at both transcription and translation levels. Bacterial phenotypic traits and lifestyles in response to diverse environmental conditions depend on changes in the internal molecular environment. However, predicting bacterial adaptability is still difficult outside of laboratory-controlled conditions. Many molecular levels can contribute to the adaptation to a changing environment, e.g., pathway structure, codon usage, metabolism. After the integration of multiple omics into the model, we propose a multi-objective optimization algorithm to find the allowable and optimal metabolic phenotypes through concurrent maximization or minimization of multiple metabolic markers. In the condition space, we propose Pareto hypervolume as a proxy for short-term multi-omic (transcriptomic and metabolic) evolution, thus enabling comparative analysis of metabolic conditions. We are therefore able to compare, evaluate and cluster different experimental conditions, models and bacterial strains according to their metabolic response in a multidimensional objective space, rather than in the original space of gene expression data (microarrays). Our approach proves useful to integrate and assess multiple omic layers in a constraint-based model, and is experimentally validated on a publicly available phenomics dataset of *E. coli*, and on a

set of genome-scale reconstructions grown in a compendium of environmental conditions. Our method, named METRADE, is freely available as a MATLAB toolbox.

In Chapter 4, we focus on the role that computation plays in the bio-inspired science. This research field was pioneered by Turing in 1952 [162], who proposed computational processes in morphogenesis. We propose a mapping between a living organism and the von Neumann architecture, where the metabolism executes reactions mapped to instructions of a Turing machine. A solution found by GDMO represents the optimal genetic knockout strategy and can play the role of an executable program stored in the "memory" of the organism. We adopt our framework to investigate scenarios of communication among bacteria, gene duplication, and lateral gene transfer events. Finally, we use this mapping to estimate the computational effort for a specific metabolic task, and the computational capability of the organism as function of communication outcomes, e.g., gene duplication events.

1.9.1 List of genome-scale metabolic models used

In each chapter, while the methodology is covered extensively, only the main results are reported. The reader is referred to the related publications, listed at the end of each chapter, for more details and for the full set of applications of the method. In Table 1.1, we list the genome-scale models used in this dissertation.

Model	Reactions	Metabolites	Genes	Ref.
Escherichia coli iAF1260	2077	1039	1260	[60]
Escherichia coli iJO1366	2251	1136	1366	[119]
Human heart mitochondrion	253	245	n.a.	[151]
Chlamydomonas reinhardtii (algal cell)	2190	1068	1080	[37]
Saccharomyces cerevisiae (yeast)	3493	2218	916	[14]
Corynebacterium glutamicum	518	399	468	[165]

Table 1.1 **List of models, number of reactions, metabolites and genes used to test the methodology proposed in this dissertation.** We have excluded exchange reactions (i.e., those modeling the flow of metabolites, e.g., glucose and oxygen, in and out of the cell) from the count. External exchange metabolites, representing one end of the exchange reactions, are also excluded. The model of human heart mitochondrion does not contain gene-to-protein association rules. In the last column we report the publication from which the original model was initially extracted.

Chapter 2

GDMO: a comprehensive framework for multi-objective Boolean optimization of metabolic networks

In this chapter we propose a comprehensive optimization method, called Genetic Design through Multi-objective Optimization (GDMO), that works at the genetic level of a metabolic model and finds the genetic strategies to be followed in order to optimize simultaneously multiple targets of interest. By exploring effectively the whole space of gene knockouts, GDMO optimizes biomass, acetate and succinate production, as well as other multiple biological functions, using FBA [120] and multi-objective optimization. We then compare GDMO with state-of-the-art methods for genetic design, namely GDLS, OptFlux, OptGene, OptKnock. Each point of the Pareto front proposed by GDMO represents a different strain or the same strain adapting differently to various sets of environmental conditions. Pareto optimality allows obtaining not only a wide range of Pareto optimal solutions, but also the best trade-off design. GDMO has been developed in collaboration with Dr J. Costanza, Dr G. Carapezza, and Prof G. Nicosia, all from the University of Catania, Italy.

With the aim of investigating metabolic networks in an automated way, we included GDMO in a framework consisting of a set of computational techniques: optimization, sensitivity analysis, robustness analysis, ε-dominance analysis, identifiability analysis (Fig. 2.1) [10]. The search for optimal configurations of the network is performed through combinatorial operators in the space of gene knockouts, where each gene is modeled by a Boolean on/off variable.

We seek the Pareto-optimal knockout strategies to simultaneously optimize two or more objective functions. The sensitivity analysis quantifies the importance of the input

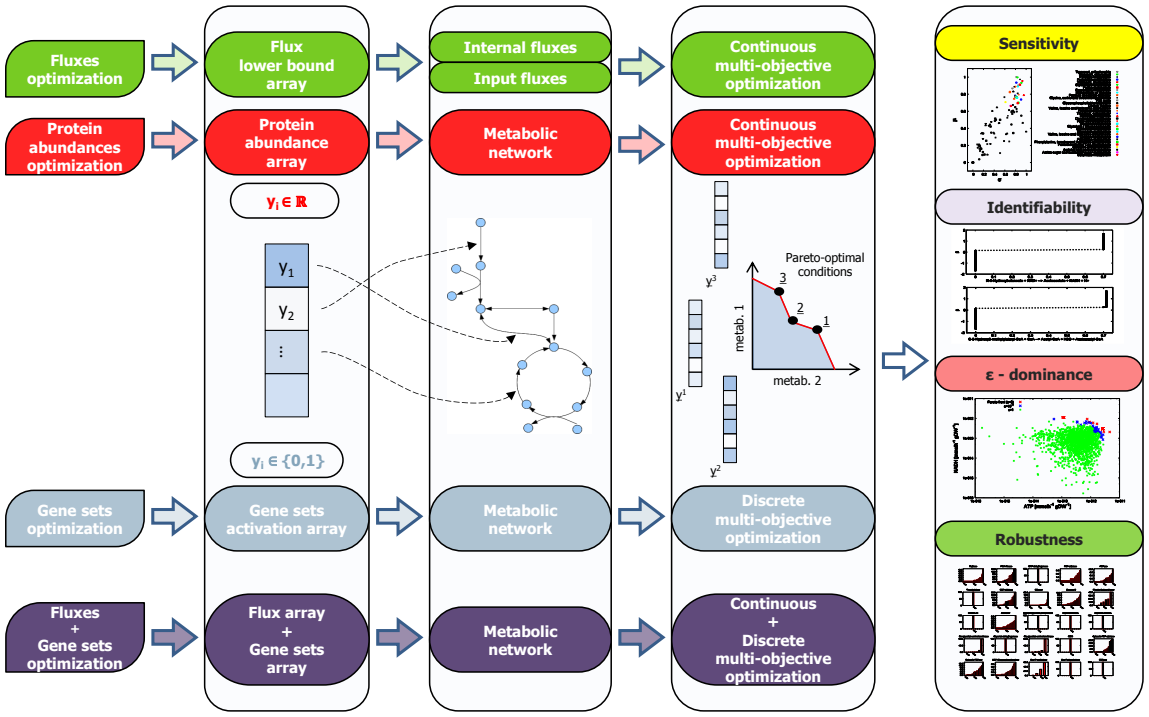


Fig. 2.1 **A framework for optimal design of metabolic networks.** The optimization framework is able to perform multi-objective optimization on genes and gene sets (sets of genes coding for a protein each). The optimization is augmented with sensitivity, identifiability, robustness and ε-dominance analysis.

variables in the model. The identifiability analysis infers functional relations between them. The robustness is used in combination with the sensitivity and quantifies if a solution is reachable even if small perturbations are applied to the system. The ε-dominance analysis identifies sub-optimal points. Note that only sensitivity and robustness are presented in this dissertation, while ε-dominance and identifiability analysis are covered in [10]. We apply our methods to a set of genome-scale FBA models.

2.1 Genetic design through multi-objective optimization (GDMO)

Genetic Design through Multi-objective Optimization (GDMO), which we published in [44], is a genetic algorithm inspired by the Non-dominated Sort Genetic Algorithm 2 (NSGA-II) [52], a combinatorial GA belonging to the class of evolutionary algorithms. GDMO is able to optimize two or more objective functions, and to find a set of non-dominated (i.e., Pareto optimal) solutions consisting of gene knockout strategies. The idea is that by finding the

right combination of genes to turn off, one can improve the bacterial secretion of a desired metabolite while keeping a high growth rate. Multi-objective optimization does not provide a single optimal knockout solution (as it occurs in single-objective optimization problems), but a set of finite optimal solutions that constitute the Pareto front.

GDMO, presented in Algorithm 1, seeks genetic knockout strategies able to simultaneously optimize multiple biological functions, e.g., maximization of biomass, maximization of acetate and minimization of the knockout cost (number of genes to turn off). The idea is that the multi-objective optimization algorithm proposes gene knockout strategies that are subsequently mapped to the FBA model. FBA is then applied in a bilevel fashion, giving preference to the biomass maximization over the maximization of the second objective. The values of these two objectives (obtained starting from the gene knockout strategies) are finally assigned to the objective functions of the optimization algorithm. Using the non-domination principle described in Section 1.4, only the best strategies are kept, and new strategies are generated using evolutionary operators. The process is iterated to create the Pareto front. The advantage of using multi-objective optimization is that it does not weight objective functions. However, in GDMO, the optimization algorithm is driving a bilevel FBA model, which gives preference to one objective over the other. Therefore, the order in which the objectives are considered in the bilevel FBA model might affect the final Pareto front. As a result, prior to generalizing our approach, one should assess if the system is sensitive to the order in which the two (or multiple) objectives are considered.

Searching for optimal genetic strategies has been modeled as a combinatorial problem, since the genetic code can be represented by a binary vector y. Each element y_l of y is a gene set, i.e., a set of genes coding for enzymes, isozymes, or enzymatic complexes that catalyze one or more reactions. Seeking genetic strategies means searching for the best binary knockout vector, where the element y_l is 1 if the lth gene set is turned off. In our method, we consider a maximum number of knockouts, i.e., a maximum number C of elements in y equal to 1. y is the vector of the decision variables to be optimized in the binary search space. GDMO starts with the initialization of a population P. The population is formed by individuals, where each individual is represented by a Boolean vector y representing a knockout strategy. During the initialization, each individual can represent the wild-type combination (all zeros), or can be obtained by random mutations.

When P is initialized, the fitness of all individuals is calculated. The fitness is defined as the measure of rank and crowding distance. Rank values are assigned to each individual based on whether or not they belong to the Pareto front. The crowding distance is a measure of how much close an individual is to its neighbors. Large average crowding distance will result in better diversity in the population, which is indeed one of the goals on GDMO.

The next steps are repeated during the generations (or iterations) of the algorithm until the number of generations or convergence is reached. In a selection step, 50% of best individuals of P are chosen to generate the new offspring. Therefore, in a combinatorial mutation operator (Algorithm 2) new individuals are created. This offspring population, called p in Algorithm 1, is merged with the entire parent population P (this procedure, called *elitism*, ensures that the best parents are kept in the following populations) and, using a tournament selection, only the best I individuals are used to form the new population P. In a tournament of two individuals (binary tournament), an individual is selected if its rank is lower and its crowding distance is larger. The binary tournament selection ensures that the best individuals in the population are selected based on their rank and crowding distance. The new population P is used in the next iteration, and the process is repeated until a stop condition is met or no further improvement is seen. At the last iteration, the individuals contained in P constitute the optimal set of knockout strategies.

Our approach solves one of the most significant bottlenecks of FBA models, namely the lack of control and optimization on genes. In this algorithm, an individual can be thought of as a genetic variant or subtype of the organism (also called *strain*), characterized by a specific gene expression profile. The populations of strains evolve towards an optimal population, i.e., a population made up of the best strains for optimizing the objective functions. This process mimics the feast and famine phases of the bacterial populations, in which only the fittest phenotypes are kept.

The results obtained with GDMO are presented in Table 2.1, where its performance is compared against alternative approaches for genetic design. To compare these methods, we run OptKnock in its GAMS implementation [30], while the GDLS, OptGene and OptFlux solutions are extracted by published data [102]. The *E. coli* network is initialized with an empty set of knockouts ("wild-type configuration"). We let GDMO evolve 1500 populations of 1000 individuals each, where each individual represents a genome-wide Boolean knockout strategy. We report the best solutions obtained by each method in terms of biomass formation, acetate and succinate production. In designing knockout intervention strategies, overproducing acetate and succinate is key for biotechnological or medical purposes.

Acetate is an important target for biotechnology, with multiple industrial applications [131]; it is central to many pathways in both aerobic and anaerobic *E. coli* metabolism. Being an intermediate metabolite, it is representative of processes not directly related to growth, and therefore it is highly indicative of metabolic flexibility for possible reorganizations that need to be performed during adaptations to environmental changes. When acetate

Algorithm 1 GDMO pseudo-code

Require: $[f, y, I, MAXgen, C, K]$
/* f output of the FBA model (two flux rates, e.g., biomass and acetate)*/
/* y knockout vector of length L */
/* L number of gene sets */
/* I number of individuals surviving in each population */
/* $MAXgen$ maximum number of populations */
/* C maximum number of knockouts allowed*/
/* K number of new children generated by each best individual */

$P_1 \leftarrow$ initial population with random binary vectors $\tilde{y}$ or null wild-type vectors y
Evaluate the rank (fitness) f of all individuals
for $i \leftarrow 1$ to $MAXgen$ **do**
/* Define p_i selecting the best $I/2$ individuals of population P_i according to the NSGA-II [52] tournament selection (lower rank and higher crowding distance). The set of individuals p_i will then be used to generate a mutated set $\tilde{p}_i$, which will be finally merged with the current population P_i. The new merged population Q_i will undergo a tournament to select the individuals that will constitute the new population P_{i+1} for the next iteration of the algorithm */
    $p_i \leftarrow$ Tournament Selection(P_i)
    $p_i \leftarrow$ best $I/2$ individuals from p_i
    for $h \leftarrow 1$ to $I/2$ **do**
        /* for each individual $p_{i,h}$ of population p_i we perform the Mutation operator (Algorithm 2) sequentially and obtain K mutated children, but only the best child will be selected to become an individual of the mutated set $\tilde{p}_i$*/
        $\tilde{y}_{i,h,1} \leftarrow$ Mutation($p_{i,h}, C/L$)
        for $j \leftarrow 2$ to K **do**
            /* perform Mutation for the most recent child $\tilde{y}_{i,h,j-1}$ and create a new child $\tilde{y}_{i,h,j}$*/
            $\tilde{y}_{i,h,j} \leftarrow$ Mutation($\tilde{y}_{i,h,j-1}, C/L$)
        end for
        Evaluate the rank (fitness) of the K children $\tilde{y}_{i,h,j},\ j = 1,\ldots,K$
        Select the best child $\tilde{y}_{i,h,j^*}$ from $\tilde{y}_{i,h,1},\ldots,\tilde{y}_{i,h,K}$
        $\tilde{p}_{i,h} \leftarrow \tilde{y}_{i,h,j^*}$ /* $\tilde{p}_{i,h}$ is the best mutation of $p_{i,h}$, $h = 1,\ldots,I/2$ */
    end for
    $Q_i \leftarrow$ Merge P_i with $\tilde{p}_i$ /* Q_i contains $I + I/2 = 3I/2$ individuals now */
    /* Perform Tournament Selection on Q_i and obtain the new P_{i+1} */
    $P_{i+1} \leftarrow$ Tournament Selection(Q_i)
    $P_{i+1} \leftarrow$ best I individuals from P_{i+1} /* The final population always contains I individuals */

    if (stop condition met) OR (no improvement in P_{i+1} compared to P_i) **then**
        return P_{i+1}
    end if
end for
return P_{i+1}

Algorithm 2 Mutation pseudo-code

Require: $[x, \alpha]$

/* $\alpha \in [0,1]$ is a real constant and defines the maximum amount of mutations that can be undergone by the Boolean vector x of L elements*/

/* Select randomly an integer value b in $[0, \lfloor \alpha L \rfloor]$ */

1: $b \leftarrow$ random (αL)

/* Select randomly b bits in x */

2: $ind \leftarrow$ random (L, b)

/* Flip bits selected */

3: $x[ind] \leftarrow \text{not}(x[ind])$

4: **return** x

is present at high levels, it inhibits cell growth and recombinant protein productivity [54]. In our simulations, we require at least 0.05 h-1 of biomass.

As shown in Table 2.1, with respect to the wild-type bacterium, GDMO and the other methods propose strategies that guarantee a high production of acetate or succinate, at the expense of the growth rate. GDMO finds the knockout strategies that yield the largest value of acetate (19.150 mmol h^{-1} gDW^{-1}) and succinate (10.610 mmol h^{-1} gDW^{-1}). Both solutions by GDLS are outperformed by the second and third solution (respectively) by GDMO. The first OptKnock solution is outperformed by the first GDMO solution. The second OptKnock solution is outperformed by the third GDMO solution only in terms of succinate, but not in terms of biomass. However the solution proposed by OptKnock is penalized by a large *knockout cost* (defined as the number genes to be turned off to implement the strategy). GDMO overcomes the other methods also proposing strategies with a lower knockout cost. We remark that a low knockout cost is a desirable feature when implementing these strategies *in vitro*. Many solutions proposed by the methods evaluated in this comparison have a knockout cost that would render the solution very hard to reproduce in laboratory (for instance, OptKnock finds solutions that would involve turning off 53 or 54 genes).

When we consider the problem of maximizing acetate/succinate production and biomass formation by searching for genetic manipulations in an FBA context, we take the on/off status of reactions as our decision variables, and we solve a *conflict problem* of maximization because the increase of one objective function (e.g., acetate) implies the decrease of the other (biomass). In this case, the slope of the Pareto front reflects a progressive lack of pathways to sustain the production of one component when we are optimizing the metabolism to maximize the other. "Jumps" (quick decreases or missing parts) mark the sudden loss

	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.	12.565 (+51.4%)	n.a. n.a.	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%)	n.a. (+12362%)	9.069 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%)	0.1165 (-77.9%)	0.1181 (-49.6%)	0.130 (-43.72%)	0.053 (-77.10%)	0.087 (-62%)
kc	n.a.	n.a.	n.a.	14	26	53	54	3	10	8
GR (%)	54.76/53.68	n.a.	n.a.	13.76	16.6	43.08	43.24	45.32	27.6	40.40
LR (%)	54.0/54.67	n.a.	n.a.	16.0	21.33	40.60	40.00	39.33	24.0	46.0
PoRA (%)	100.0/99.33	n.a.	n.a.	19.33	28.67	76.67	87.33	81.33	43.33	83.33

Table 2.1 **Comparison between GDMO and existing genetic design methods.** We compare OptFlux ([136]), OptGene ([127]), GDLS ([102]), OptKnock ([30]) and our multi-objective optimization algorithm (GDMO) to maximize the objective functions of biomass [h^{-1}], acetate and succinate production [mmol h^{-1} gDW^{-1}] (millimoles per gram of dry weight per hour) in the *E. coli* model iAF1260 [60]. In designing knockout intervention strategies, overproducing acetate and succinate is key for biotechnological or medical purposes, while keeping high biomass ensures the growth of the bacterium. For all methods, the *E. coli* network is initialized with an empty set of knockouts ("wild-type configuration"). GDMO evolved 1500 populations of 1000 individuals each, where each individual represents a genome-wide knockout strategy. We show one solution per column, and different columns indicate different objectives chosen, or different solutions obtained with the same pair of objectives ("n.a." stands for *not applicable*, i.e., that metabolite has not been chosen as objective). The third and fourth rows show the biomass and the knockout cost (kc), i.e., the number of genes that must be turned off in order to obtain that solution. The last three rows show a comparison between the robustness analysis methods. GR and LR values are global and local robustness indices (see Section 2.2). The strain is more robust when GR, LR and PoRA are high. In PoRA (pathway-oriented robustness analysis) the perturbation to evaluate robustness is carried out simultaneously for all the fluxes clustered in a metabolic pathway, therefore obtaining a robustness index for each pathway. 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) respectively.

of pathways due to the critical unavailability of an enzymatic step. In other words, they correspond to sudden decreases in the availability of entire pathways and subnetworks when a crucial hub is eliminated (e.g., the Krebs cycle). The region of the Pareto front near a jump suggests that slight changes of conditions, or a handful of genetic mutations, may result in a large change in the amount of product.

2.2 Robustness analysis

The ability of a system to adapt to perturbations due to internal or external agents, aging, temperature, environmental changes and, in our case, also due to molecular noise and mutation is one of the fundamental design principles. To optimize the production of a specific metabolite (and simultaneously the formation of biomass, which is necessary to maintain the survival of the bacterium), we first use GDMO in order to obtain a strain that maximizes the features required by us. At this point, the validity of the biological strain, designed *in-silico*, must be tested as regards the robustness and sensitivity to endogenous and exogenous perturbations. A robustness analysis is able to assess the ability of a strain to adapt to small perturbations that can occur at any stage of the biochemical processes within the bacterium, or are caused by the environment in which it reproduces. By the term "robustness" we mean the ability to maintain an acceptable performance relative to the metabolite production and biomass formation that were previously optimized from a multi-objective standpoint, even after small perturbations that the system might undergo.

There are numerous methods that can be used to fulfil this task. For instance, the theory of percolation on random graphs can be used to test the robustness of the network in case of random or targeted node deletion, or in case of random link deletion [32]. Additionally, the relationship between the general characteristics of a chemical reaction network and the sensitivity of its equilibrium can be investigated by perturbing the overall supply of reagents [148]. Finally, in [74], a combined approach of global and local robustness is proposed, named *glocal robustness*. The global analysis investigates the parameter space with the aim of finding where a circuit cell shows experimentally-observed features, while the local one determines the robustness of parameter sets sampled during the previous phase.

Here we adopt a simple definition of robustness analysis, easily applicable to similar problems in other fields. The basic principle of this analysis is as follows. 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 σ is defined by a random distribution. In order to perform a statistically-meaningful calculation of robustness, we generate a set T of trial

samples τ. Each element τ of the set T is considered robust to the perturbation, due to stochastic noise σ, for a given property (or metric) ϕ, if the following test gives 1 as result:

$$\rho(\Psi, \tau, \phi, \varepsilon) = \begin{cases} 1, & \text{if } |\phi(\Psi) - \phi(\tau)| \leq \varepsilon \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

where Ψ is the reference system, ϕ is a metric (or property), τ is a trial sample of the set T and ε is a robustness threshold. The definition of this condition makes no assumptions about the function ϕ, which is not necessarily related to the properties of the system, but must be quantifiable. The robustness Γ of a system Ψ is the number of robust trials of T, with respect to the property ϕ, over the total number of trials ($|T|$). Formally:

$$\Gamma(\Psi, T, \phi, \varepsilon) = \frac{\sum_{\tau \in T} \rho(\Psi, \tau, \phi, \varepsilon)}{|T|}, \tag{2.2}$$

where Γ is a dimensionless quantity that states, in general, the robustness of a system and, in our case, of an *E. coli* strain taken from the Pareto front.

The robustness index is a function of ε and σ, so the choice of these parameters is crucial and not a trivial task. Since we are interested in the behavior of a strain when subjected to small perturbations, and given that the behavior is acceptable when the deviations from the original value is as small as possible, we generate trial samples τ by choosing ε in the range $0\% - 10\%$ of the metric $\phi(\Psi)$ and σ in the range $0\% - 10\%$ of the perturbed variable in Ψ. Based on this principle, we evaluate two values of robustness: the *Global Robustness* value (GR), and the *Local Robustness* value (LR). These two values only differ in the type of perturbation applied and, specifically, once σ is fixed they differ in the set of variables perturbed.

Global Robustness. As regards the global robustness of a strain, a trial τ is created by perturbing all the upper bound v_j^U and lower bounds v_j^L, $j = 1, \ldots, n$ of the metabolic fluxes. We create a set T of trials, and for each of them we simultaneously perturb all the bounds and evaluate the property $\phi(\tau)$ (through FBA), which in our case can be the value of acetate, succinate, biomass or a combination of them; then, we calculate the function ρ. Once a value of ρ is obtained for each trial, we compute the value of robustness (Equation 2.2), which in this case we call *Global Robustness* because all the parameters are perturbed.

Local Robustness. In this case, we perturb again the upper bound v_j^U and lower bound v_j^L, $j = 1, \ldots, n$, of a metabolic flux. We create a sample trial by perturbing a single flux, we

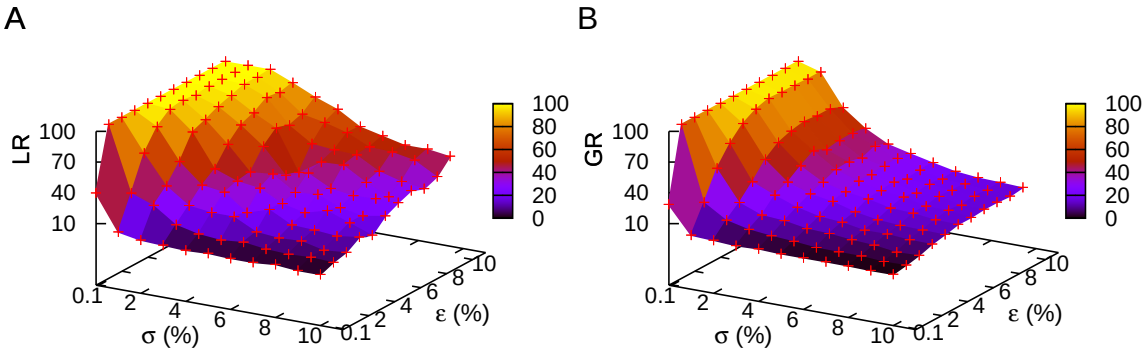


Fig. 2.2 Local (A) and global (B) robustness variation versus σ and ε values in *E. coli* iAF1260. σ is proportional to the perturbed parameters (lower and upper bounds of the fluxes), while ε is proportional to the metric (in this case acetate and biomass productions). For the local value, we report the minimum value found (this value is associated with the glucose uptake rate).

evaluate the property $\phi(\tau)$, and we calculate the function ρ. After creating a set T_τ of trials, we calculate the robustness (Equation 2.2), which in this case we call *Local Robustness*. Hence, the LR value is computed for each metabolic flux, and the value of interest is the minimum LR index found, which is large for robust systems. In Fig. 2.2 we show the robustness values found in the *E. coli* model iAF1260 as a function of ε and σ.

The robustness results associated with the solutions computed by GDMO are presented in Table 2.1. The robustness of the solutions generated by GDMO is larger than the robustness of the solutions generated through GDLS. OptKnock also ensures high robustness. Unsurprisingly, the most robust strain is the wild-type *E. coli*, as it did not undergo any genetic manipulation. Indeed, we expect that a wild-type bacterium is in a stable and robust metabolic state, while an engineered organism is able to overproduce a given metabolite at the expense of robustness.

2.3 Sensitivity analysis

Sensitivity Analysis (SA) is a method for identifying the importance of the components (inputs) in a model. Recently, SA has been applied to evaluate reactions (RoSA - Reactions oriented Sensitivity Analysis) [157] and species (SoSA - Species oriented Sensitivity Analysis) [181]. Here, the idea is to perform a pathway-oriented SA (PoSA), with the aim of detecting the most sensitive pathways in the FBA model of *E. coli*. Unlike standard SA

methods, whose inputs (reactions or species) are real-valued, PoSA is applied with Boolean inputs representing the presence or absence of each reaction in the metabolic network.

2.3.1 Morris' sensitivity analysis

The standard approach to sensitivity analysis was pioneered by Morris [115]. Specifically, the input factor space of a model is discretized and the possible input factor values are assumed to be inside a regular k-dimensional t-level grid, where t is the number of levels of the design. The elementary effect of a given value x_i on a given output G is defined as a finite difference derivative approximation, called *elementary effect*:

$$EE_i(x) = \frac{G(x_1, x_2, ..., x_{i-1}, x_i + \Delta, x_{i+1}, ..., x_k) - G(x)}{\Delta}, \tag{2.3}$$

for any x_i between 0 and $1 - \Delta$, where $x \in \{0, 1/(t-1), 2/(t-1), \ldots, 1\}^k$, and Δ is a predetermined multiple of $1/(t-1)$ that ensures that small perturbations with a big impact on G are assigned a large EE. The influence of x_i is then evaluated by averaging several elementary effects at randomly selected values of x_i on the grid. The idea is to assess the importance of each input x_i with a number of elementary effects proportional to k. If all the samples of the elementary effect of the ith input factor are zero, then x_i does not have any effect on the output, and the sample mean and its standard deviation will both be zero. For more complex interactions, due to interactions between factors and nonlinearity, high mean indicates a factor with an important overall influence on the output; high standard deviation indicates that either the factor is interacting with other factors or the factor has nonlinear effects on the output.

In Fig. 2.3, we show the result of the Morris method considering as input the uptake flux rates in the genome-scale metabolic network of the mitochondrion [151], and as output the array v of flux distribution. As a consequence we remark that, in this case, the numerator in (2.3) is replaced by the Euclidean distance $\|v(x_1, x_2, ..., x_{i-1}, x_i + \Delta, x_{i+1}, ..., x_k) - v(x)\|$. In the plot, we report the mean and the standard deviation of the distribution of elementary effects for each input. We note that the effect of varying the oxygen exchange flux is considerably higher than the effect of varying any other import flux. This is also confirmed by the different metabolic configurations that a bacterium adopts in aerobic versus anaerobic regimes. While the mean of the elementary effects is proportional to their variance for many exchange fluxes, maintaining both μ and σ is crucial in different types of sensitivity analysis, e.g., PoSA presented in the next section.

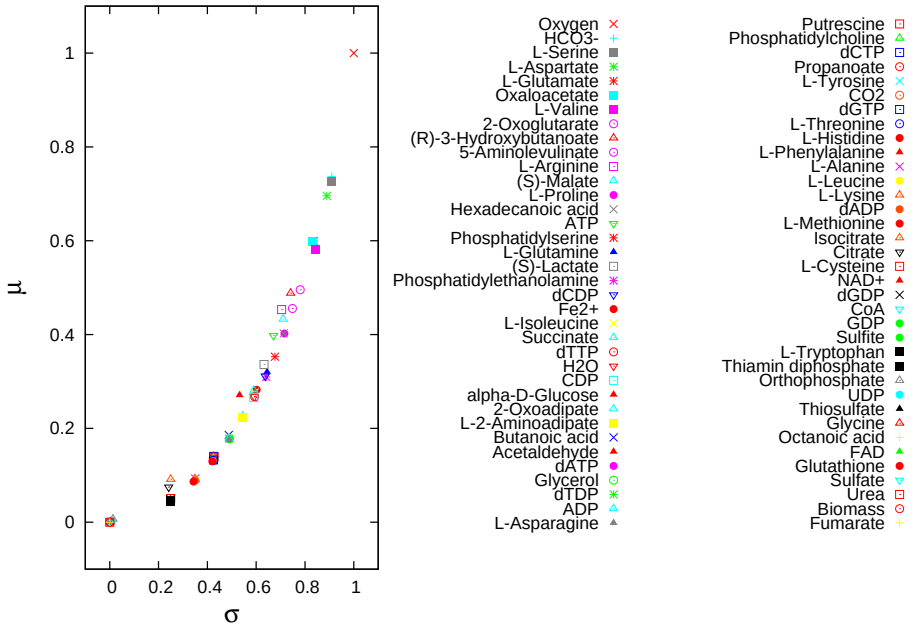


Fig. 2.3 **Sensitivity analysis on the mitochondrial FBA model.** We consider as input the uptake flux rates in the genome-scale metabolic network of the mitochondrion [151], and as output the array of flux distribution. The plot shows the mean μ and the standard deviation σ of the elementary effects computed through the Morris method (2.3) applied on the upper bounds of the exchange reaction fluxes. The legend ranks the inputs from the most to the least important in terms of sensitivity.

2.3.2 Pathway-oriented sensitivity analysis

PoSA (Pathway-oriented sensitivity analysis) investigates the knockout solution space and determines the influence of the pathways on the outputs of an FBA model. Since GDMO provides a set of feasible solutions with different genetic manipulations, investigating the PoSA sensitivity indices starting from the solutions will establish a relation between pathways and proposed manipulations, therefore allowing us to select only certain knockout strategies on the Pareto front. For instance, after the GDMO optimization, a possible use of PoSA is to help in selecting only those GDMO knockout strategies that affect genes belonging to insensitive pathways.

In PoSA, the knockout vector y used to represent the genetic manipulations is partitioned in p subsets of binary variables $\{b_1, b_2, \ldots, b_s, \ldots, b_p\}$, where each b_s includes the genetic manipulations linked to the reactions involved in the sth metabolic pathway of the network, and $|b_s| = W_s$ (number of genes in the sth pathway). The genes in the network are clustered in metabolic pathways based on the metabolic model and on previous knowledge on

metabolic networks [60]. Each pathway performs a particular task in the metabolism, e.g., the citric acid cycle, the oxidative phosphorylation, or the pentose phosphate pathway.

As part of the sensitivity evaluation, PoSA generates: (i) gene-pathway mappings (GP), defined as a $L \times p$ matrix P, where the *(l,s)*th element of P is 1 if the lth gene controls a reaction of the sth pathway, and 0 otherwise; (ii) reaction-pathway mappings (RP), defined as a $n \times p$ matrix R, where the *(j,s)*th element of R is 1 if the jth reaction belongs to the sth pathway, and 0 otherwise.

PoSA is able to rank the pathways of the metabolic network by perturbing genes in terms of knockouts. All genes in a pathway b_s are perturbed randomly. The output(s) of the model is compared with the output without the pathway perturbation. We consider as output the vector of the fluxes after performing flux balance analysis and perturbing pathways. PoSA performs combinatorial perturbations, since gene knockouts are represented by binary variables.

We define the "pathway elementary effect" for the pathway b_s as

$$PEE_s = \frac{\|v(b_1, b_2, \ldots, b_{s-1}, \tilde{b}_s, b_{s+1}, \ldots, b_p) - v(y)\|}{\Delta_s}, \tag{2.4}$$

where $\|\cdot\|$ is the 2-norm (Euclidean distance), v returns the vector of flux rates, $\tilde{b}_s$ indicates a mutation on the pathway b_s, and consists of *flipping* bits indicating activation status of genes chosen randomly in b_s: if a bit is 0 (or 1), the permutation turns it to 1 (or 0). y is the starting genetic strategy in which we are interested in computing the pathway sensitivity. Δ_s is a scale factor defined as $\Delta_s = \frac{1}{W_s} \sum_{i=1}^{W_s} \tilde{b}_s(i),\ s = 1, \ldots, p$.

Note that if a gene $b_s(i)$ in b_s is set to 1, it is knocked-out in the model; therefore, Δ_s is low if there is a low number of knocked-out genes, and as a result PEE is inversely proportional to the number of knocked-out genes. This correctly detects large changes caused by pathways in which only a few genes have been turned off. In the following applications, y is the strategy where all genes are activated and fully working in the metabolism. However, for instance, y can be a strategy on the Pareto front after multi-objective optimization. Indeed, evaluating the pathway-based sensitivity of each solution on the Pareto front is a useful criterion to select the least sensitive Pareto-optimal strategy, which is therefore safer to implement *in vitro*.

For each pathway b_s, 40 iterations are performed; at each iteration, an elementary effect is computed using (2.4), where $\tilde{b}_s$ indicates that a number of bits (indicating gene activation status) are flipped in the sth pathway only. For each iteration, both the actual bits flipped in the knockout strategy (with respect to the initial strategy y) and the number of bits flipped is chosen randomly. The number of bits flipped is between 1 and 10% of the genes in the

pathway. After all the iterations are completed for the sth pathway, the average elementary effect μ and its standard deviation σ are computed across the iterations.

In Fig. 2.4 we show the pathway sensitivity analysis applied to a model of the algal metabolism of *C. reinhardtii* [37]. The distribution of elementary effects is obtained by randomly choosing the bits to flip (i.e., genes to turn off or on). If y represents the initial all-active strategy, the mean μ and the standard deviation σ are used as an indicator of which pathway should be considered important in the overall metabolism. Similarly to Morris' sensitivity analysis, high mean indicates a pathway with an important overall influence on the output, while high standard deviation indicates a pathway whose effect is nonlinear or highly dependent on the effect of other pathways. In general, in the (μ, σ) space, highly networked cell components (e.g., those of nucleic acids, amino acid, cofactors and energetic metabolism) are at the top right. Specific, often single-reaction and abundant components (e.g., those for bacterial walls, nitrogen, glutamic and carbohydrates) are at the bottom left.

As shown in Fig. 2.4, knockouts in the set of reactions representing transport of molecules in and out of the two organelles (mitochondrion and chloroplast) are detected to be the most important in shaping the flux distribution. Surprisingly, the key pathways of pyruvate metabolism and TCA cycle are less important than the two sets of transport reactions, while still being two of the most important pathways in the overall metabolism. Compared to these key pathways, the transport reactions show a smaller value of σ, therefore indicating that their role in shaping the value of the objective function is less dependent on the other pathways. Surprisingly, although glycolysis/gluconeogenesis is the central core of bacterial biomass and energy storage, it is not ranked high by PoSA. More generally, pathways can also be clustered according to their position in the PoSA plot, which indicates their actual metabolic relevancy towards the selected objective function v.

2.4 A new FBA model: the hydrogenosome

In several unicellular eukaryotes, including ciliates, fungi, and trichomonads, instead of the traditionally studied mitochondria, there is an alternate organelle called *hydrogenosome*. Hydrogenosomes are anaerobically functioning ATP-producing organelles of mitochondrial origin that represent a particular adaptation of mitochondrial metabolism, and possess the ability of producing molecular hydrogen by using protons as electron acceptor [164]. One of the best studied hydrogenosomes, and thus the one that we consider in our analysis, is that of the sexually transmitted human parasite *Trichomonas vaginalis* [106].

Hydrogenosomes are double-membrane bound organelles. Although they were first thought to be distinct and of independent origin, the finding of a chromosome-bearing

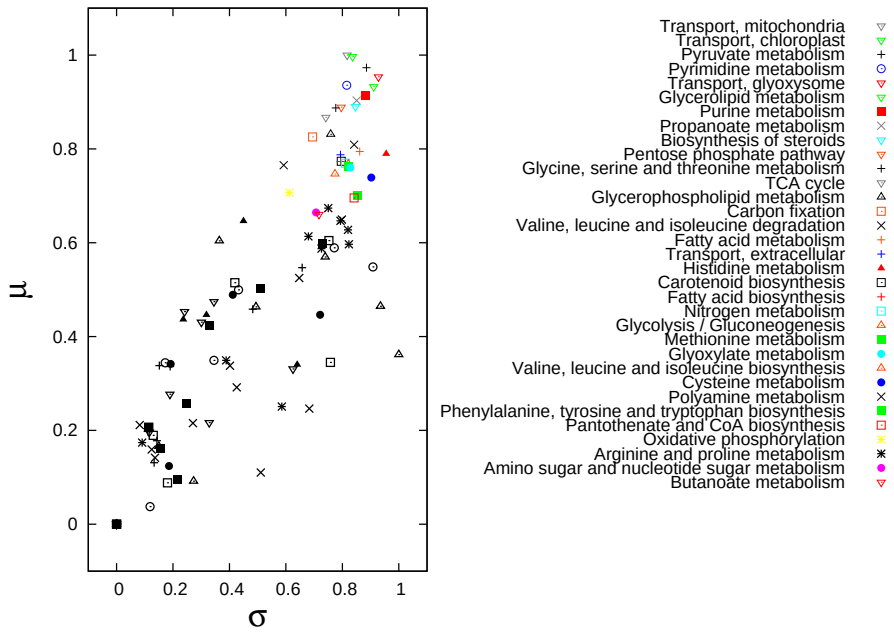


Fig. 2.4 PoSA applied to the algal metabolism of *C. reinhardtii*. Each perturbation is performed through on/off flipping of a number of genes, and the full vector of flux rates is evaluated as output v. For each pathway, the mean μ and the standard deviation σ are computed from the distribution of pathway elementary effects (2.4). In the legend, only the most sensitive pathways are reported.

hydrogenosome from the ciliate Nyctoterus ovalis, and the analysis of its sequence, proved their evolutionary relatedness with mitochondria [25]. Nowadays, it is widely accepted that hydrogenosomes are mitochondrial adaptations to anaerobic lifestyles appeared throughout the evolution of eukaryotes [164], although the mitochondrial ancestor (free-living bacteria that lived in the cell as endosymbionts) and the evolutionary driving force behind endosymbiosis are topics still open to debate. In addition to lacking mitochondria, all the organisms containing hydrogenosomes are facultative anaerobes.

The proteome of the hydrogenosome in *T. vaginalis* consists of around 200 different proteins; it is therefore significantly smaller than the proteome of yeast mitochondria. The hydrogenosome produces molecular hydrogen, acetate, carbon dioxide and ATP by the combined actions of pyruvate:ferredoxin oxido-reductase, hydrogenase, acetate:succinate CoA transferase and succinate thiokinase. In particular, pyruvate is broken down to acetate, CO_2, and molecular hydrogen. This process is coupled with ATP formation through phosphorylation. Since this organelle is not invasive, it remains outside the cell, thus it needs ATP to move and adapt to varying external conditions, e.g., to resist the immune system of the hosting organism.

The unifying features of mitochondria, mitosomes, mitochondrion-like organelles (MLOs), and hydrogenosomes are quite sparse. Almost all these organelles are linked by the metabolic process called *Fe-S cluster assembly*. The presence of cpn60 is also a unifying feature. Research on these mitochondrion-related organelles showed that ATP is not produced in all described mitosomes. Therefore, contrary to what was postulated at the beginning, energy production appears to be not the only driving force behind selection for organelle retention. Nevertheless, these organelles continue to keep at least part of the pathway involved in Fe-S cluster production [146]. It is noteworthy that ATP production in hydrogenosomes occurs through catalysis by succinyl CoA synthetase, a Krebs cycle enzyme that catalyzes the same reaction in hydrogenosomes and mitochondria.

Biochemical analyses of hydrogenosomes have also revealed significant differences between mitochondria and hydrogenosomes [29]. For instance, the enzyme responsible for the decarboxylation of pyruvate in hydrogenosomes, i.e. pyruvate/ferredoxin oxidoreductase, is significantly different from its counterpart in mitochondria, i.e. the pyruvate dehydrogenase complex. Likewise, the hydrogenase is possessed only by the hydrogenosome, and mitochondria cannot produce molecular hydrogen. Remarkably, pyruvate/ferredoxin oxidoreductase and hydrogenase are commonly found in anaerobic bacteria.

Recent research shows that the RuBisCO protein may be found in the hydrogenosome [144]. This finding can play a key role in understanding other hidden roles of RuBisCO in nature. Indeed, the hydrogenosome is the result of a genome reduction undergone by a common ancestor, and given that the RuBisCO is a vegetal protein (the most important protein in chloroplasts), apparently it is not needed in the hydrogenosome. Consequently, the presence of RuBisCO might suggest that it has a different role.

In order to investigate the hydrogenosome with the same techniques introduced before, we propose a model of the hydrogenosome metabolism in *T. vaginalis* and we evaluate the ATP flux. The model we propose is an FBA model that contains all the main reactions occurring in the organelle [117, 144] (see Table 2.2), as well as reactions dealing with the import of serine, glycine, pyruvate and malate into the hydrogenosome. Although our model already shows interesting behaviors, we plan to extend it by adding new reactions. To our knowledge, no hydrogenosome models are present in literature, thus we believe our model will have great value as the best description that we have to date of this organelle, and will provide a foundation upon which more accurate models can be built. A complete model of the hydrogenosomal metabolism would be of great biological relevance in the design of new antiparasitic drugs [106].

We specify the reaction network of the hydrogenosome through the LIM package of R, which allows generating the mass balance for each component. The list of reactions used

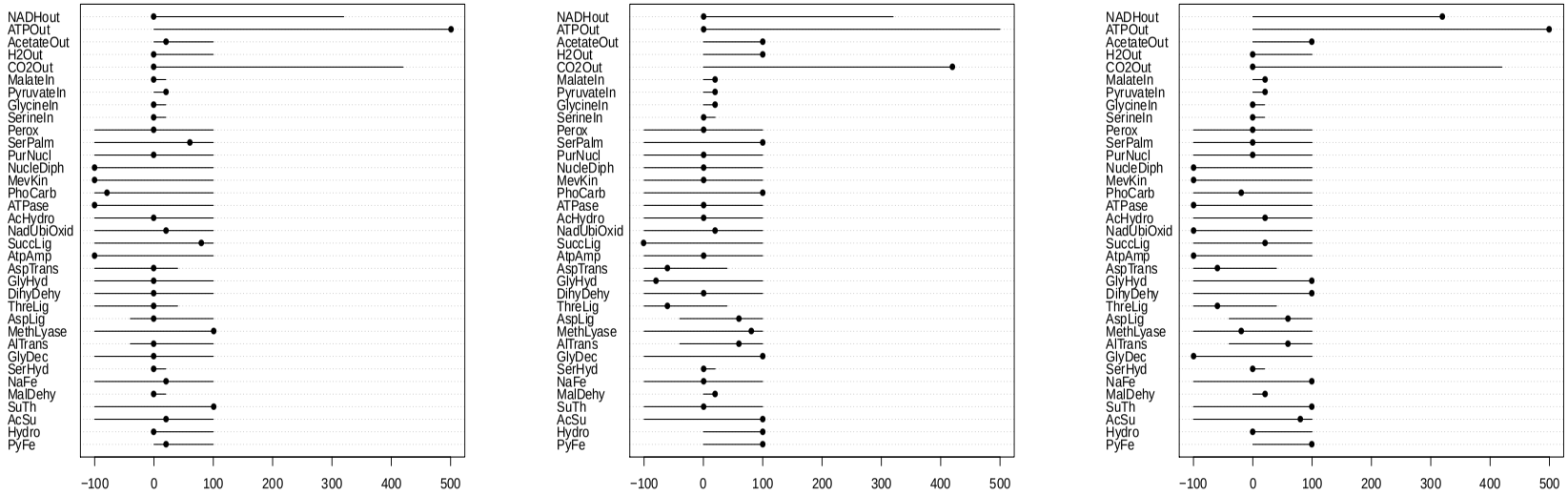
in the FBA framework is given in Table 2.2. Moreover, we estimate the optimal reaction rates in the FBA approach. In Fig. 2.5 we show the results of FBA carried out on the hydrogenosome model. The maximization of the ATP yield implies the consumption of all the H_2, CO_2 and NADH in the organelle. The maximization of CO_2 implies also the maximization of H_2, but requires the maximum import of malate, pyruvate and glycine. Moreover, it impairs the NaFe reaction, thus keeping the NADH at the initial value. The two-objective optimization of ATP and NADH causes the impairment of the Hydro reaction, thus the hydrogenosome cannot produce H_2. It is therefore evident that the hydrogenosome needs a Pareto-optimal trade-off between energy and H_2 production.

In order to obtain the optimal solution in all feasible two-objective optimizations, we randomly sample the solution space. In Fig. 2.6 we display the pairs plot and mark the optimal solution with a red dot. Most reactions are coupled together; it happens frequently that a reaction can occur only when the substrates, which are products of another reaction, are available. For instance, the ATP production depends on the acetate production, since ATP requires SuccinylCoA, which is a product of the same reaction that produces acetate. It is noteworthy that a few reactions are in conflict with all the other reactions. For instance, the NaFe reaction, which produces NADH, is in conflict with Hydro, which produces H_2. Indeed, the last row of the pairs plot shows that the NADH production is in conflict with H_2 production. The presence of a feasible point in this plot does not imply that the hydrogenosome metabolism, even if optimized, is able to reach that point. Notably, the CO_2-NADH plot shows that the hydrogenosome is versatile and can produce both, but it cannot specialize in producing only one metabolite, since there are less points near the axes and the Pareto front exhibits a higher curvature than the other fronts.

As shown in the previous sections, the FBA approach proves very useful when one needs to investigate the effect of gene/reaction knockouts. In order to impair a reaction, its lower and upper bounds can be set to zero. In Fig. 2.7 we show the output fluxes obtained re-running the metabolic network when the mevalonate kinase (MevKin) is impaired. Since the ATP is involved in this reaction, the maximization of ATP and NADH does not yield the same result of Fig. 2.5c for the output flux of ATP. Another possible use of our model is an insight into the optimization of more than two objectives (Fig. 2.8), which often requires finding trade-offs.

2.5 Related work and final remarks

Additional techniques (ε-dominance and identifiability analysis) to drive and complement the optimization process were excluded from this dissertation. The ε-dominance Pareto



(a) (b) (c)

Fig. 2.5 **FBA of the hydrogenosome metabolism.** We report the reactions of its metabolism and their fluxes (mmol h^{-1} gDW^{-1}) on the x axis. A line next to a reaction represents all the feasible flux rates for that reaction, while a point represents the value of the specific solution proposed in this configuration of the hydrogenosome. The three panels show the maximization of the ATP yield (a), the maximization of CO_2 (b), and the two-objective optimization of ATP and NADH (c).

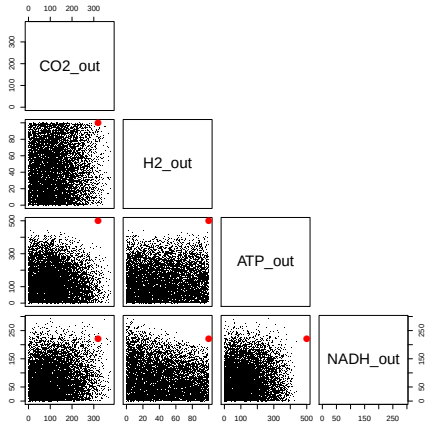


Fig. 2.6 **Two-objective optimization carried out on metabolites and reaction fluxes.** The points represent a Markov chain Monte Carlo (MCMC) sampling of the reaction network carried out with 20000 iterations, and allow finding trade-offs among the maximizations carried out in Fig. 2.5. When two reactions are mostly uncorrelated with one another, all the solutions are feasible, thus the sampling tends to a filled square (e.g., the output of H_2 and CO_2 shown in the upper-left corner). Conversely, when two reactions show correlation, the MCMC sampling of the square reveals their relationship. For instance, the NADH output and the H_2 output are in contrast with each other. The red dot is the optimal point.

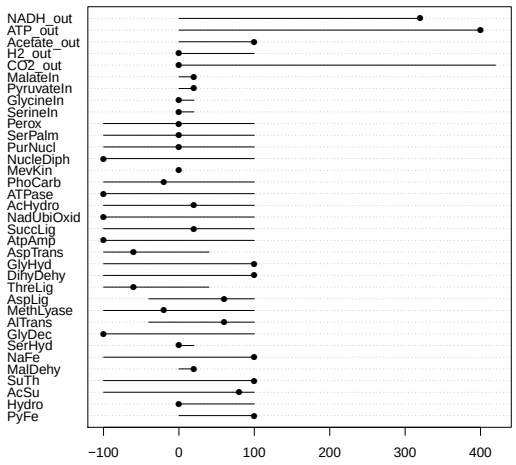


Fig. 2.7 **MevKin knockout analysis.** When the mevalonate kinase (MevKin) reaction is impaired, a knockout is performed on the metabolic network. As a result, the maximum value of ATP decreases with respect to Fig. 2.5c, where no knockouts were performed.

Abbrev.	Name	Equation
PyFe	Pyruvate:ferredoxin oxido-reductase	Pyruvate + CoA + 2*FerredoxinPos $<->$ AcetylCoA + CO2 + 2*FerredoxinNeg + 2*HPos
Hydro	Hydrogenase	2*FerredoxinNeg + 2*HPos $<->$ 2*FerredoxinPos + H2
AcSu	Acetate:succinate CoA transferase	AcetylCoA + Succinate $<->$ Acetate + SuccinylCoA
SuTh	Succinate thiokinase	ADP + Phosphate + SuccinylCoA $<->$ ATP + Succinate + CoA
MalDehy	Malate dehydrogenase (decarboxylating)	Malate + NADPos $<->$ NADH + CO2 + Pyruvate
NaFe	NADH:ferredoxin oxido-reductase	2*FerredoxinNeg + NADPos + HPos $<->$ 2*FerredoxinPos + NADH
SerHyd	Serine hydroxymethyl transferase	Serine $<->$ Glycine
GlyDec	Glycine decarboxylase	Glycine $<->$ CO2
AlTrans	alanine transaminase	LAlanine + 2Oxoglutarate $<->$ Pyruvate + LGlutamate
MethLyase	methionine gamma-lyase	LMethionine + H2O $<->$ Methanethiol + NH3 + 2Oxobutanoate
AspLig	aspartate-tRNA ligase	ATP + LAspartate + TRNAAsp $<->$ AMP + Diphosphate + LAspartylTRNAAsp
ThreLig	threonine-tRNA ligase	ATP + LThreonine + TRNAThr $<->$ AMP + Diphosphate + LThreonylTRNAThr
DihyDehy	dihydrolipoyl dehydrogenase	ProteinN6Ddihydrolipoyllysine + NADPos $<->$ ProteinN6Ddihydrolipoyllysine + NADH + HPos
GlyHyd	glycine hydroxymethyltransferase	510Methylenetetrahydrofolate + Glycine + H2O $<->$ Tetrahydrofolate + LSerine
AspTrans	aspartate transaminase	LAspartate + 2Oxoglutarate $<->$ Oxaloacetate + LGlutamate
AtpAmp	ATP:AMP phosphotransferase	ATP + AMP $<->$ 2*ADP
SuccLig	succinate-CoA ligase (GDP-forming)	GTP + Succinate + CoA $<->$ GDP + Phosphate + SuccinylCoA
NadUbiOxid	NADH:ubiquinone oxidoreductase	NADH + Ubiquinone + 5*HPos $<->$ NADPos + Ubiquinol + 4*HPos
AcHydro	acetyl-CoA hydrolase/transferase	AcetylCoA + H2O $<->$ CoA + Acetate
ATPase	H+-transporting two-sector ATPase	ATP + H2O + HPos $<->$ ADP + Phosphate + HPos
PhoCarb	phosphoenolpyruvate carboxykinase	GTP + Oxaloacetate $<->$ GDP + Phosphoenolpyruvate + CO2
MevKin	mevalonate kinase	ATP + RMevalonate $<->$ ADP + R5Phosphomevalonate
NucleDiph	nucleoside-diphosphate kinase	ATP + NucleosideDiphosphate $<->$ ADP + NucleosideTriphosphate
PurNucl	purine-nucleoside phosphorylase	PurineNucleoside + Phosphate $<->$ Purine + AlphaDribose1phosphate
SerPalm	serine C-palmitoyltransferase	PalmitoylCoA + LSerine $<->$ CoA + 3SehydroDSphinganine + CO2
Perox	peroxiredoxin	2*RSH + ROOH $<->$ RSSR + H2O + ROH

Table 2.2 **Energy-related reactions and amino acid metabolism pathways considered in our FBA model of the hydrogenosome.** Along with the exchange reactions, this set of reactions constitutes the metabolic network investigated using flux balance analysis and used to create an FBA model. Each metabolite is a node linked to other nodes through reactions, and represents a constraint in the FBA model, while the reaction rates represent the variables. Additional constraints are the lower and upper bound values for the variables of the model.

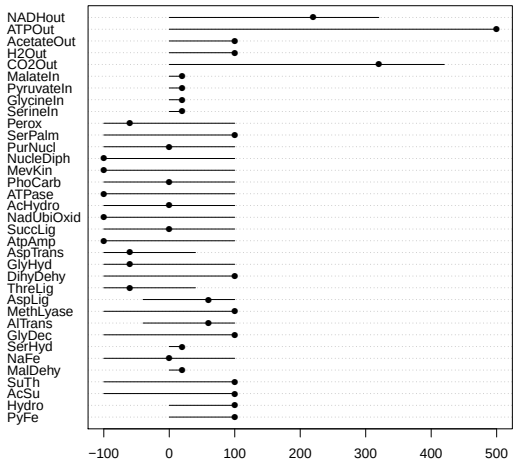


Fig. 2.8 **Trade-offs in the five-objective optimization of the hydrogenosome.** The simultaneous maximization of ATP, NADH, acetate, CO_2 and H_2 production is not possible without finding a trade-off. In this case, compared to Fig. 2.5b, the CO_2 cannot reach its maximum value due to the constraints of the metabolic network and to the concurrent maximization of the other four objectives.

front analysis extends the search space of solutions revealing other suitable (suboptimal) points. The Identifiability Analysis (IA) seeks the functional relations underlying the components of a given system, and can be used after the multi-objective optimization. Coupled with the sensitivity analysis, it gives insight into the model under investigation.

A component of a model is said to be *non-identifiable* if there is no unique solution for its estimation. The non-identifiability can be (i) *structural*, when there are functional relations among components and therefore they cannot be determined unambiguously, or (ii) *practical*, when the low amount or quality of data available does not allow one to have a good estimate for the component. From the definitions, it follows that if a model is structurally non-identifiable, it is also practically non-identifiable. Using repeated fitting to data and estimations of components, the IA is aimed at finding the structural non-identifiable components of a model, providing hints for simplifying the model and thus avoiding redundancy, or indicating where new experimental measures are needed to guarantee the identifiability of the model. In a related work, we use IA to characterize monogenic mitochondrial diseases. Our idea is that by performing IA in simulated healthy, pathological and disease conditions in a mitochondrial FBA model [151], we can characterize the onset of a disease by looking at the functional relations among flux rates. For instance, in a monogenic disease called α-ketoglutarate dehydrogenase deficiency, we simulate three

stages of the disease (healthy, inflammation and pathological stages) depending on the production of ATP. Interestingly, we find for the three stages that most functional relations between fluxes in the pathological stage have a different shape with respect to those found in the healthy and the inflammation stages (Fig. 2.9).

Part of this chapter has been published in Bioinformatics [44]. The optimization and the associated analyses were performed in collaboration with my co-authors. The hydrogenosome model, contributed by me, has been published in IEEE Transactions on Computational Biology and Bioinformatics [7]. The full pipeline, including ε-dominance and IA, has been published in PLoS One [10].

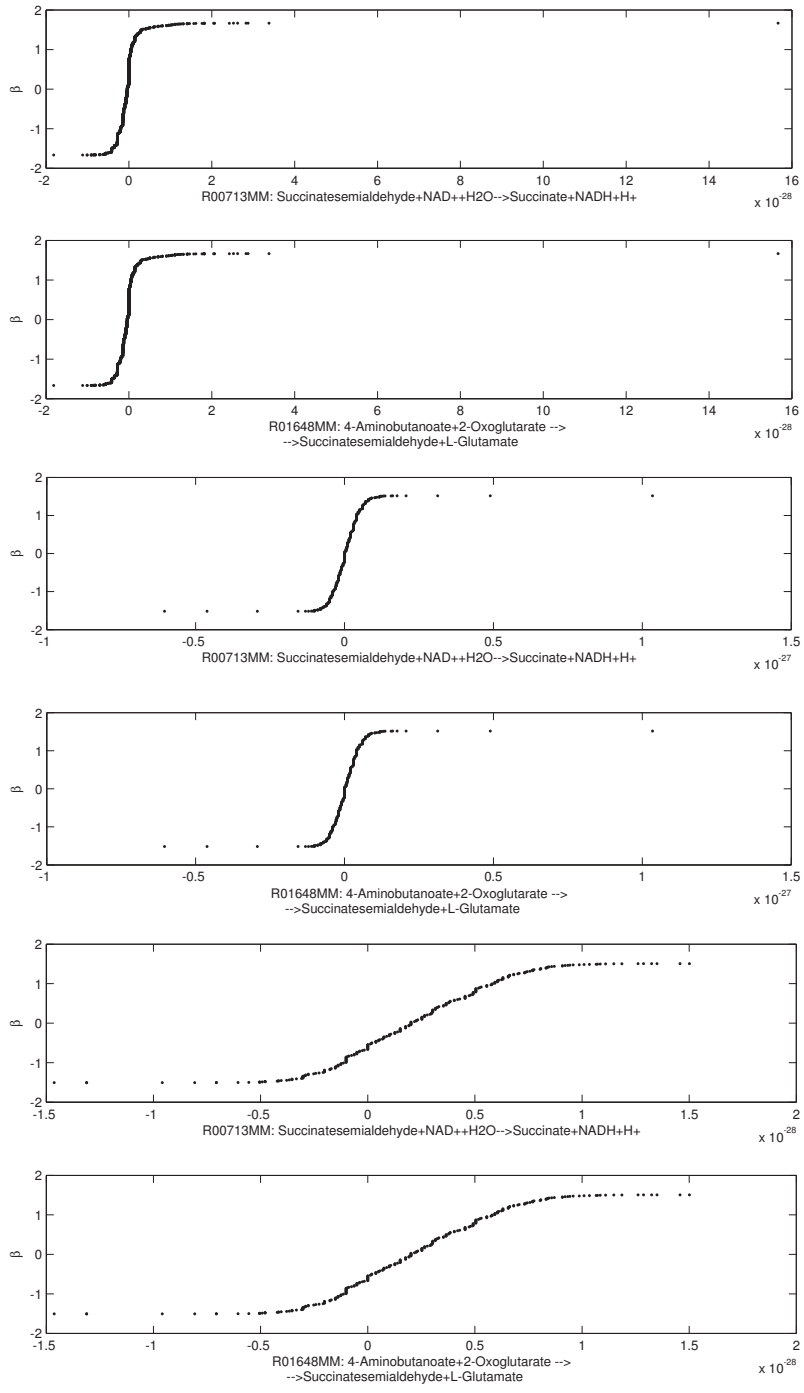


Fig. 2.9 **α-ketoglutarate dehydrogenase deficiency - healthy stage (top), inflammation stage (middle), pathological stage (bottom).** Similar behavior of β (y axis) indicates strong functional relation between reactions [10], whose flux rate is on the x axis. These relations have been found for the pair of reactions R00713MM and R01648MM (x axis) [μmol min^{-1} gDW^{-1}] in the mitochondrial FBA model [151]. The figure shows how different stages of the disease imply different shapes of β, and therefore different functional relation between the two reaction fluxes.

Chapter 3

METRADE: continuous optimization and environmental adaptability in multi-omic networks

In Chapter 2 we did not consider the environmental condition in which the bacterium is operating, and we also assumed that all genes are equally expressed. However, all organisms experience oscillating conditions that range from starvation to food richness. In particular, environmental changes represent the availability of different sources of food and constitute different growth conditions. Therefore, the optimization of multiple objectives should also indicate the capability of the organism to cope with these changes. To this end, in this chapter we propose a method to enable multi-omic flux balance analysis and a characterization of the space of environmental conditions.

We derive a multi-omic model for the *Escherichia coli* able to account for the adaptability to multiple environmental conditions, and for the temporal evolution towards the production of selected metabolites. The multi-omic model is part of a pipeline called METRADE (MEtabolic and TRanscriptomics ADaptation Estimator), where the response to the environmental conditions is then mapped to a multidimensional objective space and analyzed with various estimators. METRADE also finds the optimal gene expression profiles towards the multi-objective optimization of the production of chemicals of interest. Unlike GDMO, METRADE treats gene expression as real-valued variables representing expression levels, and not as Boolean gene knockouts.

3.1 Estimating bacterial adaptability in changing environmental conditions

Many molecular levels can contribute to adaptability: pathway structure, codon usage, metabolism. For instance, the structure of the metabolism and the pathway productivity can evolve in time due to varying environmental conditions or selective pressure [123]. Analogously, several recent examples show the coupling of codon usage to adaptive phenotypic variation, suggesting that the genotype functionality and behavior can be derived from the analysis of the evolution in the codon usage [94]. Typically, the correlation between gene expression and codon bias is large for environments similar to those in which the organism evolved, and small for dissimilar environments [169].

In this chapter, we explore bacterial adaptability by investigating experimental conditions mapped to a multidimensional objective space. To obtain a phase-space of conditions, we add the continuous gene expression and the codon usage layers to an FBA framework, therefore proposing a new multi-omic FBA model. As a first result, we are able to optimize these layers for the overproduction of metabolites of interest, predicting the short term bacterial evolution towards the optimum. Then, we present a new method to map gene expression profiles to any metabolic objective space. Since each profile is associated with a growth condition, the objective space becomes the condition phase-space, which we investigate through principal component analysis.

To build the multi-omic model, we map gene expression and codon usage to the metabolism by proposing a bilevel formulation that defines the flux bounds as a continuous function of the related expression data. We therefore provide a different model for each gene expression profile (environmental condition). This step is highly customizable in that it is possible to select a different map for each reaction in the model, thus allowing for the introduction of additional omic data, e.g., protein localization or stochasticity in the protein abundance. The type of reaction-specific information that can generate a custom map for given reactions is, for instance, the four-fold activity reduction in the isocitrate dehydrogenase enzyme when taking acetate as the carbon source [128]. In other words, we are able to generate a model tailored to any specific environmental or internal condition. The model can be further optimized by finding the optimal codon usage for a given array of gene expression.

To optimize the multi-omic layers, we propose a genetic multi-objective optimization algorithm that seeks the gene expression profiles that optimize multiple cellular functions concurrently. We use the Pareto front as a tool to seek trade-offs between two or more tasks performed by *E. coli*, and specifically to score the performance when the tasks are

contending with one another. We simultaneously optimize two or more tasks by finding the best gene expression profile and codon usage array. Most notably, this may permit to determine the best environmental condition in which a bacterium has to be grown in order to reach specific optimal output values measured in the range of objective functions chosen by the researcher. As a particular case, it is also possible to investigate the best single or multiple gene knockouts for the given set of objectives [158].

The rest of the chapter is organized as follows. First, we define the new multi-omic model by adding gene expression and codon usage layers to FBA, in order to build the level of information required for a meaningful understanding of the landscape of experimental conditions. Using this augmented FBA framework, we optimize the model and we perform a temporal analysis of bacterial evolution towards an optimal configuration using the hypervolume indicator. Then, we introduce principal component analysis to identify conditions mapped to close regions in the phase-space. The main steps of our pipeline, named METRADE, are illustrated in Fig. 3.1, with pointers to the sections where each step is performed.

The idea behind METRADE is that an accurate prediction on the relations among conditions should not disregard a multi-omic model to associate conditions to a phenotypic outcome in a set of objective spaces; indeed, a multi-omic inference cannot be performed by looking only at the gene expression profiles, but requires a mapping to the phenotype that estimates the actual effect of each condition on the bacterial physiology. Unlike GDMO, METRADE searches in the continuous space of gene expression levels. Therefore, it is not limited to knockout analysis, but also includes gene overexpression and partial knockdown. While the applications of METRADE shown in this chapter are on two-dimensional objective spaces, METRADE can be applied to spaces of more than two dimensions. In Section 3.8, we validate METRADE against a publicly available phenomics dataset of *E. coli*, and against experiments performed on cultures of *Corynebacterium glutamicum* and *Saccharomyces cerevisiae*.

To the best of our knowledge, very few approaches have been developed to take into account non-discretized gene expression levels in constraint-based models [23]. The key advantage of a continuous genotype-metabotype map is that one can reverse it, obtaining an enviromics map, which allows identifying the environmental factors leading to a pre-specified metabotype. For all we know, no prior studies have accounted for codons and combined Pareto-optimization with real-valued gene expression levels, codon usage bias, and component analysis. This paves the way towards predicting and optimizing bacterial adaptability across conditions and over time. METRADE is freely available as a toolbox extension of COBRA 2.0 [143], the most widely used MATLAB toolbox to perform FBA.

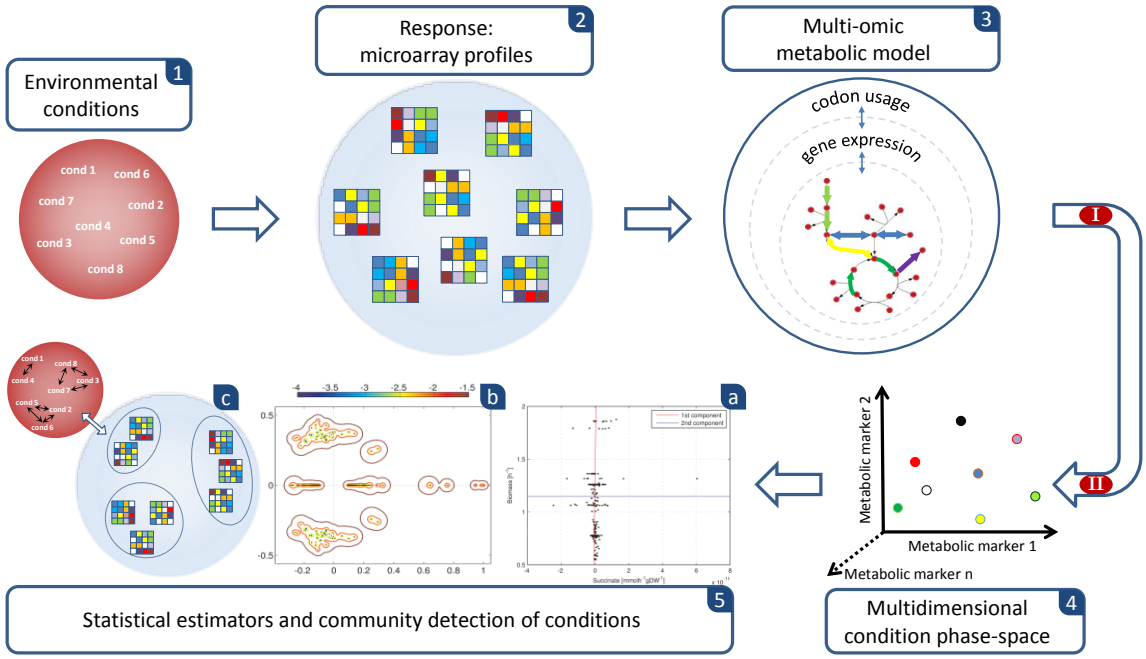


Fig. 3.1 **Pipeline of** METRADE **(MEtabolic and TRanscriptomics ADaptation Estimator). Part I (panels 1-3).** The response to environmental conditions (1) in which *E. coli* is grown (e.g., low or high glucose, aerobic or anaerobic, pH changes, antibiotics, heat shock) is measured through Affymetrix Antisense2 microarray expression profiling (2). To evaluate the environmental conditions and detect their community structure, we derive a multi-omic model (3) of the *E. coli* metabolism, taking into account gene expression (Section 3.2) and codon usage (Section 3.4).
Part II (panels 4,5). (4) The multi-omic model enables us to account for multiple growth conditions and temporal multi-objective evolution towards the production of selected metabolites through the Pareto front (Section 3.3). It is also able to associate each environmental condition with a single point in a multidimensional condition phase-space (Section 3.5). The adaptability to one condition is given by the time evolution of the bacterial genome, which can be estimated by the hypervolume indicator (Section 3.6). (5) A set of techniques can be applied to the multi-omic model with the aim of analyzing the adaptability to experimental conditions. We apply principal component analysis (5a) to the condition space in order to investigate the directions with largest variance (Section 3.7). A more detailed analysis of the condition space can be obtained through a distance matrix built on the condition phase-space (5b), and a spectral method for community detection to infer similarities among growth conditions (5c) [11].

3.2 A novel method for integration and optimization of gene expression in FBA

Each reaction in an FBA model depends on a single gene set, represented by a set of genes linked by AND/OR operators. For instance, when a gene set is composed of two genes in an AND relation, both are necessary to catalyze the corresponding reaction, and knocking out only one gene is sufficient to knock out the reaction. In this case, the gene set represents an *enzymatic complex*. Conversely, when a gene set is composed of two genes in an OR relation, the two genes synthesize for *isozymes*, which differ in the amino acid sequence, but catalyze the same reaction. Therefore, one gene is sufficient to catalyze the reaction, and all the genes must be knocked out in order to knock out the reaction.

With the aim of overcoming the limitations offered by the Boolean knockout approach, we need to formalize the AND/OR relation between genes using a real-valued map that enables us to define a variable called "gene set expression" as function of the gene expression. Let $x_i^j,\ i = 1,\dots,p$, be the gene expression levels of the genes $s_i^j,\ i = 1,\dots,p$, and let $\wedge_i s_i^{(1)}$ and $\vee_i s_i^{(2)}$, $i = 1,\dots,p$, be two basic gene sets modeling an enzymatic complex and an isozyme respectively. We adopt the following map τ between a gene set and its expression:

$$\bigwedge_{i=1,\dots,p} s_i^{(1)} \overset{\tau}{\longmapsto} \min_{i=1,\dots,p} \{x_i^{(1)}\}, \tag{3.1}$$

$$\bigvee_{i=1,\dots,p} s_i^{(2)} \overset{\tau}{\longmapsto} \max_{i=1,\dots,p} \{x_i^{(2)}\}. \tag{3.2}$$

Specifically, the expression level of an enzymatic complex, which needs all its genes to work properly, is constrained to be equal to the lowest of the expression levels of its genes. Conversely, the expression of an isozyme, which needs at least one of its genes, is the largest of the expression levels of its genes. The bounds of a reaction catalyzed by an enzymatic complex will be function of the minimum expression level of its genes, while the bounds of a reaction catalyzed by isozymes will be function of the maximum expression level of its genes. Nested gene sets are treated with the same methodology, i.e., applying (3.1) and (3.2) recursively.

We run the model to find the distribution of fluxes that optimizes multiple metabolic markers (e.g., natural and synthetic objectives). As the bounds of the fluxes depend on the gene expression, we define the following bilevel linear program:

$$\begin{array}{lll} \max & g^{\mathsf{T}} v & \\ \text{such that} & \max & f^{\mathsf{T}} v \\ & \text{such that} & Sv = 0 \\ & & v_i \geq v_i^L \cdot h(y_i) \\ & & v_i \leq v_i^U \cdot h(y_i) \end{array} \tag{3.3}$$

where f and g are n-dimensional arrays of weights associated with the first and second objective that will be selected for the optimization, and indicate how much the reaction fluxes in the vector v contribute to the objective function. v_i^L and v_i^U are the minimum and maximum flux of the wild-type configuration of the model. In the present dissertation, f and g are Boolean vectors. For instance, $f_j = 1$ and $g_k = 1$ if and only if the fluxes v_j and v_k have to be maximized as first and second objective respectively. In order to define the function h, let y_i be the gene set expression of the ith gene set, responsible for the ith reaction of the model. To map the gene set expression value into a specific condition of the model, we use the following piecewise multiplicative function:

$$h(y_i) = \begin{cases} (1 + |log(y_i)|)^{\text{sgn}(y_i - 1)} & \text{if } y_i \in \mathbb{R}^+ \setminus \{1\} \\ 1 & \text{if } y_i = 1 \end{cases} \tag{3.4}$$

where $\text{sgn}(y_i - 1) = (y_i - 1)/|y_i - 1|$.

In the FBA model, we replace the minimum and maximum flux of the ith reaction with $v_i^L \cdot h(y_i)$ and $v_i^U \cdot h(y_i)$ respectively, where v_i^L and v_i^U are the minimum and maximum flux of the wild-type configuration of the model. This choice is consistent with the fact that all the gene expression values in various conditions are relative to those of the wild-type bacterium. The gene set expression level is transformed by the logarithmic function $h(\cdot)$ so as to avoid that the genetic algorithm that we will use to perform the multi-objective optimization is driven towards high and unfeasible values of gene expression. This approach is in keeping with the "lazy step function" found in bacteria, yeast and human cells. According to this function, the mRNA levels are good indicators for the abundance of a protein (especially when averaging across populations), while post-transcriptional, post-translational and degradative regulations may fine-tune the protein abundance through miRNA [166].

On large samples, the correlation between mRNA level and protein abundance has been shown to be more evident with principal component analyses [72] and especially for

highly expressed genes, where the amount of noise is small and the correlation is high [51, 105, 169]. Overall, in single living organisms, the correlation is good except when proteins are long lasting and mRNA short lived. Furthermore, quantitative proteomics and transcriptome profiling (RNA-seq) seem to prove that there is high association between mRNA and protein levels, indicating that mRNA measures can be used as approximation for protein abundance [48, 71, 108]. In METRADE, we link the gene expression profiles with the FBA fluxes of the associated reactions in *E. coli* defining a real-valued adjustable map. Given that protein synthesis is an outcome of the expression of genes coding for protein segments, we link the gene expression values to the flux of the reactions controlled by the proteins coded by those genes. We share with E-Flux the approach of modifying the upper and lower bound of the reactions according to the gene expression value of the associated genes [42]. However, we use a logarithmic map to relate transcript level to flux bounds.

Due to the lack of direct correspondence between flux rate and transcript level (translation, degradation, and post-translational modifications are ignored), differential expression levels are adopted here. Therefore, the expression profile of a "wild-type bacterium" is considered as a control condition, and is associated with the original genome-scale model with unchanged lower and upper bounds.

While h is adjustable depending on the cell type or bacterial strain, we suggest a logarithmic map for a number of reasons. Specifically, the increase in the protein synthesis rate is fast with increasing mRNA abundance, but slower for large values of mRNA abundance [62]. Furthermore, adopting a logarithmic function in combination with the optimization algorithm avoids setting unrealistically high values of measured gene expression levels, which would be converted into weak constraints if using, e.g., a linear map. A key advantage of our map h is that it is roughly linear in the neighborhood of 1 (the wild-type gene expression level); this property models the experimental findings of high correlation and a roughly linear relation between gene expression and enzyme activity for the wild-type *E. coli* [128, 147]. Finally, several empirical evidences support our assumption [71, 125]. We remark that we use the logarithmic map to set constraints for the metabolic model, then we solve the linear program (3.3) to find the flux distribution.

A reaction whose related genes are under-expressed, is impaired through narrowing its flux bounds according to the related gene expression values, rather than completely removed from the model as done in the common approaches integrating genomics and FBA. Analogously, if those genes are over-expressed, the domain of the reaction flux is automatically enlarged, rather than left unchanged. The idea behind the integration of gene expression implemented in METRADE is presented in Fig. 3.2. We assume that the genetic

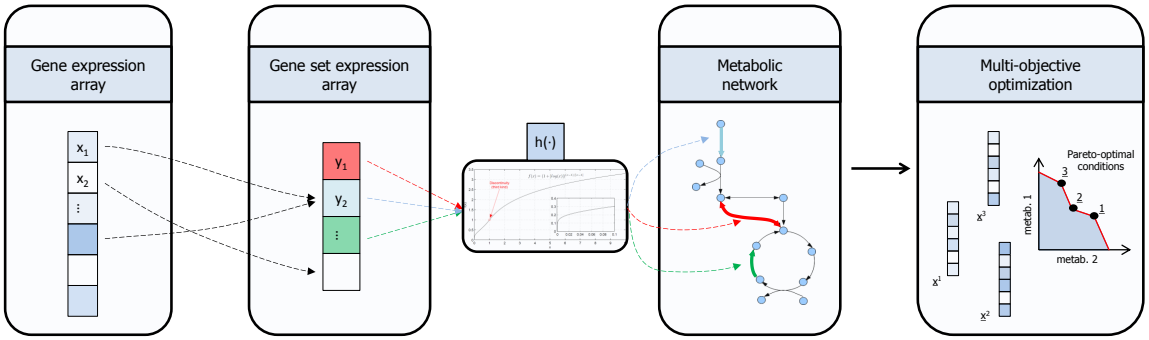


Fig. 3.2 **Flowchart of the integration of gene expression implemented in** METRADE**.** Each reaction of the metabolic network constituting the FBA model depends on a gene set. Therefore, the flux of each reaction depends on the corresponding gene set expression, computed taking into account the expression of the genes in the gene set, and their AND/OR relation. The real-valued function h converts gene set expression values into bounds for the bilevel FBA model. As a result, the decision variables for the multi-objective optimization are the gene expression values.

level is slower than the metabolic one, and therefore the steady state is reached faster than the variation of enzyme concentrations due to changes in the gene expression profile [153]. The *E. coli* metabolism is assumed to be at a steady state that relies on environmental factors and is reached quickly if compared to the variation of enzyme concentrations due to changes in the gene expression profile. Since the gene expression data are used as indicators for the activity of the associated reactions in the model, each gene expression profile is associated with a point in a multidimensional objective space.

Note that the outer maximization problem in (3.3) is subject to the inner one. More specifically, the inner maximization finds the distributions of flux in the network such that the growth rate (first objective) is maximized. In the outer maximization, all the unregulated fluxes are then distributed such that the second objective is maximized. The lower and upper bounds of the ith flux v_i depend on the expression of the genes involved in the ith reaction. The bilevel problem is finally converted to a single-level problem, and solved using the GLPK solver. It is straightforward to check that all the approaches based on Boolean gene knockouts become a particular case of METRADE, being $h(y_i) \xrightarrow{y_i \to 0} 0$ and $h(1) = 1$.

3.3 Multi-objective optimization of multi-omic models

We optimize the metabolic model through a multi-objective evolutionary algorithm, reaching an optimal configuration, according to the definitions in Section 1.4. In the Boolean evolutionary approach GDMO (Section 2.1), each individual is a strain represented by

a binary variable set representing the knockout strategy of gene sets [44]. Conversely, in METRADE the individuals are arrays of real values, each of which represents the expression level of a gene. We refer to these real-valued arrays as *gene expression arrays*. Through the function $h(\cdot)$, the gene expression arrays have a continuous effect on the FBA model, rather than only an on/off effect on reactions as in the Boolean approaches. Therefore, we are able to simulate cases where a lowly-expressed gene does not completely turn off the corresponding reaction (partial knockdown) and, analogously, a highly-expressed gene is able to increase the upper limit of the reaction flux (overexpression).

For the multi-objective optimization, we develop a parallel genetic algorithm (PGA) inspired by NSGA-II [52]. In our approach, both mutation and cross-over are allowed. A mutation represents a change in one or more gene expression levels, and proves useful to avoid the premature convergence to local optima. A cross-over consists of merging two parents to generate a child, exploiting the principle that two good solutions are able to generate a better solution. After generating an offspring population, a new population is formed selecting the best individuals from the parents of the previous generation and the current offspring. The technique of considering the best individuals of the old population is called *elitism* and ensures that good individuals are not lost during the evolution. The new population can then be subjected to a new round of mutation, crossover and selection. Our PGA is parallel, easy to use, and suited for black-box analysis. For each generation of the algorithm, we provide the Pareto optimal solutions, in order to evaluate the evolution of the Pareto front. This loop is repeated until the solutions set does not improve, or until an individual with a desired phenotype is achieved. The number of generations and the cardinality of the population are parameters chosen by the user. In our experiments we consider 1500 populations of 1000 individuals each, in order to ensure an extensive exploration of the objective space. Each point of the Pareto front is not merely a specific optimal model in the objective space, but also a gene expression array representing a specific genotype in the variable space. All the computations have been carried out on a machine with two 2.66 GHz 6-Core Intel Xeon processors and 64 GB of RAM.

We test METRADE on the *i*JO1366 *E. coli* metabolic reconstruction [119], consisting of three compartments (cytoplasm, periplasm and extracellular space), 1805 metabolites, 1366 genes, and 2583 reactions (including exchange and biomass reactions). The flux through the biomass reaction represents the rate at which the bacterium produces those metabolites necessary for its growth (e.g., amino acids, lipids, cofactors and proteins). The stoichiometry of the biomass reaction is scaled so that its flux rate equals the exponential growth rate of the bacterium. The objective functions taken into account are the fluxes representing the production of acetate, succinate, 1,2-propanediol and biomass (the major

players in synthetic biology of *E. coli*). We start from a gene expression array equivalent to the case in which all the bounds of the fluxes are left unchanged with respect to the initial model (wild-type bacterium). In Fig. 3.3 we show the regions of objective space discovered by the genetic algorithm from the first to the last generation for anaerobic and aerobic conditions. The PGA starts from an array of gene expression that, when translated into flux bounds, gives the default lower and upper bound of the initial model. As a case study, we maximize the acetate and biomass production, and succinate and biomass production. As mentioned in Chapter 2, both acetate and succinate are key molecules in biotechnology, with multiple industrial applications [131].

In the anaerobic case, we also apply the same approach with a Gaussian noise added to the initial values of the decision variables in order to avoid getting trapped in local maxima. Interestingly, in the acetate-biomass case (Fig. 3.3e), the area under the Pareto front is larger, and the number of optimal solutions is increased with respect to the case where no perturbations are applied. Furthermore, the Pareto front exhibits a curvature, although the extreme points (maximum acetate and maximum biomass) are conserved. In the succinate-biomass case, the same initial perturbation allows for a better coverage of the two-dimensional objective space (Fig. 3.3f), with a new extreme point of maximum succinate.

Without oxygen, the *E. coli* is able to grow with a maximum rate of 1.04 h^{-1}, compared to 1.24 h^{-1} reached in presence of oxygen. Nevertheless, the production of succinate in anaerobic conditions is increased, especially when searching the optimal gene expression profile starting from the initial array with added noise. The gene expression profiles optimized towards maximum succinate production yields 17.14 mmol h^{-1} gDW^{-1} (millimoles per gram of dry weight per hour) but no biomass. A more interesting solution is 6.38 mmol h^{-1} gDW^{-1} of succinate with 0.18 h^{-1} of biomass. The maximum amount of biomass that can be achieved with a nonzero succinate production (0.34 mmol h^{-1} gDW^{-1}) is 1.04 h^{-1}.

A similar pattern emerges when maximizing acetate production and biomass. Specifically, in aerobic conditions, the maximum biomass is 1.26 h^{-1} and the maximum acetate is 15.56 mmol h^{-1} gDW^{-1} (not taking into account the extreme solution with no biomass). Conversely, in anaerobic conditions, the maximum biomass is 1.04 h^{-1} (with 4.36 mmol h^{-1} gDW^{-1} of acetate production), while the maximum acetate is 19.86 mmol h^{-1} gDW^{-1}. Interestingly, both conditions share the same intermediate trade-off points with acetate production between 4.36 and 15.56 mmol h^{-1} gDW^{-1}.

The main difference between anaerobic and aerobic conditions, especially when maximizing succinate as synthetic objective, is the amount of succinate produced with an acceptable growth rate. The succinate in the cytoplasm takes part in 26 metabolic reactions.

When no oxygen is imported, only five reactions are activated: succinate is a product of Succinyl-diaminopimelate desuccinylase, O-succinylhomoserine lyase, and Fumarate depended dihydroorotate, and a reactant of Succinate dehydrogenase and Succinyl-CoA synthetase. Ten transport fluxes are responsible for transferring succinate in the periplasm and in the extracellular space. In anaerobic conditions, the transfer is performed by proton antiport.

3.4 Integration and optimization of codon usage in FBA

Manipulating and co-optimizing the gene expression and the codon usage of a bacterium enables us to design strategies for overproduction of relevant compounds, from therapeutic and industrial standpoints (e.g., amino acids and alcohols) [111]. This is of key importance when aiming at producing desired products through biosustainable processes. The idea underlying our approach is that even if two gene expression profiles are identical, the organism has the possibility to optimize its codon usage with small variations, allowing a co-optimization for a given set of objectives.

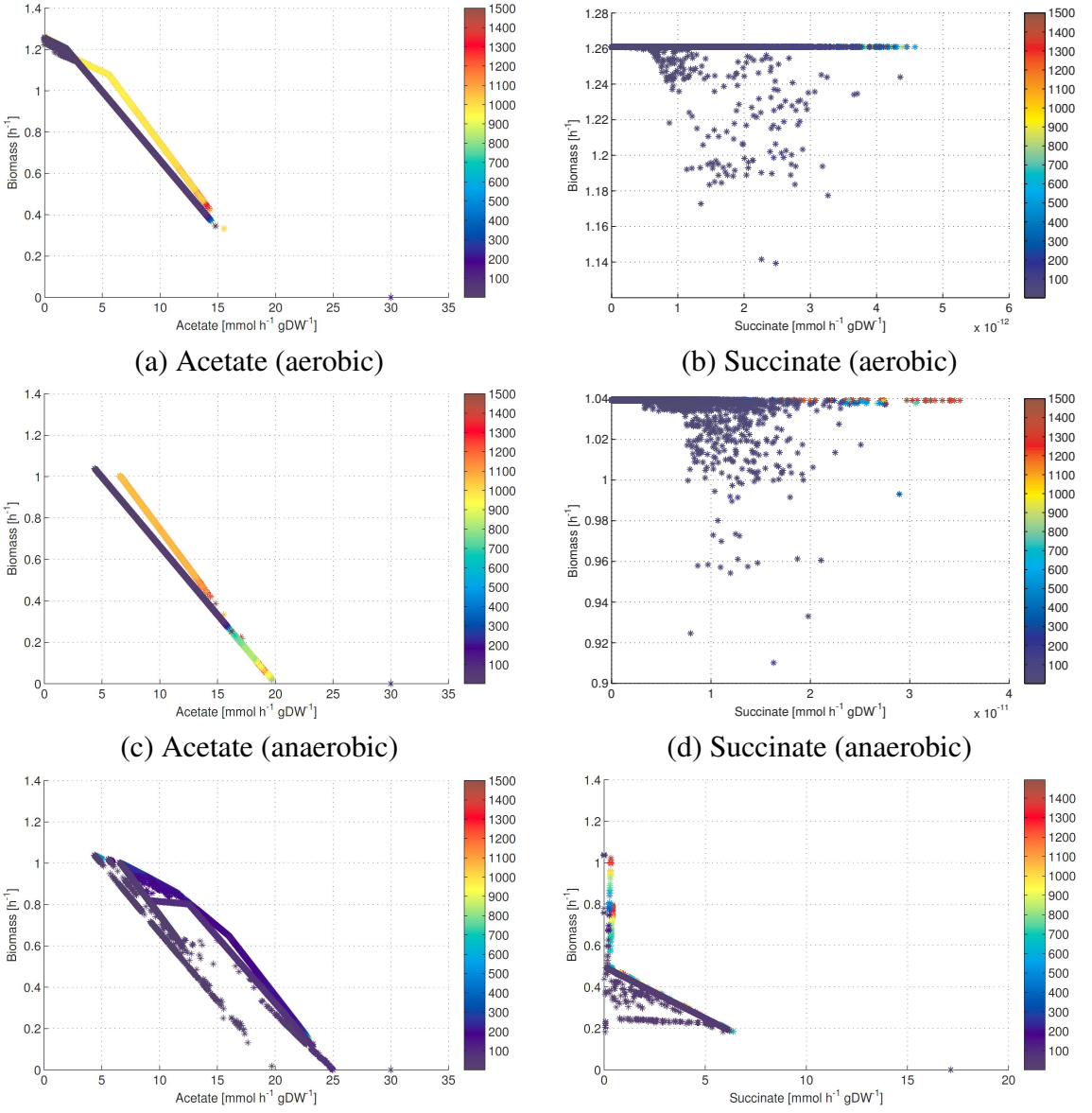
To account for the codon usage frequency in the translation process, we analyze a simplified situation where genes are made up of a slow codon $c^{(1)}$ and a fast codon $c^{(2)}$, read by two tRNAs with abundance a_1 and a_2 respectively [16, 93]. Let us denote by $c_i^{(1)}$ and $c_i^{(2)}$ the slow and fast codon usage of the ith gene. In each gene g_i, $c_i^{(1)}$ and $c_i^{(2)}$ can be used independently from one another. Since the total usage of a codon by all the genes depends on the abundance a of the corresponding tRNA, we set new constraints in our model:

$$\sum_i c_i^{(j)} = a_j, \tag{3.5}$$

where $j = 1,2$ ranges over the codons, and i ranges over the genes.

In order to optimize the codon bias, we let the codon usage values $c_i^{(j)}$ evolve using the PGA, and the constraints (3.5) hold at each generation of the algorithm. Then, the codon usage bias influences the rate of protein synthesis, and therefore the related reaction flux. If the fast codon is used more than the slow one, the translation process becomes faster. The production of the protein will be boosted by fast codons also because more ribosomes could operate on the same mRNA. To achieve this, the value y_i representing the gene set expression in METRADE is modified accordingly, thus obtaining a final variable z_i that includes the effects of the gene expression and the codon usage bias:

$$z_i = y_i - \alpha_i c_i^{(1)} + \beta_i c_i^{(2)}, \tag{3.6}$$



(a) Acetate (aerobic)

(b) Succinate (aerobic)

(c) Acetate (anaerobic)

(d) Succinate (anaerobic)

(e) Acetate (anaerobic, initial perturbation)

(f) Succinate (anaerobic, initial perturbation)

Fig. 3.3 **Pareto front produced by** METRADE **when maximizing succinate, acetate and biomass production.** On low succinate, slight variations of succinate flux cause step variations of biomass (plot (f)). The initial perturbation (e,f) is applied in anaerobic conditions on the first candidate strains and improves the convergence of the algorithm, thus permitting to avoid local maxima and to increase the coverage of the objective space. As a result, we discover a new area not explored by the algorithm applied in (c,d), including a new maximum for the succinate production. Solutions are denoted by progressively warmer colors according to the time step of the PGA in which they have been generated adaptively from the starting point.

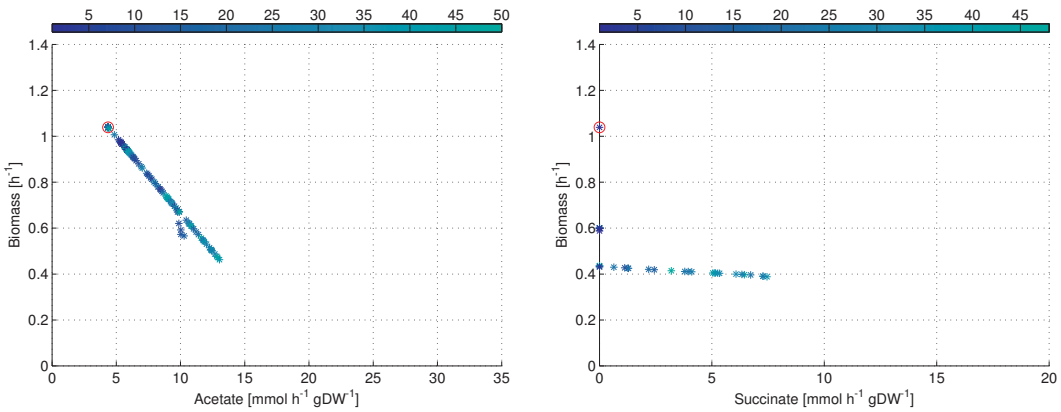
where $c_i^{(j)}$ are the values representing the codon usage (variables for the multi-objective optimization), while α_i and β_i are multiplicative constants that can be used to increase the effect of the slow codon with respect to the fast one, or vice versa. We set $\alpha_i = \beta_i = 50$, $\forall i$, so as to obtain a noticeable effect of the codon usage even with a small number of generations from the optimization algorithm (with different values of α_i and β_i we obtained different speed of convergence but the same shape of Pareto front). These two parameters and the equation (3.6) can be used to adjust the strength of the codon bias on the overall protein synthesis. As in Section 3.3, the multi-objective optimization algorithm drives a bilevel linear program that computes the flux distribution, and two of these fluxes are considered as the objective functions of the optimization algorithm. Here we compute the flux distribution through the following bilevel program:

$$\begin{array}{lll} \max & g^\intercal v & \\ \text{such that} & \max & f^\intercal v \\ & \text{such that} & Sv = 0 \\ & & v_i \geq v_i^L \cdot h(z_i) \\ & & v_i \leq v_i^U \cdot h(z_i) \end{array} \tag{3.7}$$

where f, g, v_i^L, and v_i^U have the same role as in (3.3), while h is defined in (3.4).

In this formulation, we do not take into account clusters and positions of rare codons. The definition (3.6) models the effect that the codon usage bias has on the final expression, and mimics the fact that the codon usage strongly affects translation rate and protein production [132]. In this regard, it has been recently shown that a change in the synonymous codon usage in an N-terminal peptide may result in an increase in the protein abundance up to 60-fold [69].

Once a point in the objective space has been found by METRADE starting from the associated gene set expression profile y_i, flexible codon usage $\left(c_i^{(1)}, c_i^{(2)}\right)$ allows exploring the neighborhood of that point, using the Pareto front and the optimality principle to seek solutions. In Fig. 3.4, we present the optimization of acetate, succinate and biomass production starting from a wild-type *E. coli* strain and using the codon usage values as decision variables, i.e., as input variables constituting the individuals in the multi-objective optimization process. In the acetate-biomass optimization, the acetate production increases linearly with decreasing biomass. In the succinate-biomass optimization, a slight change in the codon usage towards an increased succinate production causes the growth rate to drop from 1.04 h^{-1} to 0.60 h^{-1}. A further drop (0.39 h^{-1}) allows producing 7.45 h^{-1} gDW^{-1} of succinate.



(a) (b)

Fig. 3.4 **Optimization of codon usage for maximization of acetate and biomass production (a), and succinate and biomass production (b) starting from a wild-type** *E. coli***.** The Pareto front is obtained by applying our multi-objective optimization routine to the variables representing the codon usage, treated as input variables. We let METRADE run for 50 generations, which a preliminary analysis proved sufficient to obtain a Pareto front spanning the objective space. We denote solutions by progressively lighter colors depending on the generation in which they have been found. The red circle corresponds to a fixed array of gene expression levels (a wild-type *E. coli* strain).

While gene expression maps the external or internal condition of the organism at the transcription level, the codon usage maps quick alterations to fine tune the amount of protein produced at the translation level. A possible application of the gene expression and codon usage co-optimization is to evaluate the ratio between the gene expression and the codon usage variations with respect to a given non-optimal condition. This ratio can be computed for every pathway and exploited to highlight the difference among pathways. As a result, the Pareto front becomes a promising tool to investigate a model from a multi-omic standpoint.

3.5 Mapping genotype-phenotype associations to multidimensional objective spaces

Another useful feature of METRADE is the possibility of mapping a gene expression microarray profile directly to a bidimensional space of objective functions (e.g., acetate, biomass and succinate). Compared to Boolean associations between gene presence/absence and reaction activation/inactivation, this feature is extremely useful in the sense that it allows continuous modulation of the output, and even the effect of small variations of gene expression level is captured, and could have a significant impact on the metabolism.

Here we use a compendium of 466 *E. coli* Affymetrix Antisense2 microarray expression profiles by Faith et al. [58]. The dataset includes data collected in different media and different conditions, such as pH changes, antibiotics, heat shock, varying glucose and oxygen concentrations.

The idea behind our approach is that each condition yields a particular gene expression profile, which we convert into constraints for the FBA model in order to evaluate the condition-specific metabolic response. After defining the synthetic and natural objectives, we run the model and for each condition we obtain a point in the selected objective space. The model is run with an oxygen and glucose intake rate depending on the oxygen and glucose of the condition in which the bacterium was grown.

We assume that the genetic level is slower than the metabolic one, and therefore the steady state is reached faster than the variation of enzyme concentrations due to changes in the gene expression profile [153]. As a result, the metabolism is assumed to be always at a steady state that depends on the environmental factors. In this way, the gene expression data are used as estimator of the activity of the corresponding reaction in the model, and we are able to associate a given gene expression profile with a single point in a multidimensional and user-defined objective space. In this section, we consider the objective space of biomass-acetate flux rates, and the objective space of biomass-succinate flux rates.

Importantly, we take into account that a gene whose expression level is only slightly varied across conditions is a key gene for the organism [107]. We assume that the importance of a gene - and therefore the robustness of the reaction fluxes for which that gene is responsible - is inversely proportional to its variance across all the experimental conditions. Essential genes are in fact more tightly regulated and evolve slowly, as a high variance of gene expression level of essential genes would affect all the downstream genes and is more likely to be less tolerated in a large interaction network [66, 122]. For instance, essential genes in the metabolic network are those coding for an enzyme controlling a reaction which is upstream of many others (e.g., upstream of the TCA cycle).

To map conditions to the objective space, we solve the following bilevel problem:

$$
\begin{array}{lll}
\max & g^{\mathsf{T}} v & \\
\text{such that} & \max & f^{\mathsf{T}} v \\
 & \text{such that} & Sv = 0 \\
 & & v_i \geq v_i^L \cdot k(y_i) \\
 & & v_i \leq v_i^U \cdot k(y_i)
\end{array}
\tag{3.8}
$$

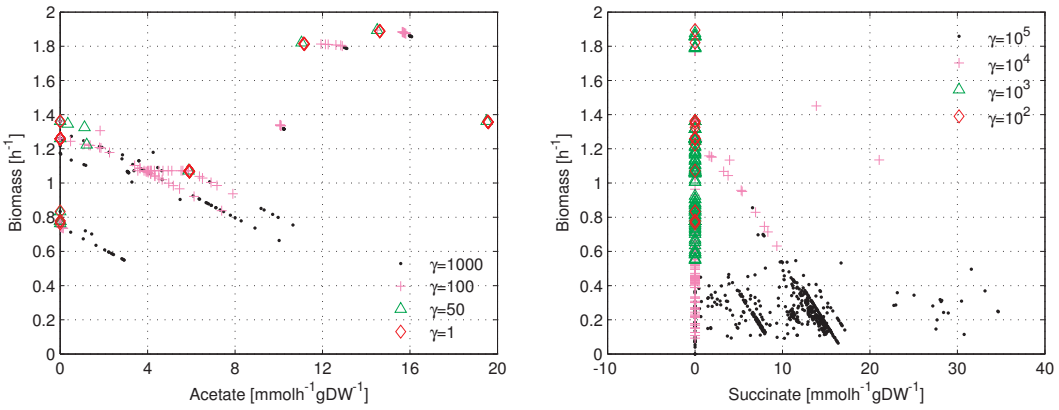
with

$$k(y_i) = \left[1 + \frac{\gamma}{\sigma_i^2} |log(y_i)|\right]^{\text{sgn}(y_i - 1)}, \tag{3.9}$$

where σ_i^2 is the variance of the gene set responsible for the ith reaction, and γ is a weight for the variance. The parameter γ represents the weight attributed to the variance as an indicator of the importance of a gene, and determines the effect of the gene expression values on the final reaction rates. The variances σ_i^2 of the gene sets are computed from the variances of the genes across the conditions in the dataset, following the same rules defined to map the gene expressions to the gene set expressions (Equations (3.1) and (3.2)).

As case-studies for this method, we choose the acetate-biomass space (Fig. 3.5a) and the succinate-biomass space (Fig. 3.5b). For increasing γ, the *E. coli* is able to move towards the production of the second objective (acetate or succinate) rather than the natural objective (biomass). By increasing the parameter γ we increase this effect, therefore even two experimental conditions with slight differences in the gene expression profiles are mapped to different points in the objective space. The map from genes to metabolism is robust with respect to perturbations of γ, while large perturbations of γ (orders of magnitude) increase the sensitivity of the metabolism to the different environmental conditions. For the succinate-biomass case, the best trade-off is reached when $\gamma = 10^4$: the best gene expression profiles are able to produce 21.06 and 13.87 mmol h^{-1} gDW^{-1} of succinate with a biomass of 1.13 and 1.45 h^{-1} respectively. Nevertheless, an excessive role attributed to gene expression as a multiplicative factor for the flux bounds (e.g., $\gamma = 10^5$ in the succinate-biomass space) leads to a reduced production of biomass (0.85 h^{-1} maximum), although providing remarkably high values of synthetic flux rates (up to 34.67 mmol h^{-1} gDW^{-1}). Conversely, for the acetate-biomass case, increasing γ improves the area of the space covered, but does not provide remarkable new regions of increased acetate and biomass yield. We also increased the order of magnitude of γ over 10^4, but we did not notice significant changes with respect to $\gamma = 10^4$.

In Fig. 3.6, we test this approach on the Colombos 2.0 compendium of microarray data [112], which includes data for 2369 measured conditions. In both the objective spaces selected, the plot identifies the subspace where the bacterium operates. Specifically, when maximizing for acetate and biomass production, *E. coli* shows greater variability and different outcomes in different conditions; the conditions are mapped to points that cover most of the objective space. Conversely, when maximizing for succinate and biomass production, only few conditions are able to ensure succinate production. Interestingly, one of the two best working conditions in the succinate-biomass space is anaerobic. The most interesting condition yields 19.66 mmol h^{-1} gDW^{-1} of acetate flux with 0.48 h^{-1} of



(a) (b)

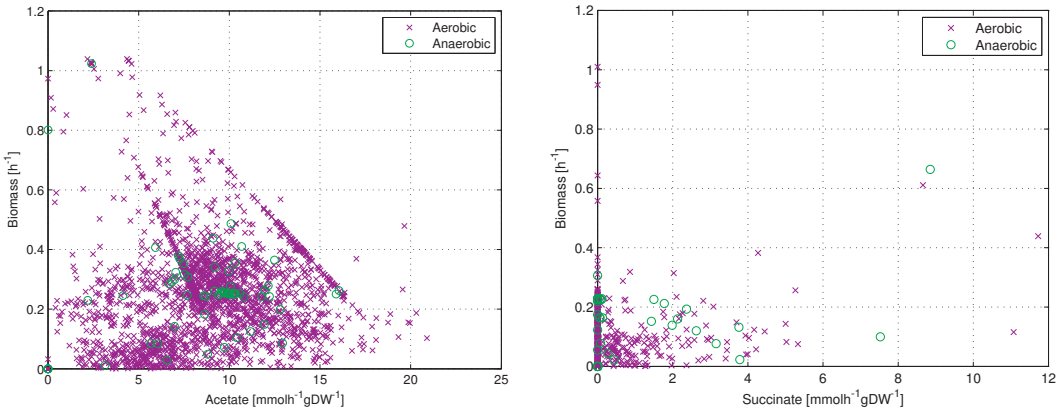
Fig. 3.5 **The 466 profiles of gene expression (each associated with one condition) positioned in the two-dimensional space acetate-biomass (a) and succinate-biomass (b).** Each point is obtained starting from a gene expression profile, converting it into a gene set expression profile y using (3.1)-(3.2), and finally applying (3.8) and considering only the fluxes associated with the objectives on the axes of the plots above. The parameter γ quantifies the effect of the gene expression values on the final reaction rates.

biomass. Only one anaerobic condition is able to ensure high biomass. In the succinate-biomass space most experimental conditions yield low value of succinate and biomass, but the best anaerobic condition gives 8.85 mmol h^{-1} gDW^{-1} of succinate flux with 0.66 h^{-1} of biomass.

3.6 The hypervolume indicator as a measure of adaptation over time

A measure for the volume of the dominated portion of the objective space is the *hypervolume indicator*. The hypervolume allows comparing different Pareto sets and evaluating the evolution of a Pareto set over time. In our two-objective spaces, since our aim is the maximization of the objectives, we choose $O = (0,0)$ as a reference point. Intuitively, the reference point determines if the area of the objective space where the algorithm performs its search is above or below the Pareto front. Let the subset $X \subset \mathbb{R}^2$ represent the Pareto front. We define the hypervolume indicator as the Lebesgue measure of the space dominated by X with respect to the reference point O, namely the space between X and O:

$$I_H(X) = \int_{\mathbb{R}^2} \mathbf{1}_{H(X,O)}(z)\,\mathrm{d}z, \tag{3.10}$$



(a) (b)

Fig. 3.6 **The** 2369 **Colombos gene expression microarray profiles mapped to a bidimensional space of objective functions acetate-biomass (a) and succinate-biomass (b).** Among the 2369 conditions (obtained with different pH, antibiotics, heat shock, glucose concentrations), 72 conditions are anaerobic. Each gene expression profile is translated into flux bounds and gives the lower and upper bound of the model using (3.9); then, the bilevel problem (3.8) is solved with acetate-biomass and succinate-biomass as objectives, thus obtaining a point in each of the two objective spaces.

where H is the set of points dominated by the Pareto front X. $\mathbf{1}_{H(X,O)}$ indicates the characteristic function and has value 1 at points of $H(X,O)$ and 0 at points of $\mathbb{R}^2 \setminus H(X,O)$. When the number of Pareto solutions is finite, i.e., $X = \{(p_1,q_1),\ldots,(p_n,q_n)\}$, the hypervolume equals

$$I_H(X) = \sum_{i=1}^{n} q_i(p_i - p_{i-1}), \tag{3.11}$$

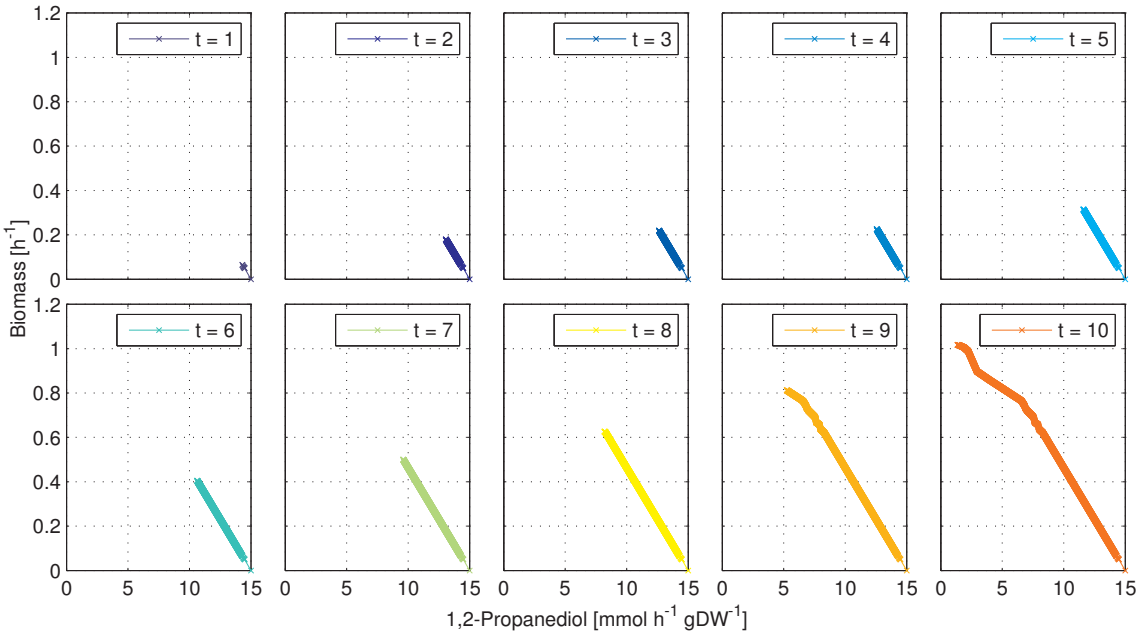
where for convenience of notation we set $p_0 = 0$. Given the set of Pareto optimal solutions, the hypervolume can be exploited in the decision-making process, e.g., through the selection of a hypervolume-maximizing subset of the Pareto-optimal set.

During the growth process of a bacterial population, the short term evolution of a single strain ranges from a wild-type configuration towards an optimized configuration where two objectives are taken into account. In order to gain insights into the evolution of an *E. coli* strain on a short temporal scale, and to provide a more accurate description of its order of growth, we analyze the dynamics of the strain on a bidimensional objective space consisting of biomass and 1,2-propanediol production (Fig. 3.7a). Different strains may evolve on large temporal scales on the same bidimensional space, starting from different initial points. Here we take into account the initial point that refers to the wild-type *E. coli*

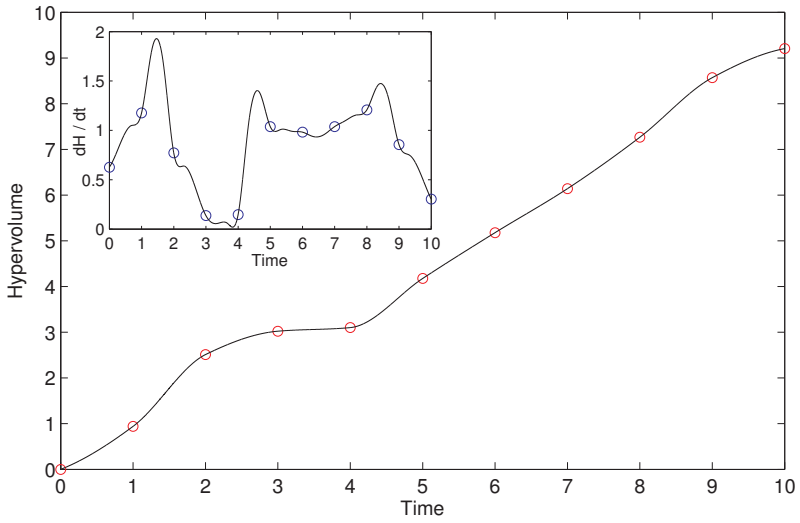
K-12 MG1655 grown on morpholinepropanesulfonic acid (MOPS) minimal medium, with anaerobic aeration, culture temperature of 37°C and 11 mM of glucose.

In order to investigate the evolving tradeoff in the objective space, we use the Pareto front as a measure for the evolution of the strain. We remark that the Pareto front shows the metabolic tradeoff between two objectives in the objective space. In METRADE, each point of the Pareto front represents the objective vector (flux rates in the objective space) corresponding to an optimal array of gene expression values (in the search space). Specifically, we quantify the evolution by computing the hypervolume indicator (and its first derivative) of ten Pareto fronts obtained during the evolutionary process of the PGA underlying METRADE. The hypervolume is an indicator for the size of the space covered by the Pareto front in a multidimensional objective space. Among its properties, it is strictly monotonic with respect to strict Pareto dominance. Therefore, the ideal Pareto front, reached only asymptotically when the number of populations generated approaches infinity, achieves the maximum hypervolume available for the system [15].

We propose the hypervolume as a proxy for the versatility of an organism and for its ability to ensure simultaneous production of multiple chemicals. An assumption of most FBA models is that a wild-type bacterium is specialized towards the production of biomass only, therefore lying on one axis of the multidimensional space, i.e., with null hypervolume. In Fig. 3.7a, the PGA adaptively finds the trade-off for the K-12 MG1655 *E. coli* that moves on the objective space towards maximization of $1,2$-propanediol production and growth rate. We captured ten snapshots of the evolution at ten equally distributed time steps (i.e., every 150 populations generated by the PGA). During the evolution towards multiple objectives, the bacterium moves and covers increasing portions of the objective space. The size of the space covered can be measured with the hypervolume indicator, while the speed of evolution can be associated with the hypervolume first derivative. In Fig. 3.7b, we plot the evolution of the hypervolume indicator over time. At each time step, we compute the hypervolume of the corresponding Pareto front using Eq. (3.11), after sorting its points (p_i, q_i) in ascending order of p. The initial growth ends after two time steps, and starts again after four time steps, decreasing at the final step. The hypervolume shows a plateau between two phases of increase. The derivative of the hypervolume indicator highlights alternating periods of slow and fast evolution that could be associated with the feast and famine growth phases of the bacterial population.



(a)



(b)

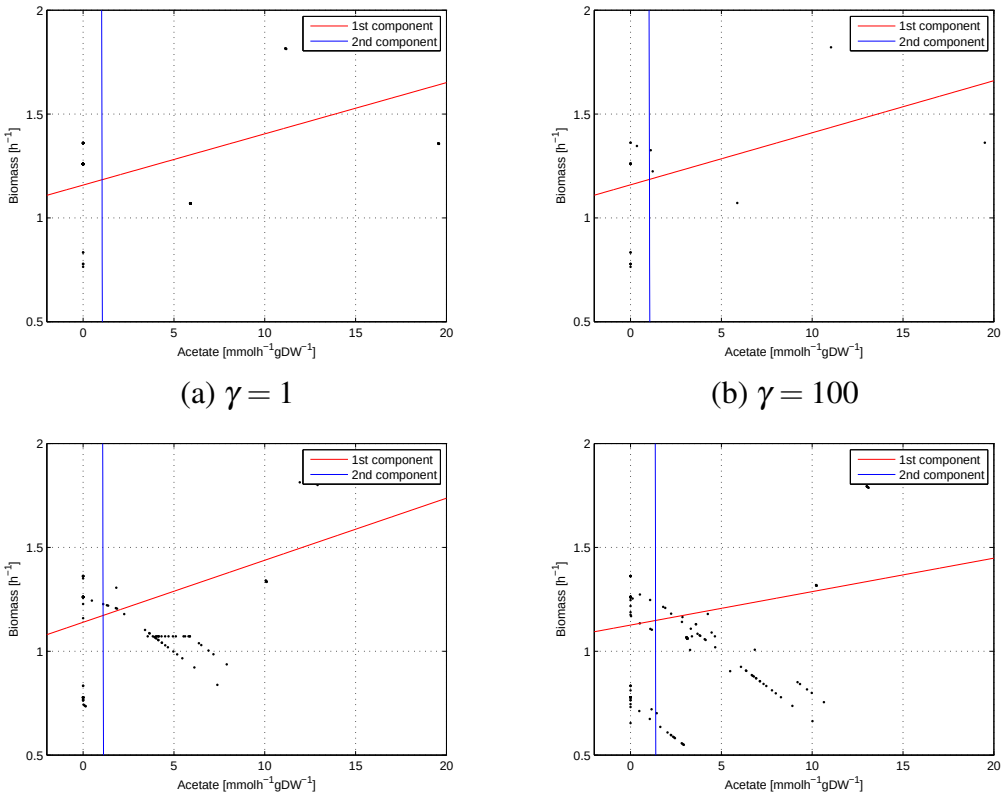
Fig. 3.7 **Temporal evolution of *E. coli* K-12 MG1655 grown on MOPS minimal medium when optimizing concurrently towards $1,2$-propanediol production and growth rate using METRADE.** (a) Evolving metabolic tradeoff towards the optimal production rates. The figure shows the progression of the Pareto front during the evolution of the algorithm, at 10 time steps. (b) Hypervolume indicator over time. We evaluate the hypervolume of the Pareto front at each of the 10 steps, using Eq. (3.11), after sorting the points (p_i, q_i) of the Pareto front in ascending order of p. The discrete time points have been interpolated with a cubic piecewise polynomial.

3.7 Principal component analysis on multi-objective spaces reveals genes-to-protein effect

In the objective spaces obtained when mapping the gene expression profiles to the target metabolites, we perform a principal component analysis (PCA) based on the singular value decomposition (SVD) of the data obtained by mapping each gene expression condition on the objective space. We obtain Fig. 3.8 starting from Fig. 3.5a, and Fig. 3.9 starting from Fig. 3.5b. This is equivalent to finding the system of axes in which the covariance matrix is diagonal. The PCA is often used to detect redundancy of information due to the fact that group of variables may vary together, and therefore can be replaced by a single variable. This is achieved through the definition of new variables as linear combinations of the original variables. The new variables, called *principal components*, are orthogonal to each other (so as to avoid redundancy) and represent an orthogonal basis. The simplification is achieved by discarding those components that explain little variance in the data, i.e., the components corresponding to the smallest eigenvalues of the covariance matrix.

We apply the PCA to the points representing the *E. coli* conditions mapped to the bidimensional objective space. The eigenvalues l_1 and l_2 of the covariance matrix indicate the variance explained by the first and second principal components respectively. The eigenvector with the largest eigenvalue l_1 represents the direction of maximal variation of the points in the objective space. Combining PCA and the map between gene expression profiles and multidimensional objective spaces allows assessing the relative impact of γ on the position of each gene expression profile in the objective space. As we also introduced in Section 3.5, the choice of this parameter merits further experimental investigation (e.g., with a parameter optimization algorithm). More specifically, our results show that the parameter γ, which represents the effect of the gene expression on the final reaction bounds of the FBA model, has a direct effect on the direction of the maximum variance, as shown in Table 3.1.

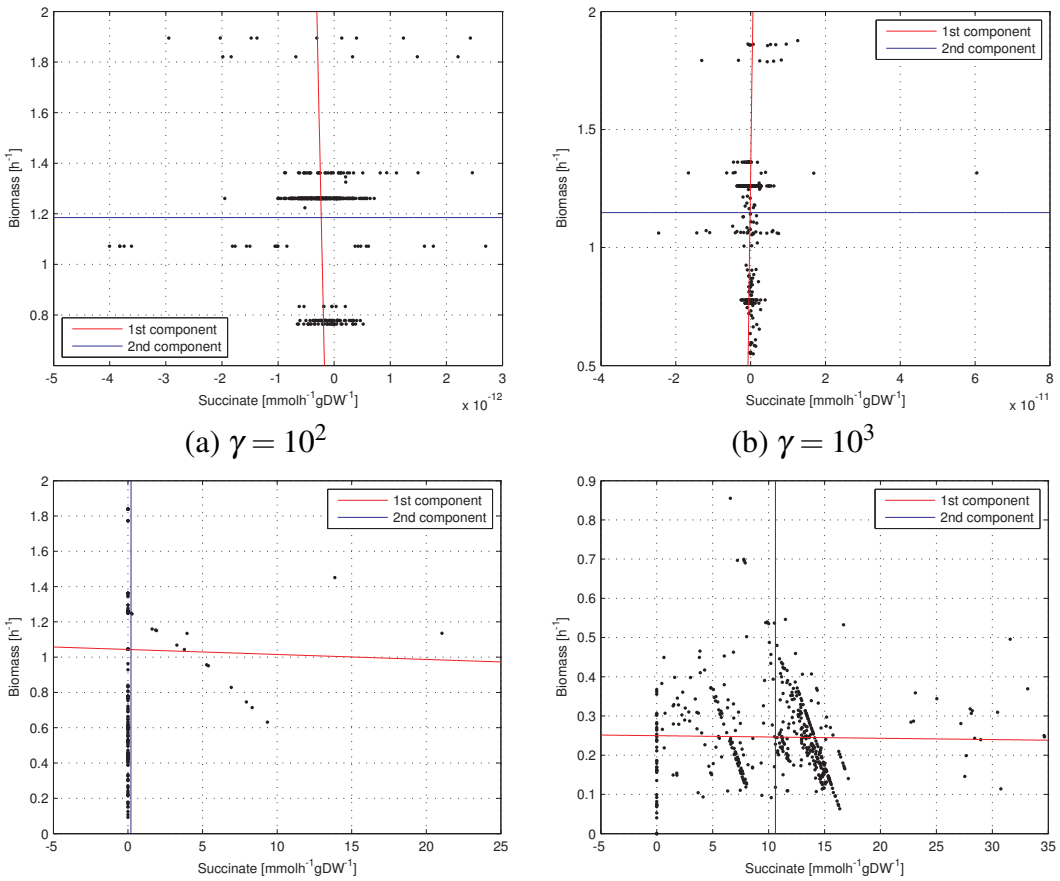
In the application shown in Figs. 3.8 and 3.9, PCA has been used to quantify the effect of the parameter γ in our method. We note that, while in two dimensions PCA can be generally avoided and in some cases replaced by visual inspection, the combination between METRADE and PCA is useful when optimizing for more than two objectives. Using PCA, the dimensionality of the phase-space of conditions can be reduced by looking at the directions of "phenotypic" maximum variance across a given set of environmental conditions.



(a) $\gamma = 1$ (b) $\gamma = 100$

(c) $\gamma = 500$ (d) $\gamma = 1000$

Fig. 3.8 **PCA in the acetate-biomass objective space.** The two principal components are computed using the principal component analysis on the centered data $(x - \mu_x, y - \mu_y)$, but are plotted on the original data. The slope of the first principal component (i.e., the direction of maximum variance of the data), depends on the parameter γ, the multiplicative factor that influences the effect of the gene expression data on the upper and lower bounds of the reaction fluxes in the FBA model. The second principal component is always perpendicular to the first component (not highlighted in these plots due to different scales used).



(a) $\gamma = 10^2$ (b) $\gamma = 10^3$

(c) $\gamma = 10^4$ (d) $\gamma = 10^5$

Fig. 3.9 **PCA in the succinate-biomass objective space.** The plots have been obtained with the method described in Fig. 3.8. With respect to the first two cases, the direction of the first principal component in the last two cases is remarkably different.

γ	pc_1	pc_2	l_1	l_2
1	$(0.9997, 0.0247)$	$(0.0247, -0.9997)$	12.9026	0.0460
100	$(0.9997, 0.0251)$	$(0.0251, -0.9997)$	12.7476	0.0463
500	$(0.9996, 0.0299)$	$(0.0299, -0.9996)$	9.5999	0.0462
1000	$(0.9999, 0.0161)$	$(0.0161, -0.9999)$	11.2712	0.0597

γ	pc_1	pc_2	l_1	l_2
10^2	$(9.7072 \cdot 10^{-14}, -1)$	$(-1, -9.7072 \cdot 10^{-14})$	0.0542	$4.1806 \cdot 10^{-25}$
10^3	$(-8.6034 \cdot 10^{-13}, 1)$	$(-1, -8.6034 \cdot 10^{-13})$	0.0626	$1.5392 \cdot 10^{-23}$
10^4	$(1, -2.8217 \cdot 10^{-3})$	$(-2.8217 \cdot 10^{-3}, -1)$	2.1323	0.1267
10^5	$(-1, -3.2925 \cdot 10^{-4})$	$(3.2925 \cdot 10^{-4}, -1)$	37.6676	0.0121

Table 3.1 Values of γ, the principal component coefficients pc_1 and pc_2 (expressed as pair (a,b) of the line $ax+by=0$), and the principal component variances l_1 and l_2 (i.e., the eigenvalues of the covariance matrix) of the gene expression data in the acetate-biomass objective space (top) and succinate-biomass objective space (bottom). The eigenvalues l_1 and l_2 indicate the variance explained by the first and second principal components respectively. The eigenvector with the largest eigenvalue l_1 represents the direction of maximal variation of the points in the objective space.

3.8 Approaches towards experimental validation of METRADE predictions

In silico predictions of flux distributions may be validated *in vivo* through biological experiments aimed at measuring the flux rates of a set of biochemical reactions. The most common approach for measuring flux rates is called metabolic flux analysis (MFA). MFA uses carbon ^{13}C-labeled substrates to quantify intracellular fluxes in the central metabolism only [180], although an attempt to extend MFA to genome-scale models has recently been published and assessed [70]. An approach that uses ^{13}C labeling data to constrain metabolic models has been proposed by Martín et al. [110]. MFA is used in the study of microbial physiology, but currently not in mammalian cells and plants due to more complex metabolic networks and growth media where applying MFA is more challenging. Furthermore, carrying out a carbon-labeling experiment is not simple and requires a number of experimental steps [31].

An easier way to partially validate FBA-based flux distributions is the measurement of the growth rate, or the rate of production of a specific compound in a cell culture. The experimental values are then compared with the predicted values to check the correctness of the FBA-based method used. At this stage, wrong predictions are not necessarily due to

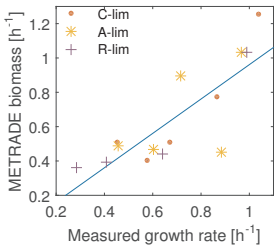
the method, and may also lead to discovery of gaps in the metabolic model (e.g., missing genes, missing reactions, or incomplete gene-protein-reaction associations).

In this section we compare the results obtained by our method with a publicly available set of experiments performed on *Escherichia coli*, where the growth rate was measured, and with the experimental results on *Corynebacterium glutamicum* and *Saccharomyces cerevisiae* (yeast) provided by Dr J. Dias, Dr D. Dikicioglu and Prof S. Oliver, from the Department of Biochemistry, University of Cambridge, UK. For *C. glutamicum*, we are able to compare our predictions with the measured value of flux rate for succinate production. For *S. cerevisiae*, we apply METRADE on four microarray profiles mapped to the ATP-biomass and D-glucose-glycerol objective spaces, and we obtain insights into the flux distributions, which were subsequently confirmed by our collaborators. We would like to acknowledge them for the permission to include in this dissertation some of the results we obtained starting from their experimental data on *S. cerevisiae* and *C. glutamicum*. The collaboration is still ongoing and we plan to obtain and validate further predictions of metabolic flux rates in yeast and *E. coli*.

3.8.1 Case study I: *Escherichia coli*

To validate METRADE on *E. coli*, we use the phenomics dataset by Hui et al. [78]. The compendium contains 14 expression profiles in different growth conditions, with measured growth rates between 0.28 and 1.04 h^{-1}. The different conditions were obtained with: (i) titrated catabolic flux through controlled inducible expression of the lacY gene; (ii) titrated anabolic flux through controlled expression of GOGAT; (iii) inhibition of protein synthesis with an antibiotic (chloramphenicol). Overall, we obtain remarkable results in predicting the growth rate from the expression profile associated with each condition. On the full dataset, we obtain a strong correlation between predicted and measured growth rates (Pearson's $r = 0.81$, p-value $= 4.38 \cdot 10^{-4}$, and Spearman's $\rho = 0.78$, p-value $= 9.21 \cdot 10^{-4}$). The best results are obtained with the subset of conditions representing inhibited protein synthesis by supplying chloramphenicol to the growth medium (Pearson's $r = 0.92$, Spearman's $\rho = 1$). A comparison between predicted and measured growth rates is reported in Figure 3.10.

Changing flux rates *in vitro* or performing gene overexpression and partial knockdown suggested by METRADE is less straightforward than performing gene knockout. However, the number of techniques to perform gene expression changes in bacteria is rapidly increasing. The methods available to date have been recently reviewed by Yen et al. [176]. Our predictions on overexpression or partial gene knockdown can be implemented using plasmids or through promoter engineering based on CRISPR-Cas, homologous recombination or transposable elements, RNA programming devices [34], and engineering of



Index	Condition type	Growth limitation	Pearson r	Spearman ρ
1-5	C-lim	catabolic	0.90	0.70
6-10	A-lim	anabolic	0.55	0.30
11-14	R-lim	polymerization	0.92	1
1-14	Full dataset	(variable)	**0.81** ($4.38 \cdot 10^{-4}$)	**0.78** ($9.21 \cdot 10^{-4}$)

Fig. 3.10 **METRADE predictions and measured growth rates in a phenomics dataset.** (Left) METRADE predictions and measured growth rates for each subset of the dataset used in this study. The subsets of conditions are denoted by (i) *C-lim*: titrated catabolic flux through controlled inducible expression of the lacY gene; (ii) *A-lim*: titrated anabolic flux through controlled expression of GOGAT; (iii) *R-lim*: inhibition of protein synthesis with an antibiotic (chloramphenicol). (Right) Spearman's ρ and Pearson's r correlation coefficients between METRADE and experiments. The p-value is reported in brackets. While obtaining a good overall correlation (final row of the table) between experimentally-measured growth rates and the biomass predicted by METRADE, we obtain the best results in the conditions representing inhibited protein synthesis by supplying chloramphenicol to the growth medium.

ligands to create sensors for regulation of gene expression [173]. Modulation of gene (or protein) expression level can be achieved through engineering of ribosomal binding sites [59, 129, 140] and promoters [3, 75, 170].

3.8.2 Case study II: *Saccharomyces cerevisiae*

As a first step, we test the multi-objective optimization in METRADE on a model of *Saccharomyces cerevisiae S288c*. We first optimize the ATP synthase flux rate and concurrently maximize the biomass (Fig. 3.11). The model is constrained using experimentally-derived upper bounds obtained by measuring uptake rates of 15 metabolites. Then, we perform a variability analysis combined with the multi-objective optimization analysis. Specifically, we focus on 149 fluxes provided by our collaborators and we perform a flux variability analysis across the front by computing their flux rates for each solution on the Pareto front. Across the Pareto front, we correctly predict changes in the acidity of the system. Indeed, carbon dioxide (CO_2) and carbonic acid (H_2CO_3) are kept in balance, and the flux variability shows that their flux vary considerably along the front. These predictions were experimentally confirmed by our collaborators. Another interesting result is predicted among the amino-acids, where alanine showed a large variability across the front. Depending on the region of the Pareto front, alanine is imported or released, therefore fulfilling both catabolic and anabolic functions.

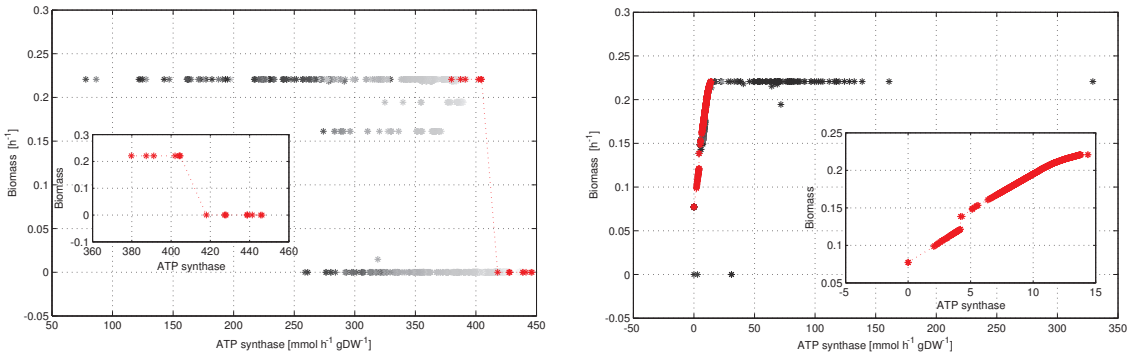


Fig. 3.11 **Multi-objective optimization of *S. cerevisiae* in the configuration that minimizes (a) and minimizes (b) ATP while maximizing the growth, applying constraints derived from experimental uptake rates of** 15 **metabolites.** We show all the solutions found by our algorithm: red points are non-dominated, black points are dominated. Each point is a particular set of values for the flux rates in the model.

As a second step, we use METRADE to integrate transcriptomics into the yeast metabolism. We consider a set of four experimental conditions (glucose- and ammonium-limited cultures of two different deletion mutants, HO and QDR3). Both genes are not included in the metabolic reconstruction, and their effect on the metabolism is still a matter of debate. QDR3 is an efflux pump that acts as a hydrogen anti-porter. The analysis on the QDR3 deletion would therefore give insights into the role played by QDR3 in the yeast metabolism. We take the HO deletion as a control condition. As for microarray data, we use the four microarray profiles corresponding to the four experimental conditions, and we use METRADE to map them to the glucose-glycerol and ATPsynthase-biomass objective spaces (Fig. 3.12), and finally to evaluate the resulting flux distributions. Remarkably, the insights provided by METRADE were used by our collaborators to reduce the number of orphan reactions, i.e., those that are not associated with any gene. While 33.9% of the reactions in the complete network are orphan reactions, at least 21.6% of their fluxes show a variation in one of the four conditions. We obtain the best result (27.1%) when minimizing the glucose uptake and maximizing the glycerol production.

3.8.3 Case study III: *Corynebacterium glutamicum*

Corynebacterium glutamicum is a gram-positive bacterium used to industrially produce amino acids such as L-glutamate and L-lysine. Amino acids are important industrial products with a market of three million tons per year ranging from pharmaceutical products to food and beverage [19]. Here, we consider a model of 518 reactions, 399 metabolites,

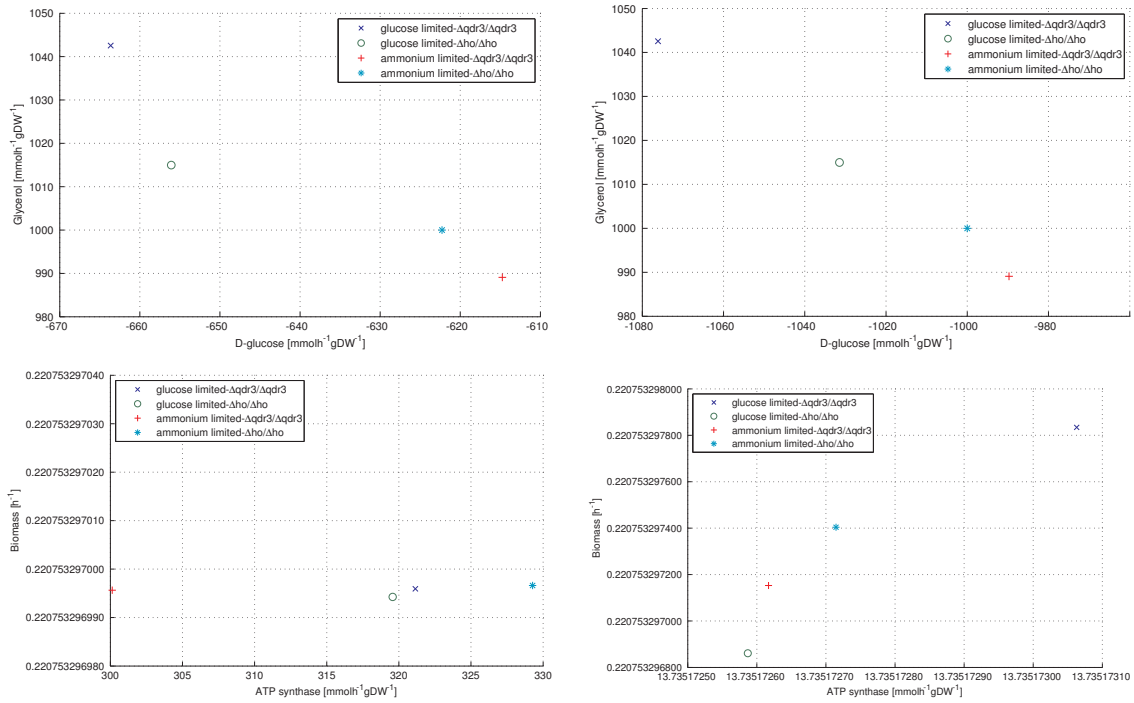
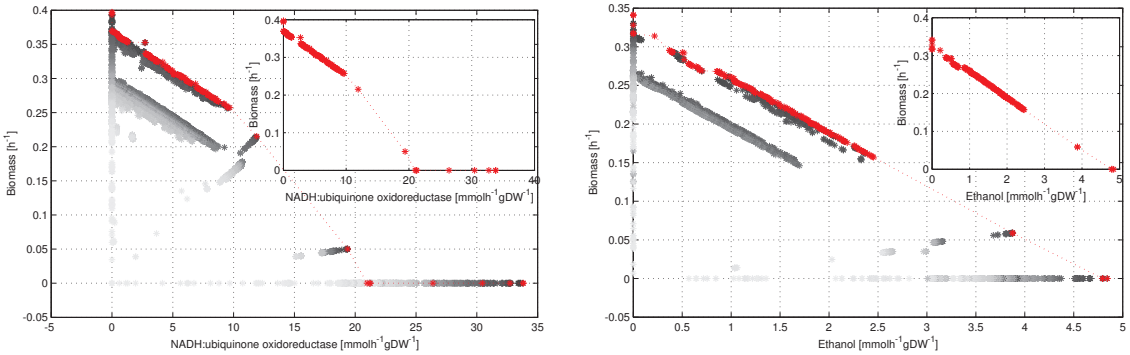


Fig. 3.12 **Mapping of four experimental conditions onto the objective space of glucose-glycerol and ATPsynthase-biomass.** Two different deletion mutants, HO and QDR3, were grown in glucose- and ammonium-limited cultures. For each condition, we use METRADE to constrain the model starting from the microarray data obtained in each condition, and we use bilevel linear programming to optimize towards maximum glucose uptake and maximum glycerol (a), minimum glucose uptake and maximum glycerol (b), maximum ATP synthase and maximum biomass (c), minimum ATP synthase and maximum biomass (d).



(a) (b)

Fig. 3.13 **Multi-objective optimization of NADH:ubiquinone oxidoreductase, ethanol production and biomass for the *C. glutamicum*.** The Pareto front is also a powerful tool for comparison of metabolic capability. In (a), the shape of the Pareto shows that, when the bacterium is pushed towards increasing the flux of NADH dehydrogenase while ensuring a high growth rate, only a few areas of the objective space are viable. Furthermore, the relation between the two objectives is not linear, unlike when overproducing ethanol and biomass (b).

and 468 genes [165]. To constrain the model, we use experimental flux measurements for eleven exchange reactions: L-malate, fumarate, L-alanine, L-valine, L-Glutamate, acetate, lactate, pyruvate, formate, glucose, CO_2.

In a first setting, we optimize biomass as a first objective to maximize, NADH production as a second objective to minimize, and the difference between our estimated succinate production and the succinate production measured experimentally as a third objective to minimize. The FBA controls NADH and biomass, while the PGA underlying METRADE controls the biomass and the third objective. This approach is also called *goal programming*, since it consists of defining an objective function as the absolute deviations from the targets to the objectives [41]. We test this approach in three different experimental conditions. For the first experimental condition, the best non-dominated point of METRADE consists of 1.6863 mmol h^{-1} gDW^{-1} of succinate against 1.5272 mmol h^{-1} gDW^{-1} measured in the experiment. For the second experiment, 0.6595 mmol h^{-1} gDW^{-1} against 0.5880 mmol h^{-1} gDW^{-1}. For the third experiment, we predict 0.4297 mmol h^{-1} gDW^{-1} of succinate flux, against 0.4074 mmol h^{-1} gDW^{-1} found in the experiments. We note that in all the three experiments our predictions are remarkably close to the experimentally-measured exchange flux. With the same technique, we maximize the NADH:ubiquinone oxidoreductase, ethanol and biomass (Fig. 3.13).

To further test METRADE, we compare the *in silico* results, obtained through multi-objective optimization, with *in vitro* results, obtained for succinate production in specific

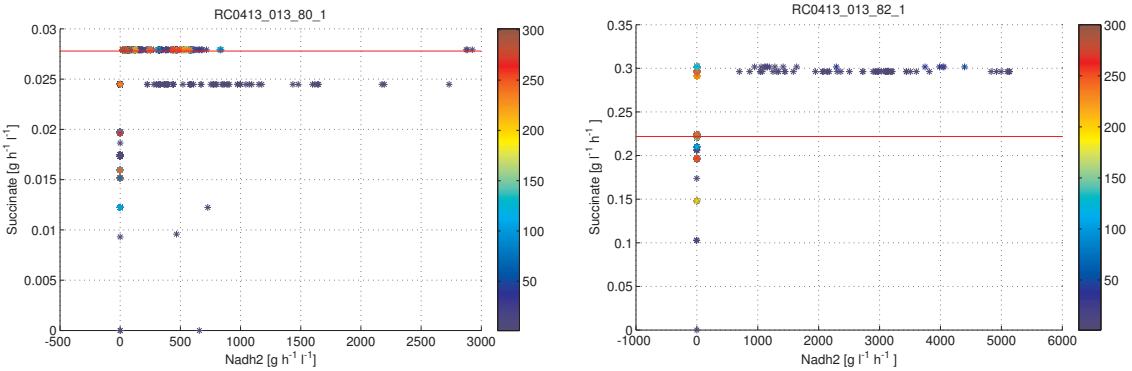


Fig. 3.14 **Comparison between the points obtained by METRADE and the experimental value (red line) of succinate in two different strains of *Corynebacterium glutamicum* RC0413_013_80_1 and RC0413_013_82_1**. The reaction named "Nadh2" is a reversible reaction modeling the production/intake of NADH, with formula nad[c] + h[c] <=> nadh[c]. Solutions found by METRADE are denoted by progressively warmer colors according to the population in which they have been generated by the PGA.

strains (RC0413_013_80_1 and RC0413_013_82_1). In two different experiments, we consider the strains able to maximize the production of succinate. As shown in Fig. 3.14, in both cases there is a very good agreement between the experiments and our results, with the Pareto front always automatically recovering points whose succinate is the succinate measured experimentally, and in some cases also suggesting metabolic configurations that would allow increased production of succinate with minimum NADH production.

3.9 Discussion

In this chapter we proposed METRADE, a multi-omic extension of FBA and GDMO able to map environmental conditions to the variation of gene expression. We then investigated the effect of different conditions on the phenotype (through the metabolic map). Our method associates each gene expression profile with a flux distribution represented by a point in a multidimensional objective space, which therefore becomes a condition phase-space. Finally, we focused on the condition phase-space and we proposed a set of methods to analyze the adaptability to experimental conditions. The principal component analysis allowed exploring the directions with largest variance, indicating the relation between the gene expression profiles and the variance in the two or more objectives chosen. The hypervolume indicator used within METRADE focused on how the trade-off evolves in a bidimensional objective space, highlighting periods of slow and fast evolution.

The importance of mapping microarray profiles to a metabolic model before further analyzing (e.g. clustering) their associated conditions, compared to the analysis performed directly on gene expression data, is as follows. First, multi-omic models provide means for clustering genes in their relative pathway; as a result, pathways can be effectively clustered and ranked through an effect-based approach (i.e., looking at the distance between their output outcomes in the phenotypic space) rather than looking merely at the expression profile of their genes. Second, the model acts as a ranking and noise-reduction tool, since the effect of low-importance genes is filtered out even if their expression is highly variable across conditions; without the multi-omic model, these genes would be incorrectly regarded as key genes to differentiate conditions. Third, performing inference directly on gene expression values may lead to incorrect prediction of the centrality of a gene whose level seems to be highly correlated with many other genes, but with only a marginal role in the metabolism (e.g., no impact on the biomass and on key metabolites).

The use of gene expression and codon usage as layers of the multi-omic model allows simulating growth on different media or environmental conditions. On the integration of gene expression data in the metabolic model, we note that related methods (e.g., GIMME [20] and iMAT [184]) need binarization or discretization of expression values. Conversely, we do not need to set any arbitrary threshold, and therefore we do not need to assume that only some reactions are present in the model. A further limitation of these methods is that considering only one objective or a linear combination of objectives (usually encoded in the biomass reaction) is not sufficient to find all the possible metabolic states in which the maximization of a given product is performed with the maximization of the biomass on various substrates. In this regard, the use of a multi-objective algorithm is key to obtain all the possible trade-off relationships among objectives, thus overcoming one of the major limitations of bilevel FBA, namely the estimation of only one point solution in the bidimensional objective space. With our multi-objective approach, we provide a wider range of solutions ensuring optimal values for all the objectives.

METRADE is freely available as an extension of the COBRA 2.0 toolbox, and can be easily integrated with the methods we used in Chapter 2 to investigate the Pareto front: (i) the *sensitivity analysis*, to quantify the importance of parameters and variables in the model; (ii) the *robustness analysis*, to evaluate how a proposed strain is robust to local and global perturbations [44]; (iii) the *identifiability analysis*, to find functional relations between variables [8]; and (iv) the ε*-dominance analysis*, to consider all the feasible solutions that are dominated with a tolerance ε with respect to Pareto solutions.

Finally, METRADE can be used to detect "internal" communities of conditions, where the closeness is measured on the response of the multi-omic model; it is therefore possible to

create a correspondence with the "external" communities of conditions, where the closeness is usually knowledge-based or measured directly on the features of the environmental conditions. Due to the continuous nature of METRADE, we expect it will be used for calibration of genome-scale models, and in combination with dynamical aspects of FBA with the aim of detecting communities of conditions over time (e.g., reiterated shifts to completely different external conditions or growth environments). Coupled with advanced prediction techniques, for instance Bayesian "missing values" methods, METRADE can infer the response to conditions for which gene expression data are missing or incomplete. Therefore, it could represent an innovative tool for biologists for investigating important aspects of bacterial evolution, such as: (i) how genomic, metabolic, transcriptomic variations shape the complex adaptation landscape of bacteria; (ii) the ecological (condition-based) degree of coherency for bacterial genospecies; (iii) the relationship between speciation, ecotypes and ecological (condition-based) diversity; (iv) the adaptive response to different dosages of antibiotics and bacteriostatic chemicals.

3.10 Related work and final remarks

The work presented in this chapter has been accepted for publication in Nature Scientific Reports [11]. A spectral method for community detection, the pseudospectrum and its bagplot were used to further analyze the experimental conditions and to shed light on the structure of a distance matrix built on the space of environmental conditions. The algorithm for community detection of conditions aims at clustering conditions according to their similarity in the objective space. The pseudospectrum is instead aimed at gaining insights into the distribution of the position of each condition in the objective space in presence of uncertainty due to external perturbations.

Part of the method for integration of omic data in the metabolic model, coupled with a Bayesian method for inferring pathway cross-correlations and pathway activation profiles, has been published in ACS Synthetic Biology [12]. I performed all the analyses described in these papers and included in the dissertation. The Bayesian analysis, not included in the dissertation, was contributed by N. Pratanwanich.

Chapter 4

Metabolic models as biological computers

Here we introduce a relation between computation and metabolism explained through a formalism that associates the structure of any bacterium with a von Neumann architecture. First, we discuss this mapping by viewing the metabolism as a Minsky register machine with universal computing capability. Then, we discuss the effect that various events (e.g., motility, communication, gene duplication) may have on the computation performed by a bacterium. Finally, we highlight the changes occurring in the computation capability as a consequence of a duplication event followed by a mutation.

The aim of this chapter is to propose a way to interpret the metabolism as a Turing machine and, in general, the full cell as a von Neumann architecture. This also leads to estimate the "computational power" of a set of reactions, which can be defined as the number of final states that they can reach, and depends on the reactions themselves and on various constraints that must hold. Metabolic engineering (for example strategies proposed by GDMO and METRADE in the previous chapters) is a way to program these machines and drive them towards a desired output.

4.1 Can biological processes compute?

If Turing were a first-year graduate student interested in computers, he would probably migrate into the field of computational biology. In 1952, he outlined computational processes in morphogenesis [162], thus thinking of the biological development of an organism as a consequence of the computation that it can perform. Following Turing's idea on morphogenesis, many biological processes have been recently analyzed from a computational

standpoint. For instance, in 1995, Bray argued that a single protein is a computational or information-carrying element, being able to convert input signals into an output signal [26]. Turing's idea is that the computation carried out by an organism allows it to move from one development pattern into another. For instance, multi-cellular organisms can be thought of as the product of the computation started from a single cell, which is capable of running a program like that of a computer [28]. This leads us to speculate that the instructions of the code executed in an organism are responsible for driving its behavior and evolution.

Evolution had already been associated with computation many years before, by von Neumann and Burks [168], who constructed a self-replicating cellular automaton with the aim of developing synthetic models of a living organism. Recently, the theory of self-reproducing machines has been thoroughly analyzed by means of text register machines [116] or self-modifying register machines [109]. Several programming languages have been developed and analyzed for modeling biological computational processes. A computational process has been discovered also in ciliates during the unscrambling of genes [97], showing many points in common with the Adleman's algorithm [1], where a DNA computer is able to solve the instances of the NP-complete Hamiltonian path problem. Therefore, a computation having a *micronuclear sequence* as input can yield a functional *macronuclear gene*, meaning that a guided genome recombination system can simulate a Turing machine [4]. In a molecular machine, there is increasing evidence that the DNA is the part of the cell simulating a memory storage [150].

Here we propose a relation between computation and metabolism explained through an effective formalism. In particular, we associate the structure of a bacterium with a von Neumann architecture, showing that the components of a bacterium can be mapped to a processing unit, a control unit, a memory storing the "program" of the bacterium, and an input-output section. The genome sequence is thought of as an executable code specified by a set of commands in a sort of ad-hoc low-level programming language. In this way, the bacterium becomes a molecular machine with computation capability. Furthermore, the set of all its chemical reactions represents a processing unit, and we show that the entire metabolic network works as a Turing Machine [5].

Using the metabolism to actually perform computation means that one needs to be able to explicitly track the exact number of each molecular species (as in the Gillespie algorithm [68]). This is currently possible for extremely small systems with a limited number of reactions. *In vitro*, "single molecule tracking", i.e. a set of methods for tracking single molecules in living cells, is still in its infancy [95].

The methodology proposed in this chapter aims at addressing the challenge pointed out by Weiss [172], i.e., to ensure robust computation in cells, with reliable and reproducible

results. This would lead to effectively modify and harness biological computers for our purposes. Running a program in a molecular machine can represent an effective intervention in a cell, driving it towards the modification of its behavior according to the available inputs and the desired outputs. Although modeling the whole life of an organism as a Turing machine would certainly be computationally unfeasible, our approach is aimed at explaining the single operation executed by a bacterium in light of a computational instruction. Interestingly, our approach can be readily used to evaluate the computational effort for a specific task, or the computational capability of the whole organism under investigation.

4.2 Abstract models of computation

4.2.1 Turing machine

A Turing machine (TM) is a mathematical model describing a read/write head manipulating symbols on a tape. This model, proposed by Alan Turing in 1936 [163], is widely recognized as the first model of a "general purpose" computer. Formally, a TM is defined as a 7-uple

$$\mathscr{T} = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r), \tag{4.1}$$

where: Q is a finite set of states of the machine; Σ is a finite input alphabet and does not contain the blank symbol; $\Gamma \supset \Sigma$ is a finite tape alphabet and contains the blank symbol. The initial input received by $\mathscr{T}$ is a set of symbols in Σ (and therefore also in Γ) written in the first n locations of the tape. Since Σ does not contain the blank symbol, the first blank symbol on the tape indicates the end of the input. The head starts at the leftmost location of the tape. The function

$$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\} \tag{4.2}$$

is the transition function controlling the steps of the machine: when $\mathscr{T}$ is in a given state $q \in Q$ and its head is in a location containing $a \in \Gamma$, if $\delta(q,a) = (r,b,R)$, the machine replaces a with b, leaves state q, enters state r, and moves right (or moves left if $\delta(q,a) = (r,b,L)$). Note that if the machine is in the leftmost (or rightmost) position of the tape, and δ indicates to move left (or right, respectively), the machine stays in place. The states $q_0, q_a, q_r \in Q$, with $q_a \neq q_r$, are three distinguished states indicating respectively the start state, the accept state, and the reject state. The machine halts when it reaches q_a or q_r.

Intuitively, a TM is a finite state machine working on a tape. The tape is divided into discrete locations, where each location can contain one symbol from a finite alphabet.

Given its current state and the symbol found in the current location, the TM writes a new symbol in that location and moves in a new location. The TM is now in a new state. Note that the new location can be the same as the old location, i.e., the machine stays in place; analogously, the new symbol written in a location can be the same symbol that was already in that location. The machine has a specified initial state, and may also decide to halt. Given the state of the machine and the symbol that the machine is currently reading, the operations performed by the machine are described in a program specifying the actions to perform. Therefore, a TM provides a formal definition of computability in the discrete domain. At a given time, the *configuration* of the TM is given by the state of the tape, the state of the head and its position on the tape.

4.2.2 Minsky register machine

A counter machine, also called Minsky register machine (RM), is also an abstract model of computation. The formal definition of Minsky machine

$$\mathscr{M} = (D, H, i_0, i_1, \varphi) \tag{4.3}$$

includes a finite set D of states, a finite set $H = \{H_r\}_r$ of registers, and a multivalued mapping that governs the transition from one state to another:

$$\varphi : D \backslash \{i_0\} \longrightarrow \{(H_r, i), (H_r, j, k) \mid H_r \in H,\ i, j, k \in D\}. \tag{4.4}$$

The initial and the halting states are two distinguished elements $i_1, i_0 \in D$. The RM executes two basic increment/decrement instructions: (i) $inc(i, r, j)$ to increment register r by 1 and move from state i to state j according to $\varphi(i) = j$; (ii) $dec(i, r, j, k)$, with $H_r > 0$, to decrement register r by 1 and move from state i to state j ($\varphi(i) = j$); if $H_r = 0$, the machine moves from state i to state k ($\varphi(i) = k$). The RM also has a *halt* instruction that halts its operation, setting the state i_0.

A two-register Minsky RM has been proven to be equivalent to a TM [113]. Intuitively, a RM is a multi-tape TM with the tapes restricted to act like simple registers (i.e., "counters"). A register is represented by a left-handed tape that can hold only positive integers by writing stacks of marks on the tape; a blank tape represents the count '0'. The RM registers are monosymbolic and not bounded, i.e., they are stacks containing the same symbol repeated. Each register can be either incremented or decremented (if it is nonzero, otherwise the instruction is ignored and the machine proceeds to the next instruction).

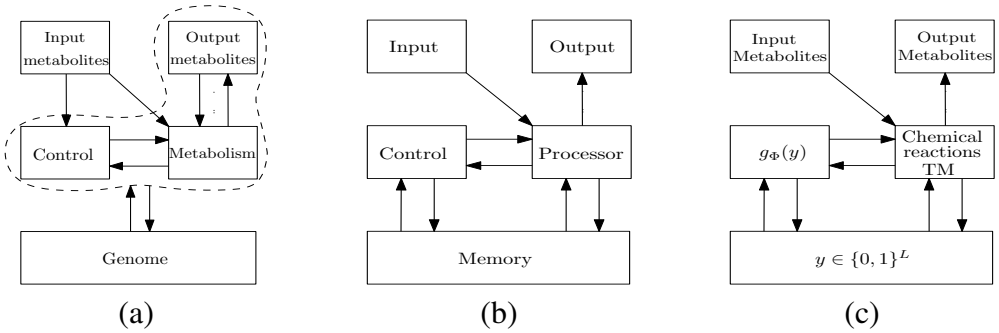


Fig. 4.1 **Comparison among biological systems (a), von Neumann architecture (b), and bacteria (c).** The string y in (c) is a program stored in the RAM and written in an ad-hoc low level programming language. The map g_Φ represents the control unit: it interprets the binary string y and turns gene sets off. The processing unit is the metabolism of the bacterium, composed of all the chemical reactions that take place in it. The goal is to produce desired metabolites as output of the molecular machine.

4.3 Bacteria as von Neumann architectures

It is well known that a von Neumann architecture is composed of a processing unit, a control unit, a memory to store both data and instructions, and input-output mechanisms. Similarly, the bacterium takes as input the chemicals (substrates) necessary for its growth and duplication, and uses its biochemical network (coded by the genes of its genome) to produce metabolites as output [5]. Inspired by Brent and Bruck [27], who studied similarities and differences between biological systems and von Neumann computers, in Fig. 4.1 we propose a mapping between the von Neumann architecture and bacteria. This mapping leads to thinking of the metabolism as a Turing machine, and of the knockout string y obtained by GDMO as a "program" that drives the machine.

Let us consider the multiset Y of the bits of y. A partition Π of the multiset $Y = \{y_1, y_2, ..., y_L\}$ is a collection $\{b_1, b_2, ..., b_p\}$ of submultisets of Y that are nonempty, disjoint, and whose union equals Y. The elements $\{b_s\}_{s=1,...,p}$ of a partition are called blocks. We denote by $P(Y;p)$ the set of all partitions of Y with p blocks. $P(Y;p)$ has a cardinality equal to the Stirling number, namely $|P(Y;p)| = S_{L,p}$ [155]. In order to formalize the behavior of the control unit, let us define the function:

$$g_\Phi : \{0,1\}^L \longrightarrow \bigcup_{y \in \{0,1\}^L} P(Y;p) \tag{4.5}$$

$$\bar{y} \in \{0,1\}^L \longmapsto \Pi \in P(\bar{Y};p), \tag{4.6}$$

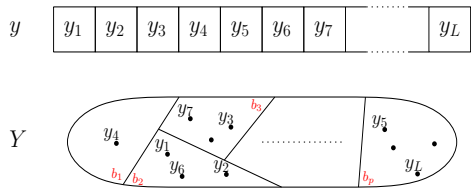


Fig. 4.2 **The multiset Y associated with y is partitioned by Π in p blocks.** The elements of Π are submultisets of Y, since y is a string of bits, thus 0 and 1 may occur more than once in the same subset. In this example, $\Pi = \{\{y_4\},\{y_1,y_6,y_2\},\ldots,\{y_5,...,y_L\}\}$, $\Phi = \{\{4\},\{1,6,2\},\ldots,\{5,...,L\}\}$.

where the partition Π is uniquely determined by the pathway-based subdivision of the chemical reaction network that we find in metabolic models. This subdivision, in turn, can be formalized as a p-blocks partition Φ of the set of the bit indices in the string y. In particular, if we denote by $[L]$ the set of the first L natural numbers, we have $\Phi \in P([L];p)$. According to the partition Φ, the control function g_Φ partitions the multiset Y associated with the string y (see Fig. 4.2).

The function g_Φ plays the role of the control unit, since it interprets the binary string y and turns gene sets on and off, according to the pathway-based partitioning of the reactions occurring in the bacterium. Each element of the partition Π is the submultiset b_s of all the gene sets that play a role in the reactions belonging to the sth pathway. In other words, g_Φ turns syntax into biological semantics. The processing unit of the bacterium could be modeled as the collection of all its chemical reactions. In this regard, a TM can be associated with the chemical reaction network of a bacterium [152]. The useful metaphor to frame a biological systems as a von Neumann computer hinges on the representation of the metabolism as a TM [5], as summarized in Table 4.1.

Biological organism	von Neumann	Bacterium
input metabolites	input	input metabolites
output metabolites	output	output metabolites
control	control	pathway handling g_Φ
genome	memory	knockout string
metabolism	processor	TM

Table 4.1 **Bacteria as von Neumann architectures.** Dictionary translating the general biological organism (left) into the computational concept of the von Neumann architecture (center) and the equivalent in a bacterium, according to our framework (right).

The sections of the von Neumann architecture mapped to an evolving bacterium are shown in Fig. 4.3. The substrates required for the growth of the bacterium are represented by the input module of the architecture. The memory module contains the knockout string y obtained by GDMO, which plays the role of the "program" responsible for protein

production in the cell. The chemical reaction network reads and executes the program y, converting input metabolites into output products [5]. The control unit g activates or deactivates gene sets translating the binary string y (syntax) into knockout instructions performed in the metabolic network (semantics). The network of the bacterial metabolism and the availability of nutrients drive this process. The evolution is represented as an increased capability of biological computation (e.g. ability of producing new chemicals) due to an enlarged genome. We will analyze this scenario in Section 4.5.

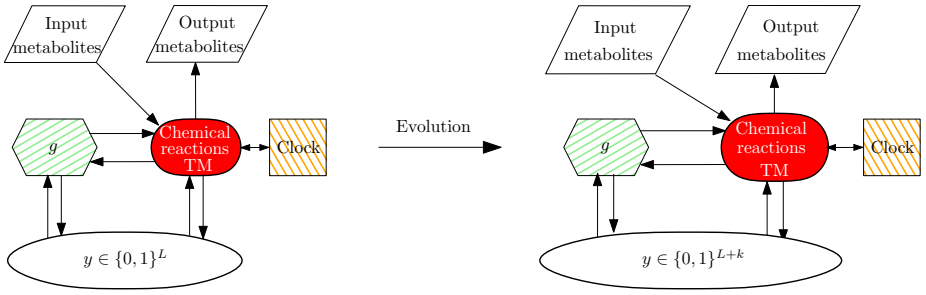


Fig. 4.3 **The sections of a von Neumann computer can be found in evolving bacteria.** The string y is a knockout genetic strategy, i.e., the program stored in the RAM and responsible for protein synthesis. The function g represents the control unit of the computer, interprets the program y and handles gene sets so as to produce desired amount of proteins. The metabolism is the processing unit, composed of all the chemical reactions of the bacterium and mapped through a Minsky machine. The clock module communicates with the register logic module in a way that ensures a low probability of error per step. Metabolites of interest are produced as output of the molecular machine. The evolution and growth of a bacterium depend on duplication and mutation events occurring in the program (genome) and in its length.

4.4 Metabolic networks as Turing machines

Interestingly, the metabolism of a bacterium can be regarded as the processing unit of the von Neumann architecture, as it has been proved to be able to perform computation [152]. More specifically the metabolism can be mapped to a Minsky register machine (RM), where each register represents the molecular count of a single chemical species, and is a left-handed tape that stores non-negative integers by writing stacks of marks on the tape. A RM is equivalent to a multi-tape TM.

4.4.1 A map between a chemical reaction network and a register machine

The chemical reaction network of a bacterium can be simulated with a RM, by defining sets for state species and register species [152]. In the set of register species $\{H_r\}$, each H_r counts how many molecules of species r are present, and corresponds to the rth register of the RM. In the set of state species $\{D_i\}$, each D_i corresponds to the ith state of the RM. More specifically, D_i is an auxiliary species representing only one of the states of the metabolic network, and does not correspond to any existing chemical species. For every state of the RM, the molecular count of the state species associated with that state will be 1, and the count of all the other state species will be 0. The instruction $inc(i,r,j)$ is mapped to the chemical reaction $D_i \rightarrow D_j + H_r$, and the instruction $dec(i,r,j,k)$ is mapped to the reaction $D_i + H_r \rightarrow D_j$ if $H_r > 0$, or to $D_i \rightarrow D_k$ if $H_r = 0$. Note that the reaction $D_i \rightarrow D_k$ can occur only if H_r is over, since the rth register cannot be decreased and the reaction $D_i + H_r \rightarrow D_j$ is not feasible. In this way, the bacterium executes a "test for zero". Finally, the *halt* instruction is equivalent to the cell death, when no further chemical reactions take place [64].

As regards the computation rate in a cell, given that the total number of proteins per cell is approximately $5 \cdot 10^7$ (e.g., for the budding yeast [67]), and on average enzymes carry out $10\ s^{-1}$ reactions (far lower than the common examples found in textbooks [17]), we have $5 \cdot 10^8$ instructions performed every second by the molecular machine. Each instruction consists of multiple decrement and increment instructions for the registers involved in the reaction, e.g., executed through a multi-core architecture. This leads us to estimate a $5 \cdot 10^8$ Hz (or 0.5 GHz) computation rate in each yeast cell. Reversible reaction can be translated into two different instructions, therefore increasing the computational capability of the molecular machine.

A program executed in an organism could be able to implement the genetic strategy proposed by GDMO as the program y. Furthermore, since the simulated TM is an universal machine, our mapping would allow a bacterium to theoretically perform any computation performed by a computer, using its species and reactions characterized by their flux. However, the desired output can only be guaranteed with high probability, due to the inherent stochasticity and constraints of chemical reaction networks.

4.4.2 A clock module to enforce the order of reactions

In order to enforce the order of reactions that take place in the register machine, a clock module can be used to communicate with the register logic module [43]. The clock sends

a signal in the form of an auxiliary species C to the register module, and when a step is performed in that module, C is converted into another auxiliary species T, which is sent back to the clock. After a given delay, the clock sends C again to perform another step. Like the state species $\{D_i\}_i$, C and T are auxiliary species with no actual chemical counterpart.

The delay can be controlled in a way that ensures a low probability of error per step. More specifically, an error of the register machine consists of disabling a decrement reaction $D_i + H_r \to D_j$ involving a species H_r that is still available but with only one molecule left ($|H_r| = 1$). This error can happen if a new clock signal C is sent before the reaction $D_i + H_r \to D_j$ takes place during a decrement step, therefore D_i has not been captured and the reaction $D_i + C \to D_0 + T$ could take place first; if this happens, the presence of D_0 would incorrectly indicate that the register H_r is empty.

The case where $|H_r| = 1$ is indeed the scenario most prone to error because of the low probability that the last decrement step $D_i + H_r \to D_j$ takes place quickly, due to the presence of only one molecule of H_r in the chemical solution. Since the decrement reaction has a rate independent of the clock (e.g., related to the FBA flux rate at steady state, or to the number of molecules of reactants), the role of the clock here must be to delay the release of C. This imposed delay maximizes the probability that the correct decrement reaction $D_i + H_r \to D_j$ occurs first, and therefore minimizes the probability of error per step. If this happens correctly, D_i is consumed and consequently $D_i + C \to D_0 + T$ is not allowed to take place, while the reaction $D_j + C \to D_0 + T$ will take place if another decrement is requested for H_r.

4.5 Gene duplication and transfer in communicating molecular machines

Bacterial conjugation is a genetic transfer that involves cell-to-cell interaction between donor and recipient cells. The importance of horizontal/lateral gene transfer (LGT) in shaping the genomes of prokaryotic organisms has been recognized in recent years as a result of analyses carried out on an increasing number of available genome sequences. LGT is largely due to the transfer and recombination activities of mobile genetic elements. Let us consider the case that genes on some chromosome can circulate and be transmitted among interacting bacteria. (This can occur, for instance, when a bacterium acquires antibiotic resistance, i.e., tries to become immune to some antibiotic drug.) Here we try to address the question whether the computation occurring in each bacterium is affected, and to what extent. An exchange of genes modifies the program running in each molecular machine, and

consequently affects the protein production in all the bacteria involved, except the donors if the gene is copied. This leads to a change in the evolution of the bacterial network [21].

In a communication session between two bacteria, a bacterium duplicates one piece of its genome and sends it to another bacterium, which engulfs it and increases its computational capability, consequently decreasing the speed of cell division. Therefore, the computation in the second bacterium changes accordingly, whereas the operations taking place in the first bacterium are not affected. Indeed, in the mapping between a bacterium and a von Neumann architecture, the metabolites are mapped to the variables of the code. Since the engulfed genes were needed by the second bacterium, they are likely to be responsible for reactions involved in the production or consumption of its key metabolites (i.e., increment or decrement of internal variables). Hence, they are likely to have a significant effect on the next instructions performed by the TM, as well as on the whole metabolic network. Conversely, since the first bacterium performs a duplication that is not needed by itself, the operations associated with the gene duplication are expected to be carried out by dedicated tapes of the TM.

Gene duplication is another source of new cellular functions, giving improved computational capability to the organism. In an evolutionary process, given a sequence of genes, a subsequence of genes (e.g., an operon), or even a single gene are often duplicated and inserted somewhere else in the sequence [38]. This process is referred to as gene amplification or gene duplication. Given an organism with L gene sets, and assuming each gene set is composed of a single gene, without loss of generality, let us denote by y the array representing the sequence of its genes. Let us assume that the duplication of the last k genes is performed:

$$y = (y_1, ..., y_L) \quad \longrightarrow \quad y = (y_1, ..., y_L, y_{L+1}, ..., y_{L+k}). \tag{4.7}$$

Initially, the *condition of duplication* holds, i.e., the last gene was duplicated k times, namely $y_l = y_{l-k}$, $\forall l = L+1, ..., L+k$. In general, the duplication is a stochastic process and the condition of duplication is not always true. After the duplication, the string becomes $y = (y_1, ..., y_{L'})$, where $L' = L+k$, due to the fact that mutations affect both new and existing genes. Each gene y_l codes for a reaction, say $D_i \to D_j + H_r$, and therefore for the instruction $inc(i, r, j)$ in the RM. After the duplication, both y_l and y_{l+k} will be responsible for the same reaction $D_i \to D_j + H_r$. Conversely, after the mutation, the mutated gene y'_{l+k} will be responsible for another reaction, say $D_{i'} \to D_{j'} + H_{r'}$. If the reaction is not already controlled by other genes, a new reaction $inc(i', r', j')$ is now operating in the RM, namely there has been an increase of complexity of the metabolic machine.

Each gene engulfment, or gene duplication followed by a mutation, can change the computational capability of the organism, i.e., the performance of the metabolic machine associated with it. This increases the number of feasible solutions in the multi-objective optimization, and therefore the area under the Pareto front. In cases like biofilms or bacterial communities, the goal may be global and shared among bacteria. As shown in Fig. 4.4, a bacterium can engulf a piece of genome duplicated and sent by another bacterium. This process increases the genome length, allowing more knockout strategies, a different shape of the Pareto front, and causing not merely an increased complexity of the control function, but also a larger capability of intake and output production.

A mutation can be thought of as a stochastic process proposing a new instruction for the machine, while natural selection, acting as a biological ratchet gear, can keep it or discard it. Consequently, the combination of stochastic processes and natural selection shapes the computational complexity of an evolving organism. In particular, the genome amplification increases the number of available chemical reactions, creating new increment and decrement instruction in the RM associated with the metabolism, thus increasing the computational power of the whole metabolic machine. However, since a longer genome needs to be duplicated during cell division, the growth rate is likely to be affected. Further discussion on the topic can be found in [6].

4.6 Related work and final remarks

The main theoretical results presented in this chapter have been published in Theoretical Computer Science [9], and in the Proceedings of the Turing Centenary Conference [5]. The problem of communicating bacteria is further discussed in a paper published in Nano Communication Networks [6]. I performed all the analyses related to the mapping described in this chapter.

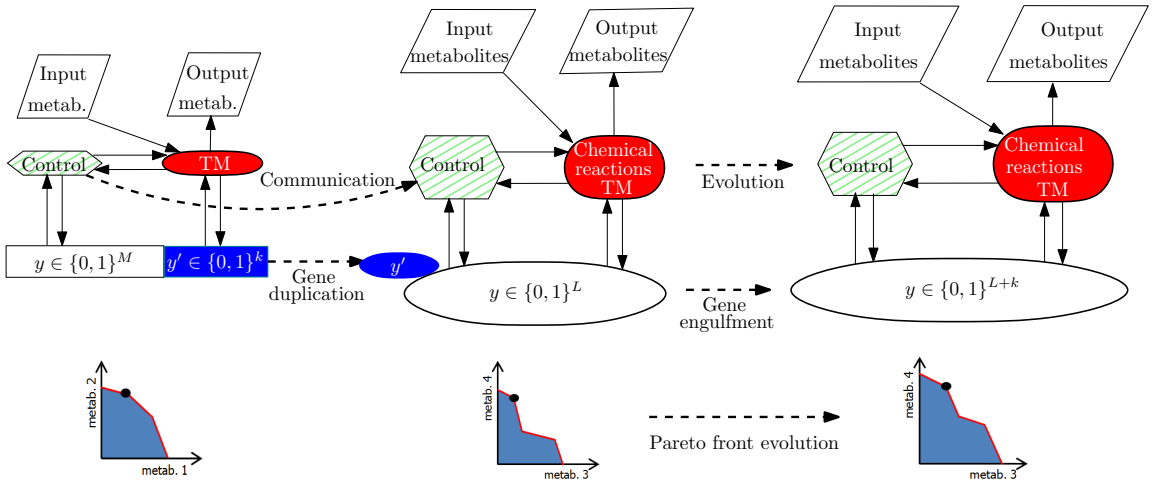


Fig. 4.4 **Interacting bacteria can be thought of as communicating von Neumann architectures (left, center).** Due to the communication between the two bacteria (left, center), the second bacterium engulfs a piece of genome (right) duplicated by the first bacterium. In a multi-objective optimization analysis, this results in an increment of the area under the Pareto front. In other words, the evolution and the growth of a bacterium can be the result of duplication and mutation events in the genome of another bacterium. For instance, the bacterium may engulf k genes and increase the size of its chemical reaction network, therefore increasing also the ability to import and export metabolites.

Chapter 5

Conclusions and future directions

The advent of high throughput data in biology requires computational techniques for data mining, analysis and extraction of knowledge. Biologically-inspired computation has been used to infer mathematical models, parameter values, or to capture states and transitions at the molecular level [61]. As a result, mathematics, computer science and biology are now tightly linked.

The tools provided in this dissertation represent a starting point for modeling adaptive dynamics in genome-scale models, and for elucidating genotype-phenotype relationships in bacterial cells. In Chapter 2, we developed an optimization framework for genome-scale models, in which we proposed GDMO, a method for multi-objective metabolic engineering based on FBA. We adopt the GDMO algorithm to find genetic manipulations in order to maximize multiple biological requirements, i.e., growth rate and a product of biotechnological interest. In the same optimization framework, we propose a solution for the problem of making the sensitivity analysis pathway-dedicated: we develop a pathway-based sensitivity analysis (PoSA) to investigate the functional components of the model and compute their sensitivity. The robustness analysis supports GDMO and PoSA in that it indicates the components of the model that are likely to "fail".

The Pareto front has a close link with biotechnological productivity. For the biosynthesis, Pareto optimality is important to obtain not only a wide range of Pareto optimal solutions, but also the best trade-off design, providing a visualization of the optimization process and significant details for metabolic design automation. The size of non-dominated solutions, the first derivative and the area under the Pareto front could play a key role for the best design within the same organism or between different organisms. For instance, a reduced size of the Pareto front could suggest the incompleteness of the model in terms of number of reactions modeled; in this case, Pareto optimality could be thought of as a parameter describing the improvement of a model for a bacterium with respect to a previous model

for the same bacterium. The idea of multi-objective optimality can also be harnessed in a network of organisms. Indeed, two communicating organisms can define their behavior by making a trade-off decision using a shared Pareto front in which each organism is associated with one or more axes of the objective space.

Exploratory analysis and comparative metabolic models also suggest that the area under the Pareto front provides an estimate of the number of intermediate solutions which may be exploited for biotechnological purposes (optimization of an additional objective) or to build synthetic pathways (synthetic biology). Given two bacteria or two conditions for the same bacterium, a larger area under the Pareto front probably represents better conditions for adding or optimizing pathways leading to new products. The robustness analysis allows for the discrimination of the optimal strains generated as a result of the multi-objective optimization: the higher the robustness indices, the larger the probability that the bacterium, reproduced in laboratory, maintains the desired performance. Overall, local robustness analysis enables us to reach a better understanding of the fragility of the metabolic network.

Bacterial adaptability to new environmental conditions involves shifts in the gene expression and in the biochemical network, also in places that are not always related directly with the adaptation. For instance, an increase in the amount of a given nutrient supplied to the bacterium will increase the request for the enzymes – and consequently for production of reactants – able to produce that nutrient [134]. Although increasing research efforts have been allocated for attempts at understanding the relation between gene expression changes and phenotype in bacteria, little is known about the contribution of the different omics and different objectives to the phenotypic adaptability and evolution.

To account for bacterial adaptability and omic layers of information, in Chapter 3 we derived METRADE, a novel multi-omic FBA method that implements and combines multiple target optimization with gene expression and codon usage as additional layers for the most complete metabolic data available for *Escherichia coli*. Given an *E. coli* strain with a specific gene expression configuration, METRADE is able to determine the best codon usage for each gene so as to maximize or minimize desired objective functions. We show how the integration of multiple omics can be used to compare different bacterial strains and evaluate the optimal behavior of a bacterium under various conditions. Specifically, the environmental conditions are mapped to genotypic data (gene expression profiles), and finally to phenotypic data (predictions of two or more optimized variables). This enables the use of the Pareto front to predict where the metabolism operates in the objective space [145], to investigate scenarios of adaptability over time, and, coupled with its hypervolume, as an indicator of the evolution of a strain.

GDMO and METRADE scale effectively as the size of the metabolic system and the number of genetic manipulations increase. Moreover, our results show that the multi-objective approach is suitable for discovering genetic design strategies. Our software tools allow for the analysis of an organism in a dynamic multi-omic fashion (genomics, transcriptomics, proteomics) [50, 83], e.g., by evaluating temporal changes in gene expression, codon usage and flux rates at various cellular levels. This allows for these layers to be interpreted as a whole, and to evaluate connections and interactions among them. We provide our algorithms as a MATLAB toolbox.

As described above, the growing availability of multi-omic data, in combination with methods like METRADE and frequently updated genome-scale models, provides a highly comprehensive view of cellular processes at the levels of mRNA, proteins, metabolites, and reaction fluxes. However, to date, interactions at the pathway level cannot be measured directly, and methodologies for prediction of pathway cross-correlations from FBA and reaction fluxes are still missing. Therefore, in joint work with N. Pratanwanich, using METRADE and a genome-scale model of *E. coli* we developed a hybrid method combining multi-omics FBA and Bayesian inference [12]. The aim was to investigate the cellular activities of a bacterium from the transcriptomic, fluxomic and pathway standpoints under different environmental conditions. We determined the degree of pathway responsiveness and detected pathway cross-correlations (*crosstalks*) starting from gene expression profiles. A dynamical analysis allowed us to predict profiles of pathway activation over time in changing environmental conditions.

In this research work, we were particularly interested in investigating pathway crosstalks in different glucose and oxygen conditions. In Fig. 5.1, we take into account the maximization of biomass and acetate in *E. coli*, and we compare pathway responsiveness to three different conditions, showing the correlations between pathways. In Table 5.1 we show the responsiveness of the three most responsive pathways in different conditions, with changing glucose and oxygen uptake. The pathway PID:5 (nucleotide salvage) is the most responsive in those conditions in which only one pathway is responsive. Furthermore, when two pathways are highly responsive, these are PID:17 (valine, leucine, and isoleucine metabolism) and PID:25 (alanine and aspartate metabolism). For those conditions that have three responsive pathways, these are PID:5, PID:17, PID:25. This shows strong correlation among these three pathways when the model is aimed at the maximization of acetate and biomass.

While metabolic models provide useful insights into complex cellular processes, the development of a whole-cell multiscale model is still considered to be the Holy Grail in computational systems biology [160]. The work from Karr et al. [85] provides a first draft of

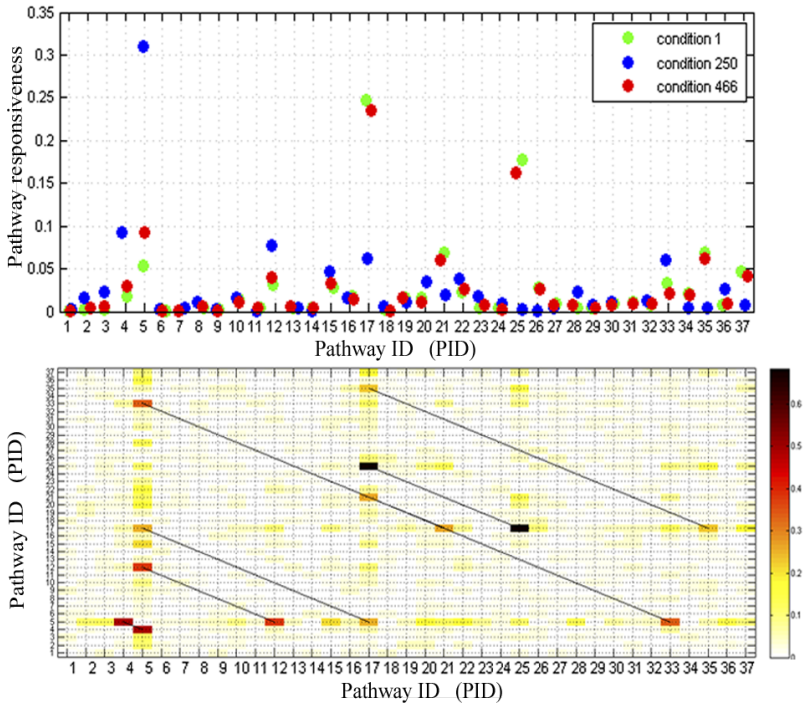


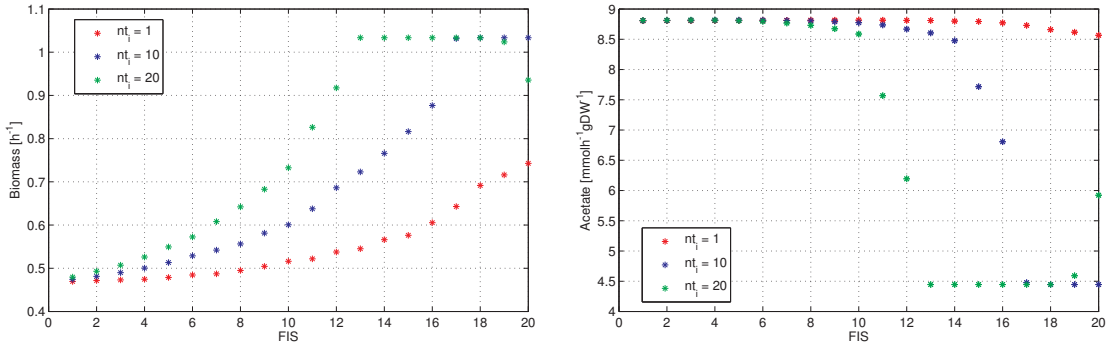
Fig. 5.1 **Bayesian pathway analysis in *E. coli*.** Comparison of pathway responsiveness to different conditions (top). For example, among the 37 pathways in the *E. coli* model, two pathways (PID:17 and 25) have high responsiveness to condition 1 and 466, while only one pathway (PID:5) highly responds to condition 250. Crosstalks between pathways can be viewed as a correlation matrix (bottom). The 466 experimental conditions we used for this analysis are publicly available in [58].

a whole-cell simulation for a small bacterium, but we are still far from reaching whole-cell modeling for larger cells. Starting from the *E. coli* FBA metabolic reconstruction, we are also interested in a coarse-grained approach to whole cell modelling. We are currently focusing on the regulation of the energy metabolism by the master regulator FIS. The core of our approach is represented by a dynamical hybrid model of FIS regulation: the expression of the FIS protein (modelled as an ordinary differential equation) regulates the twisting of the DNA (modelled as a discrete birth/death process representing the number of twists in the DNA), which regulates access of genetic material to the transcriptional machinery, and in turn controls the production of FIS. FIS is linked to the metabolism by modelling the way in which it regulates 104 genes in the FBA model. Another parameter that affects regulation is DNA coiling. Our preliminary results (Fig. 5.2) show how FIS and DNA coiling are able to regulate the growth rate and the synthetic yield of *E. coli*.

Further extensions of FBA in the field of whole-cell modeling may focus on combining estimated mutation rates, transcriptomic program complexity and variance, and selection

	High glucose			Low glucose	
PID	Pathway	Aerobic	Anaerobic	Aerobic	Anaerobic
5	Nucleotide salvage	0.0865	0.0965	0.1366	0.1714
17	Valine, leucine, and isoleucine metabolism	0.2219	0.2147	0.1974	0.1590
25	Alanine and aspartate metabolism	0.1544	0.1487	0.1285	0.1076

Table 5.1 **Average responsiveness of the most responsive pathways across aerobic and anaerobic conditions of high and low glucose in *E. coli*.** The high/low glucose threshold is 10 $\text{mmolh}^{-1}\text{gDW}^{-1}$. PID:5 is important in anaerobic conditions on low glucose, while PID:17 and PID:25 both exhibit a key role in aerobic conditions on high glucose, highlighting a pathway crosstalk between them.



(a) (b)

Fig. 5.2 **Variation of biomass and acetate against FIS concentration (fold-change with respect to the initial concentration).** The parameter nt_i is associated with DNA twists, varies in {1, 10, 20}, and models the level of bending and coiling of the DNA structure, which affects rate of translation of genes into proteins. For every value of nt_i, the biomass increases as the FIS value rises, reaching the maximum value for $nt_i = 10$. The acetate reaches the highest values for $nt_i = 1$.

coefficients, with the aim of providing an upper-bound estimation of the number of traits that can become and remain adapted by direct natural selection, i.e., the "many-traits" and "few-traits" phenotype-fitness maps [56]. The framework can also be extended to other bacteria that show variation of codon usage [94], and to other environments. A very interesting medical application would be to simulate *in silico* the tumor conditions analyzed for the *Clostridium* bacteria [135].

With our framework, we also plan to study endosymbiosis and host-pathogen interactions, therefore designing, analyzing and optimizing the metabolism of systems composed of different interacting species living and interacting in the same context, as introduced by Cottret et al. [45]. This type of analysis could highlight the complementarity of separate metabolic networks. For instance, mitochondria and chloroplasts are different organelles, (usually) both found in plants, and are part of the same functional pipeline: starting from

CO_2, the photosynthesis in the chloroplast creates glucose that enters the mitochondria to create ATP. We plan to study and optimize interacting metabolic networks using the DyMMM framework [182].

Genome-scale modeling and the methods proposed in this dissertation can be also applied to human cells. Recently, a community-driven reconstruction of the human metabolism has been developed, which incorporates also information on enzymes and genome, including the relationships between genes, proteins and reactions [159]. This model, named Recon2, contains all the chemical reactions known to take place in the human cell, and has been recently tailored to different tissues, therefore obtaining unprecedented tools for analysis of cell-specific behavior. We will add additional omic layers to the models, for more accurate predictions of the phenotype (Fig. 5.3). Furthermore, using multi-tissue phenotype data sets, we would like to better calibrate Recon2 and improve its prediction accuracy. We are also interested in adapting these techniques for recently developed cancer models [118], and in cross-comparing their behavior with that of normal cells across a wide set of environmental conditions.

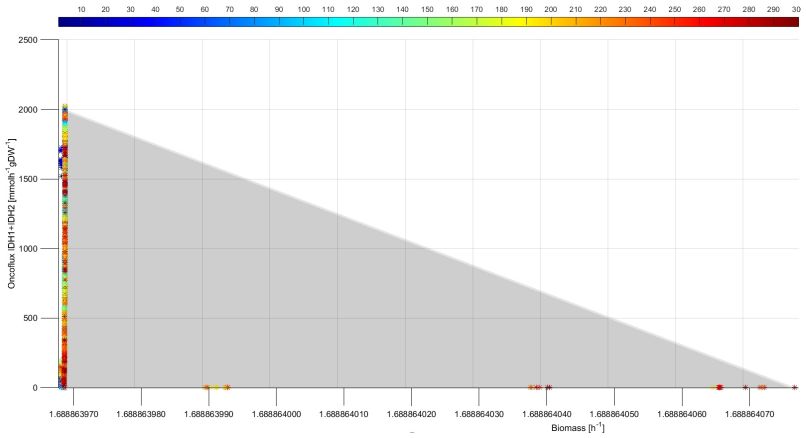


Fig. 5.3 **Multi-objective minimization of isocitrate dehydrogenase (shown to be a metabolic oncogenic factor) and maximization of growth rate in a human cell.** This technique is useful to detect the regions where the cell operates in the objective space. Each point in the phenotypic objective space represents a solution, and corresponds to a specific gene expression profile. Solutions are denoted by progressively warmer colors according to the time step when they have been generated in the optimization process.

Both GDMO and METRADE could be further extended and tuned to specific cases, in presence of additional information on enzymes. A further extension of the framework could be the task of inferring the topology of the network of conditions using a multidimensional scaling approach [178]. Such computational analysis can have a great impact especially for the large fraction of microorganisms that have been already identified but never cultured so

far. For instance, a step forward would be to infer pathways to combat condition-dependent infections caused by bacteria involved in both plant and animal infections.

As regards human infections caused by bacteria, using multi-objective optimization and model-predictive control applied to multi-scale models (e.g., a host-pathogen model of *E. coli* and human metabolism Recon2), we aim to model the problem of antimicrobial resistance, recently reported as a major health concern by the World Health Organization [175], the UK and US governments through the launch of the global "combatting antimicrobial resistance" initiative. In this regard, due to memory effects and relationships among the drug targets, recent experiments disproved the common idea that the overall effect of a combination of drugs is less or equal to the sum of its ingredients. Therefore, the design of an optimized multi-drug therapy would dramatically improve its positive effect.

In Chapter 4, to paraphrase Stan Ulam in his autobiography [77], we focused not only on what computer science can do for biology, but also on how biology can inspire studies in theoretical computer science. We showed how robust genetic interventions in cells can be framed as optimal programs to be run in a molecular machine, in order to extend and modify the behavior of cells and cell aggregates. For instance, programs can instruct cells to make logic decisions according to environmental factors and current cell state. A program embedded in a cell could allow its metabolic network to work with a specific user-imposed aim, and therefore to perform computation.

References

[1] Adleman, L. (1994). Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024.

[2] Åkesson, M., Förster, J., and Nielsen, J. (2004). Integration of gene expression data into genome-scale metabolic models. *Metabolic engineering*, 6(4):285–293.

[3] Alper, H., Fischer, C., Nevoigt, E., and Stephanopoulos, G. (2005). Tuning genetic control through promoter engineering. *Proceedings of the National Academy of Sciences of the United States of America*, 102(36):12678–12683.

[4] Amos, M. (2004). *Cellular Computing*. Series in Systems Biology. Oxford University Press.

[5] Angione, C., Carapezza, G., Costanza, J., Lió, P., and Nicosia, G. (2012). Computing with Metabolic Machines. *Turing-100. Volume 10 of EPiC Series*, 10:1–15.

[6] Angione, C., Carapezza, G., Costanza, J., Lió, P., and Nicosia, G. (2013a). Design and strain selection criteria for bacterial communication networks. *Nano Communication Networks*, 4(4):155–163.

[7] Angione, C., Carapezza, G., Costanza, J., Lió, P., and Nicosia, G. (2013b). Pareto Optimality in Organelle Energy Metabolism Analysis. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 10(4):1032–1044.

[8] Angione, C., Costanza, J., Carapezza, G., Lió, P., and Nicosia, G. (2013c). A design automation framework for computational bioenergetics in biological networks. *Molecular BioSystems*, 9(10):2554–2564.

[9] Angione, C., Costanza, J., Carapezza, G., Lió, P., and Nicosia, G. (2015a). Analysis and design of molecular machines. *Theoretical Computer Science*, 599:102–117.

[10] Angione, C., Costanza, J., Carapezza, G., Lió, P., and Nicosia, G. (2015b). Multi-Target analysis and design of Mitochondrial Metabolism. *PLoS One*, 10(9):e0133825.

[11] Angione, C. and Lió, P. (2015). Predictive analytics of environmental adaptability in multi-omic network models. *Scientific Reports*, 5:15147.

[12] Angione, C., Pratanwanich, N., and Lió, P. (2015c). A hybrid of metabolic flux analysis and Bayesian factor modeling for multi-omics temporal pathway activation. *ACS Synthetic Biology*, 4(8):880–889.

[13] Arias, C. F., Catalán, P., Manrubia, S., and Cuesta, J. A. (2014). toyLIFE: a computational framework to study the multi-level organisation of the genotype-phenotype map. *Scientific reports*, 4:7549.

[14] Aung, H. W., Henry, S. A., and Walker, L. P. (2013). Revising the representation of fatty acid, glycerolipid, and glycerophospholipid metabolism in the consensus model of yeast metabolism. *Industrial Biotechnology*, 9(4):215–228.

[15] Bader, J. and Zitzler, E. (2011). Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76.

[16] Bagnoli, F. and Liò, P. (1995). Selection, mutations and codon usage in a bacterial model. *Journal of Theoretical Biology*, 173(3):271–281.

[17] Bar-Even, A., Noor, E., Savir, Y., Liebermeister, W., Davidi, D., Tawfik, D., and Milo, R. (2011). The moderately efficient enzyme: evolutionary and physicochemical trends shaping enzyme parameters. *Biochemistry*, 50(21):4402–4410.

[18] Barker, B., Sadagopan, N., Wang, Y., Smallbone, K., Myers, C. R., Xi, H., Locasale, J. W., and Gu, Z. (2014). A robust and efficient method for estimating enzyme complex abundance and metabolic flux from expression data. *arXiv preprint arXiv:1404.4755*.

[19] Becker, J., Zelder, O., Häfner, S., Schröder, H., and Wittmann, C. (2011). From zero to hero - design-based systems metabolic engineering of corynebacterium glutamicum for l-lysine production. *Metabolic engineering*, 13(2):159–168.

[20] Becker, S. A. and Palsson, B. Ø. (2008). Context-specific metabolic networks are consistent with experiments. *PLoS computational biology*, 4(5):e1000082.

[21] Ben-Hur, A. and Siegelmann, H. (2003). Computation in gene networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 14(1):145–151.

[22] Birch, E. W., Udell, M., and Covert, M. W. (2014). Incorporation of flexible objectives and time-linked simulation with flux balance analysis. *Journal of theoretical biology*, 345:12–21.

[23] Blazier, A. S. and Papin, J. A. (2012). Integration of expression data in genome-scale metabolic network reconstructions. *Frontiers in Physiology*, 3:299.

[24] Bordbar, A., Monk, J. M., King, Z. A., and Palsson, B. O. (2014). Constraint-based models predict metabolic and associated cellular functions. *Nature Reviews Genetics*, 15(2):107–120.

[25] Boxma, B., de Graaf, R., van der Staay, G., van Alen, T., Ricard, G., Gabaldón, T., van Hoek, A., Moon-van der Staay, S., Koopman, W., van Hellemond, J., et al. (2005). An anaerobic mitochondrion that produces hydrogen. *Nature*, 434(7029):74–79.

[26] Bray, D. et al. (1995). Protein molecules as computational elements in living cells. *Nature*, 376(6538):307–312.

[27] Brent, R. and Bruck, J. (2006). 2020 computing: Can computers help to explain biology? *Nature*, 440(7083):416–417.

[28] Bryant, B. (2012). Chromatin computation. *PloS one*, 7(5):e35703.

[29] Bui, E., Bradley, P., and Johnson, P. (1996). A common evolutionary origin for mitochondria and hydrogenosomes. *Proceedings of the National Academy of Sciences*, 93(18):9651.

[30] Burgard, A., Pharkya, P., and Maranas, C. (2003). Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnology and bioengineering*, 84(6):647–657.

[31] Calheiros Gomes, L. and Simões, M. (2012). 13c metabolic flux analysis: from the principle to recent applications. *Current Bioinformatics*, 7(1):77–86.

[32] Callaway, D. S., Newman, M. E., Strogatz, S. H., and Watts, D. J. (2000). Network robustness and fragility: Percolation on random graphs. *Physical review letters*, 85(25):5468.

[33] Cannarozzi, G., Schraudolph, N. N., Faty, M., von Rohr, P., Friberg, M. T., Roth, A. C., Gonnet, P., Gonnet, G., and Barral, Y. (2010). A role for codon order in translation dynamics. *Cell*, 141(2):355–367.

[34] Carothers, J. M., Goler, J. A., Juminaga, D., and Keasling, J. D. (2011). Model-driven engineering of rna devices to quantitatively program gene expression. *Science*, 334(6063):1716–1719.

[35] Carrera, J., Estrela, R., Luo, J., Rai, N., Tsoukalas, A., and Tagkopoulos, I. (2014). An integrative, multi-scale, genome-wide model reveals the phenotypic landscape of escherichia coli. *Molecular systems biology*, 10(7):735.

[36] Chandrasekaran, S. and Price, N. D. (2010). Probabilistic integrative modeling of genome-scale metabolic and regulatory networks in escherichia coli and mycobacterium tuberculosis. *Proceedings of the National Academy of Sciences*, 107(41):17845–17850.

[37] Chang, R., Ghamsari, L., Manichaikul, A., Hom, E., Balaji, S., Fu, W., Shen, Y., Hao, T., Palsson, B., and Salehi-Ashtiani, K. (2011). Metabolic network reconstruction of chlamydomonas offers insight into light-driven algal metabolism. *Molecular systems biology*, 7(1).

[38] Chauve, C., El-Mabrouk, N., and Tannier, E. (2013). *Models and algorithms for genome evolution*. Springer.

[39] Chindelevitch, L., Trigg, J., Regev, A., and Berger, B. (2014). An exact arithmetic toolbox for a consistent and reproducible structural analysis of metabolic network models. *Nature communications*, 5.

[40] COBRA commentary (2015). http://nbviewer.ipython.org/gist/aebrahim/c58ab87d398288187505.

[41] Coello, C. A. (2000). An updated survey of ga-based multiobjective optimization techniques. *ACM Computing Surveys (CSUR)*, 32(2):109–143.

[42] Colijn, C., Brandes, A., Zucker, J., Lun, D., Weiner, B., Farhat, M., Cheng, T., Moody, D., Murray, M., and Galagan, J. (2009). Interpreting expression data with metabolic flux models: predicting mycobacterium tuberculosis mycolic acid production. *PLoS computational biology*, 5(8):e1000489.

[43] Cook, M., Soloveichik, D., Winfree, E., and Bruck, J. (2009). Programmability of chemical reaction networks. *Algorithmic Bioprocesses*, pages 543–584.

[44] Costanza, J., Carapezza, G., Angione, C., Lió, P., and Nicosia, G. (2012). Robust Design of Microbial Strains. *Bioinformatics*, 28(23):3097–3104.

[45] Cottret, L., Milreu, P., Acuña, V., Marchetti-Spaccamela, A., Stougie, L., Charles, H., and Sagot, M. (2010). Graph-based analysis of the metabolic exchanges between two co-resident intracellular symbionts, baumannia cicadellinicola and sulcia muelleri, with their insect host, homalodisca coagulata. *PLoS computational biology*, 6(9):e1000904.

[46] Covert, M. W., Schilling, C. H., and Palsson, B. (2001). Regulation of gene expression in flux balance models of metabolism. *Journal of theoretical biology*, 213(1):73–88.

[47] Covert, M. W., Xiao, N., Chen, T. J., and Karr, J. R. (2008). Integrating metabolic, transcriptional regulatory and signal transduction models in Escherichia coli. *Bioinformatics*, 24(18):2044–2050.

[48] Csárdi, G., Franks, A., Choi, D. S., Airoldi, E. M., and Drummond, D. A. (2014). Accounting for experimental noise reveals that mRNA levels, amplified by post-transcriptional processes, largely determine steady-state protein levels in yeast. *bioRxiv*, page 009472.

[49] Csárdi, G., Franks, A., Choi, D. S., Airoldi, E. M., and Drummond, D. A. (2015). Accounting for experimental noise reveals that mrna levels, amplified by post-transcriptional processes, largely determine steady-state protein levels in yeast. *PLoS Genetics*, 11(5).

[50] De Keersmaecker, S. C., Thijs, I., Vanderleyden, J., and Marchal, K. (2006). Integration of omics data: how well does it work for bacteria? *Molecular microbiology*, 62(5):1239–1250.

[51] de Sousa Abreu, R., Penalva, L. O., Marcotte, E. M., and Vogel, C. (2009). Global signatures of protein and mRNA expression levels. *Mol. BioSyst.*, 5(12):1512–1526.

[52] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197.

[53] Dethlefsen, L. and Schmidt, T. M. (2005). Differences in codon bias cannot explain differences in translational power among microbes. *BMC bioinformatics*, 6(1):3.

[54] Dittrich, C. R., Bennett, G. N., and San, K.-Y. (2005). Characterization of the acetate-producing pathways in escherichia coli. *Biotechnology progress*, 21(4):1062–1067.

[55] Dobzhansky, T. (1973). Nothing in biology makes sense except in the light of evolution. *American Biology Teacher*, 35:125–129.

[56] Draghi, J. A., Parsons, T. L., Wagner, G. P., and Plotkin, J. B. (2010). Mutational robustness can facilitate adaptation. *Nature*, 463(7279):353–355.

[57] Edwards, J. S. and Palsson, B. Ø. (2000). Metabolic flux balance analysis and the in silico analysis of Escherichia coli K-12 gene deletions. *BMC bioinformatics*, 1(1):1.

[58] Faith, J. J., Hayete, B., Thaden, J. T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J. J., and Gardner, T. S. (2007). Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS biology*, 5(1):e8.

[59] Farasat, I., Kushwaha, M., Collens, J., Easterbrook, M., Guido, M., and Salis, H. M. (2014). Efficient search, mapping, and optimization of multi-protein genetic systems in diverse bacteria. *Molecular systems biology*, 10(6).

[60] Feist, A. M., Henry, C. S., Reed, J. L., Krummenacker, M., Joyce, A. R., Karp, P. D., Broadbelt, L. J., Hatzimanikatis, V., and Palsson, B. Ø. (2007). A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. *Molecular systems biology*, 3(1).

[61] Fellermann, H. and Cardelli, L. (2014). Programming chemistry in dna-addressable bioreactors. *Journal of The Royal Society Interface*, 11(99):20130987.

[62] Firczuk, H., Kannambath, S., Pahle, J., Claydon, A., Beynon, R., Duncan, J., Westerhoff, H., Mendes, P., and McCarthy, J. E. (2013). An in vivo control map for the eukaryotic mRNA translation machinery. *Molecular systems biology*, 9(1):635.

[63] Fong, S. S., Joyce, A. R., and Palsson, B. Ø. (2005). Parallel adaptive evolution cultures of Escherichia coli lead to convergent growth phenotypes with different gene expression states. *Genome research*, 15(10):1365–1372.

[64] Franco, G., Guzzi, P. H., Manca, V., and Mazza, T. (2006). Mitotic oscillators as mp graphs. In *Membrane Computing*, pages 382–394. Springer.

[65] Fraser, H. B. (2011). Genome-wide approaches to the study of adaptive gene expression evolution. *Bioessays*, 33(6):469–477.

[66] Fraser, H. B., Hirsh, A. E., Steinmetz, L. M., Scharfe, C., and Feldman, M. W. (2002). Evolutionary rate in the protein interaction network. *Science*, 296(5568):750–752.

[67] Futcher, B., Latter, G., Monardo, P., McLaughlin, C., and Garrels, J. (1999). A sampling of the yeast proteome. *Molecular and Cellular Biology*, 19(11):7357–7368.

[68] Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of computational physics*, 22(4):403–434.

[69] Goodman, D. B., Church, G. M., and Kosuri, S. (2013). Causes and effects of n-terminal codon bias in bacterial genes. *Science*, 342(6157):475–479.

[70] Gopalakrishnan, S. and Maranas, C. D. (2015). 13c metabolic flux analysis at a genome-scale. *Metabolic Engineering*.

[71] Guimaraes, J. C., Rocha, M., and Arkin, A. P. (2014). Transcript level and sequence determinants of protein abundance and noise in Escherichia coli. *Nucleic acids research*, 42(8):4791–4799.

[72] Gunasekera, K., Wüthrich, D., Braga-Lagache, S., Heller, M., and Ochsenreiter, T. (2012). Proteome remodelling during development from blood to insect-form trypanosoma brucei quantified by silac and mass spectrometry. *BMC genomics*, 13(1):556.

[73] Gustafsson, C., Govindarajan, S., and Minshull, J. (2004). Codon bias and heterologous protein expression. *Trends in biotechnology*, 22(7):346–353.

[74] Hafner, M., Koeppl, H., Hasler, M., and Wagner, A. (2009). 'glocal' robustness analysis and model discrimination for circadian oscillators. *PLoS computational biology*, 5(10):e1000534.

[75] Hammer, K., Mijakovic, I., and Jensen, P. R. (2006). Synthetic promoter libraries–tuning of gene expression. *Trends in biotechnology*, 24(2):53–55.

[76] Henry, I. and Sharp, P. M. (2007). Predicting gene expression level from codon usage bias. *Molecular biology and evolution*, 24(1):10–12.

[77] Hoffman, P. (1987). The man who loves only numbers. *Atlantic Monthly*, 260(5):60.

[78] Hui, S., Silverman, J. M., Chen, S. S., Erickson, D. W., Basan, M., Wang, J., Hwa, T., and Williamson, J. R. (2015). Quantitative proteomic analysis reveals a simple strategy of global resource allocation in bacteria. *Molecular systems biology*, 11(2):784.

[79] Ibarra, R. U., Edwards, J. S., and Palsson, B. O. (2002). Escherichia coli k-12 undergoes adaptive evolution to achieve in silico predicted optimal growth. *Nature*, 420(6912):186–189.

[80] Jakočiūne, D., Bisgaard, M., Hervé, G., Protais, J., Olsen, J. E., and Chemaly, M. (2014). Effects of environmental conditions on growth and survival of salmonella in pasteurized whole egg. *International journal of food microbiology*, 184:27–30.

[81] Jensen, P. A. and Papin, J. A. (2011). Functional integration of a metabolic network model and expression data without arbitrary thresholding. *Bioinformatics*, 27(4):541–547.

[82] Jovanovic, M., Rooney, M. S., Mertins, P., Przybylski, D., Chevrier, N., Satija, R., Rodriguez, E. H., Fields, A. P., Schwartz, S., Raychowdhury, R., et al. (2015). Dynamic profiling of the protein life cycle in response to pathogens. *Science*, 347(6226):1259038.

[83] Joyce, A. R. and Palsson, B. Ø. (2006). The model organism as a system: integrating'omics' data sets. *Nature Reviews Molecular Cell Biology*, 7(3):198–210.

[84] Kanehisa, M., Goto, S., Hattori, M., Aoki-Kinoshita, K. F., Itoh, M., Kawashima, S., Katayama, T., Araki, M., and Hirakawa, M. (2006). From genomics to chemical genomics: new developments in kegg. *Nucleic acids research*, 34(suppl 1):D354–D357.

[85] Karr, J., Sanghvi, J., Macklin, D., Gutschow, M., Jacobs, J., Bolival, B., Assad-Garcia, N., Glass, J., and Covert, M. (2012). A whole-cell computational model predicts phenotype from genotype. *Cell*, 150(2):389–401.

[86] Keseler, I. M., Bonavides-Martínez, C., Collado-Vides, J., Gama-Castro, S., Gunsalus, R. P., Johnson, D. A., Krummenacker, M., Nolan, L. M., Paley, S., Paulsen, I. T., et al. (2009). Ecocyc: a comprehensive view of escherichia coli biology. *Nucleic acids research*, 37(suppl 1):D464–D470.

[87] Khodayari, A., Zomorrodi, A. R., Liao, J. C., and Maranas, C. D. (2014). A kinetic model of escherichia coli core metabolism satisfying multiple sets of mutant flux data. *Metabolic Engineering*, 25:50–62.

[88] Kim, M. K. and Lun, D. S. (2014). Methods for integration of transcriptomic data in genome-scale metabolic models. *Computational and structural biotechnology journal*, 11(18):59–65.

[89] King, Z. A., Lloyd, C. J., Feist, A. M., and Palsson, B. O. (2015). Next-generation genome-scale models for metabolic engineering. *Current opinion in biotechnology*, 35:23–29.

[90] Kitano, H. (2002). Systems biology: a brief overview. *Science*, 295(5560):1662–1664.

[91] Klamt, S., Schuster, S., and Gilles, E. D. (2002). Calculability analysis in underdetermined metabolic networks illustrated by a model of the central metabolism in purple nonsulfur bacteria. *Biotechnology and bioengineering*, 77(7):734–751.

[92] Kleessen, S. and Nikoloski, Z. (2015). Computational approaches to dissect and understand mechanisms of adaptation. *Molecular Mechanisms in Plant Adaptation*, pages 193–215.

[93] Klumpp, S., Dong, J., and Hwa, T. (2012). On ribosome load, codon bias and protein abundance. *PloS one*, 7(11):e48542.

[94] Krisko, A., Copic, T., Gabaldón, T., Lehner, B., and Supek, F. (2014). Inferring gene function from evolutionary change in signatures of translation efficiency. *Genome biology*, 15(3):R44.

[95] Kusumi, A., Tsunoyama, T. A., Hirosawa, K. M., Kasai, R. S., and Fujiwara, T. K. (2014). Tracking single molecules at work in living cells. *Nature chemical biology*, 10(7):524–532.

[96] Lackner, D. H., Schmidt, M. W., Wu, S., Wolf, D. A., and Bahler, J. (2012). Regulation of transcriptome, translation, and proteome in response to environmental stress in fission yeast. *Genome biology*, 13(4):R25.

[97] Landweber, L. and Kari, L. (2003). Universal molecular computation in ciliates. *Evolution as Computation*, pages 257–274.

[98] Larocque, M., Chénard, T., and Najmanovich, R. (2014). A curated C. difficile strain 630 metabolic network: prediction of essential targets and inhibitors. *BMC systems biology*, 8(1):117.

[99] Lee, D., Smallbone, K., Dunn, W. B., Murabito, E., Winder, C. L., Kell, D. B., Mendes, P., and Swainston, N. (2012). Improving metabolic flux predictions using absolute gene expression data. *BMC systems biology*, 6(1):73.

[100] Li, J. J., Bickel, P. J., and Biggin, M. D. (2014). System wide analyses have underestimated protein abundances and the importance of transcription in mammals. *PeerJ*, 2:e270.

[101] Llaneras, F. and Picó, J. (2008). Stoichiometric modelling of cell metabolism. *Journal of bioscience and bioengineering*, 105(1):1–11.

[102] Lun, D. S., Rockwell, G., Guido, N. J., Baym, M., Kelner, J. A., Berger, B., Galagan, J. E., and Church, G. M. (2009). Large-scale identification of genetic design strategies using local search. *Molecular Systems Biology*, 5(296):1.

[103] Machado, D. and Herrgård, M. (2014). Systematic evaluation of methods for integration of transcriptomic data into constraint-based models of metabolism. *PLoS Comput. Biol*, 10:e1003580.

[104] Mahadevan, R., Edwards, J. S., and Doyle, F. J. (2002). Dynamic flux balance analysis of diauxic growth in escherichia coli. *Biophysical journal*, 83(3):1331–1340.

[105] Maier, T., Schmidt, A., Güell, M., Kühner, S., Gavin, A.-C., Aebersold, R., and Serrano, L. (2011). Quantification of mrna and protein and integration with protein turnover in a bacterium. *Molecular Systems Biology*, 7(1):511.

[106] Mallo, N., Lamas, J., and Leiro, J. M. (2013). Hydrogenosome metabolism is the key target for antiparasitic activity of resveratrol against trichomonas vaginalis. *Antimicrobial agents and chemotherapy*, 57(6):2476–2484.

[107] Mar, J. C., Matigian, N. A., Mackay-Sim, A., Mellick, G. D., Sue, C. M., Silburn, P. A., McGrath, J. J., Quackenbush, J., and Wells, C. A. (2011). Variance of gene expression identifies altered network constraints in neurological disease. *PLoS genetics*, 7(8):e1002207.

[108] Marguerat, S., Schmidt, A., Codlin, S., Chen, W., Aebersold, R., and Bähler, J. (2012). Quantitative analysis of fission yeast transcriptomes and proteomes in proliferating and quiescent cells. *Cell*, 151(3):671–683.

[109] Marion, J. (2012). From turing machines to computer viruses. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1971):3319–3339.

[110] Martín, H. G., Kumar, V. S., Weaver, D., Ghosh, A., Chubukov, V., Mukhopadhyay, A., Arkin, A., and Keasling, J. D. (2015). A method to constrain genome-scale models with 13 c labeling data. *PLOS Comput Biol*, 11(9):e1004363.

[111] McCloskey, D., Palsson, B. Ø., and Feist, A. M. (2013). Basic and applied uses of genome-scale metabolic network reconstructions of Escherichia coli. *Molecular Systems Biology*, 9(1):661.

[112] Meysman, P., Sonego, P., Bianco, L., Fu, Q., Ledezma-Tejeida, D., Gama-Castro, S., Liebens, V., Michiels, J., Laukens, K., Marchal, K., et al. (2014). Colombos v2. 0: an ever expanding collection of bacterial expression compendia. *Nucleic acids research*, 42(D1):D649–D653.

[113] Minsky, M. (1967). *Computation*. Prentice-Hall.

[114] MONGOOSE commentary (2015). http://groups.csail.mit.edu/cb/mongoose/.

[115] Morris, M. (1991). Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174.

[116] Moss, L. (2008). Confusion of memory. *Information Processing Letters*, 107(3):114–119.

[117] Müller, M., Mentel, M., van Hellemond, J., Henze, K., Woehle, C., Gould, S., Yu, R., van der Giezen, M., Tielens, A., and Martin, W. (2012). Biochemistry and evolution of anaerobic energy metabolism in eukaryotes. *Microbiology and Molecular Biology Reviews*, 76(2):444–495.

[118] Nam, H. et al. (2014). A systems approach to predict oncometabolites via context-specific genome-scale metabolic networks. *PLoS computational biology*, 10(9):e1003837.

[119] Orth, J., Conrad, T., Na, J., Lerman, J., Nam, H., Feist, A., and Palsson, B. (2011). A comprehensive genome-scale reconstruction of Escherichia coli metabolism. *Molecular systems biology*, 7(1):535.

[120] Orth, J., Thiele, I., and Palsson, B. (2010). What is flux balance analysis? *Nature biotechnology*, 28(3):245–248.

[121] Ozbudak, E. M., Thattai, M., Kurtser, I., Grossman, A. D., and van Oudenaarden, A. (2002). Regulation of noise in the expression of a single gene. *Nature genetics*, 31(1):69–73.

[122] Pál, C., Papp, B., and Hurst, L. D. (2001). Highly expressed genes in yeast evolve slowly. *Genetics*, 158(2):927–931.

[123] Pál, C., Papp, B., and Lercher, M. J. (2005). Adaptive evolution of bacterial metabolic networks by horizontal gene transfer. *Nature Genetics*, 37(12):1372–1375.

[124] Palsson, B. Ø. (2015). *Systems Biology: Constraint-Based Reconstruction and Analysis*. Cambridge University Press.

[125] Paltanea, M., Tabirca, S., Scheiber, E., and Tangney, M. (2010). Logarithmic growth in biological processes. In *Computer Modelling and Simulation (UKSim), 2010 12th International Conference on*, pages 116–121. IEEE.

[126] Parker, G. A., Smith, J. M., et al. (1990). Optimality theory in evolutionary biology. *Nature*, 348(6296):27–33.

[127] Patil, K. R., Rocha, I., Förster, J., and Nielsen, J. (2005). Evolutionary programming as a platform for in silico metabolic engineering. *BMC bioinformatics*, 6(1):308.

[128] Peng, L. and Shimizu, K. (2003). Global metabolic regulation analysis for Escherichia coli k12 based on protein expression by 2-dimensional electrophoresis and enzyme activity measurement. *Applied Microbiology and Biotechnology*, 61(2):163–178.

[129] Pfleger, B. F., Pitera, D. J., Smolke, C. D., and Keasling, J. D. (2006). Combinatorial engineering of intergenic regions in operons tunes expression of multiple genes. *Nature biotechnology*, 24(8):1027–1032.

[130] Pharkya, P., Burgard, A. P., and Maranas, C. D. (2003). Exploring the overproduction of amino acids using the bilevel optimization framework optknock. *Biotechnology and bioengineering*, 84(7):887–899.

[131] Potera, C. (2005). Making succinate more successful. *Environmental health perspectives*, pages A833–A835.

[132] Raj, A. and van Oudenaarden, A. (2009). Single-molecule approaches to stochastic gene expression. *Annual review of biophysics*, 38:255.

[133] Reed, J. L., Vo, T. D., Schilling, C. H., and Palsson, B. Ø. (2003). An expanded genome-scale model of Escherichia coli K-12 (iJR904 GSM/GPR). *Genome Biology*, 4(9):R54.

[134] Retchless, A. C. and Lawrence, J. G. (2012). Ecological adaptation in bacteria: Speciation driven by codon selection. *Molecular Biology and Evolution*, 29(12):3669–3683.

[135] Roberts, N. J., Zhang, L., Janku, F., Collins, A., Bai, R.-Y., Staedtke, V., Rusk, A. W., Tung, D., Miller, M., Roix, J., Khanna, K. V., Murthy, R., Benjamin, R. S., Helgason, T., Szvalb, A. D., Bird, J. E., Roy-Chowdhuri, S., Zhang, H. H., Qiao, Y., Karim, B., McDaniel, J., Elpiner, A., Sahora, A., Lachowicz, J., Phillips, B., Turner, A., Klein, M. K., Post, G., Diaz, L. A., Riggins, G. J., Papadopoulos, N., Kinzler, K. W., Vogelstein, B., Bettegowda, C., Huso, D. L., Varterasian, M., Saha, S., and S, Z. (2014). Intratumoral injection of clostridium novyi-nt spores induces antitumor responses. *Science Translational Medicine*, 6:249ra111.

[136] Rocha, M., Maia, P., Mendes, R., Pinto, J. P., Ferreira, E. C., Nielsen, J., Patil, K. R., and Rocha, I. (2008). Natural computation meta-heuristics for the in silico optimization of microbial strains. *BMC bioinformatics*, 9(1):499.

[137] Romero, I. G., Ruvinsky, I., and Gilad, Y. (2012). Comparative studies of gene expression and the evolution of gene regulation. *Nature Reviews Genetics*, 13(7):505–516.

[138] Rossell, S., Huynen, M. A., and Notebaart, R. A. (2013). Inferring metabolic states in uncharacterized environments using gene-expression measurements. *PLoS Comput Biol*, 9(3):e1002988.

[139] Ryu, J., Kim, H. U., and Lee, S. Y. (2015). Reconstruction of genome-scale human metabolic models using omics data. *Integrative Biology*.

[140] Salis, H. M., Mirsky, E. A., and Voigt, C. A. (2009). Automated design of synthetic ribosome binding sites to control protein expression. *Nature biotechnology*, 27(10):946–950.

[141] Sanchez, A., Choubey, S., and Kondev, J. (2013). Regulation of noise in gene expression. *Annual review of biophysics*, 42:469–491.

[142] Schellenberger, J., Park, J. O., Conrad, T. M., and Palsson, B. Ø. (2010). Bigg: a biochemical genetic and genomic knowledgebase of large scale metabolic reconstructions. *BMC bioinformatics*, 11(1):213.

[143] Schellenberger, J., Que, R., Fleming, R. M., Thiele, I., Orth, J. D., Feist, A. M., Zielinski, D. C., Bordbar, A., Lewis, N. E., Rahmanian, S., et al. (2011). Quantitative prediction of cellular metabolism with constraint-based models: the cobra toolbox v2. 0. *Nature protocols*, 6(9):1290–1307.

[144] Schneider, R. E., Brown, M. T., Shiflett, A. M., Dyall, S. D., Hayes, R. D., Xie, Y., Loo, J. A., and Johnson, P. J. (2011). The trichomonas vaginalis hydrogenosome proteome is highly reduced relative to mitochondria, yet complex compared with mitosomes. *International journal for parasitology*, 41(13):1421–1434.

[145] Schuetz, R., Zamboni, N., Zampieri, M., Heinemann, M., and Sauer, U. (2012). Multidimensional optimality of microbial metabolism. *Science*, 336(6081):601–604.

[146] Shiflett, A. and Johnson, P. (2010). Mitochondrion-related organelles in parasitic eukaryotes. *Annual review of microbiology*, 64:409.

[147] Shimizu, K. (2004). Metabolic flux analysis based on 13c-labeling experiments and integration of the information with gene and protein expression patterns. In *Recent Progress of Biochemical and Biomedical Engineering in Japan Ii*, pages 1–49. Springer.

[148] Shinar, G., Alon, U., and Feinberg, M. (2009). Sensitivity and robustness in chemical reaction networks. *SIAM Journal on Applied Mathematics*, 69(4):977–998.

[149] Shoval, O., Sheftel, H., Shinar, G., Hart, Y., Ramote, O., Mayo, A., Dekel, E., Kavanagh, K., and Alon, U. (2012). Evolutionary trade-offs, pareto optimality, and the geometry of phenotype space. *Science*, 336(6085):1157–60.

[150] Siuti, P., Yazbek, J., and Lu, T. K. (2013). Synthetic circuits integrating logic and memory in living cells. *Nature biotechnology*, 31(5):448–452.

[151] Smith, A. and Robinson, A. (2011). A metabolic model of the mitochondrion and its use in modelling diseases of the tricarboxylic acid cycle. *BMC systems biology*, 5(1):102.

[152] Soloveichik, D., Cook, M., Winfree, E., and Bruck, J. (2008). Computation with finite stochastic chemical reaction networks. *Natural Computing*, 7(4):615–633.

[153] Sorokina, O., Corellou, F., Dauvillée, D., Sorokin, A., Goryanin, I., Ball, S., Bouget, F.-Y., and Millar, A. J. (2011). Microarray data can predict diurnal changes of starch content in the picoalga ostreococcus. *BMC systems biology*, 5(1):36.

[154] Sridhara, V., Meyer, A. G., Rai, P., Barrick, J. E., Ravikumar, P., Segrè, D., and Wilke, C. O. (2014). Predicting growth conditions from internal metabolic fluxes in an in-silico model of E. coli. *PloS one*, 9(12):e114608.

[155] Stanton, D. and White, D. (1986). *Constructive combinatorics*. Springer.

[156] Stoletzki, N. and Eyre-Walker, A. (2007). Synonymous codon usage in Escherichia coli: selection for translational accuracy. *Molecular biology and evolution*, 24(2):374–381.

[157] Stracquadanio, G., Umeton, R., Papini, A., Liò, P., and Nicosia, G. (2010). Analysis and optimization of c3 photosynthetic carbon metabolism. In *BioInformatics and BioEngineering (BIBE), 2010 IEEE International Conference on*, pages 44–51. IEEE.

[158] Takeuchi, R., Tamura, T., Nakayashiki, T., Tanaka, Y., Muto, A., Wanner, B. L., and Mori, H. (2014). Colony-live–a high-throughput method for measuring microbial colony growth kinetics–reveals diverse growth effects of gene knockouts in Escherichia coli. *BMC microbiology*, 14(1):171.

[159] Thiele, I., Swainston, N., Fleming, R. M., Hoppe, A., Sahoo, S., Aurich, M. K., Haraldsdottir, H., Mo, M. L., Rolfsson, O., Stobbe, M. D., et al. (2013). A community-driven global reconstruction of human metabolism. *Nature biotechnology*, 31(5):419–425.

[160] Tomita, M. (2001). Whole-cell simulation: a grand challenge of the 21st century. *Trends in biotechnology*, 19(6):205–210.

[161] Tomita, M., Hashimoto, K., Takahashi, K., Shimizu, T. S., Matsuzaki, Y., Miyoshi, F., Saito, K., Tanida, S., Yugi, K., Venter, J. C., et al. (1999). E-CELL: software environment for whole-cell simulation. *Bioinformatics*, 15(1):72–84.

[162] Turing, A. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72.

[163] Turing, A. M. (1936). On computable numbers, with an application to the entscheidungsproblem. *J. of Math*, 58(345-363):5.

[164] Van Der Giezen, M. (2009). Hydrogenosomes and mitosomes: Conservation and evolution of functions1. *Journal of Eukaryotic Microbiology*, 56(3):221–231.

[165] van Ooyen, J., Noack, S., Bott, M., Reth, A., and Eggeling, L. (2012). Improved l-lysine production with corynebacterium glutamicum and systemic insight into citrate synthase flux and activity. *Biotechnology and bioengineering*, 109(8):2070–2081.

[166] Vogel, C. and Marcotte, E. M. (2012). Insights into the regulation of protein abundance from proteomic and transcriptomic analyses. *Nature Reviews Genetics*, 13(4):227–232.

[167] von Kamp, A. and Klamt, S. (2014). Enumeration of smallest intervention strategies in genome-scale metabolic networks. *PLoS computational biology*, 10(1).

[168] Von Neumann, J., Burks, A. W., et al. (1966). Theory of self-reproducing automata.

[169] Wagner, A. (2000). Inferring lifestyle from gene expression patterns. *Molecular biology and evolution*, 17(12):1985–1987.

[170] Wang, H. H., Kim, H., Cong, L., Jeong, J., Bang, D., and Church, G. M. (2012). Genome-scale promoter engineering by coselection mage. *Nature methods*, 9(6):591–593.

[171] Weatheritt, R. J. and Babu, M. M. (2013). The hidden codes that shape protein evolution. *Science*, 342(6164):1325–1326.

[172] Weiss, R., Knight, T., and Sussman, G. (2001). Cellular computation and communication using engineered genetic regulatory networks. *Cellular computing*, pages 120–1.

[173] Win, M. N., Liang, J. C., and Smolke, C. D. (2009). Frameworks for programming biological function through rna parts and devices. *Chemistry & biology*, 16(3):298–310.

[174] Wintermute, E. H., Lieberman, T. D., and Silver, P. A. (2013). An objective function exploiting suboptimal solutions in metabolic networks. *BMC systems biology*, 7(1):98.

[175] World Health Organization (April 2014). *Antimicrobial resistance*. Fact sheet N. 194.

[176] Yen, J. Y., Tanniche, I., Fisher, A., Gillaspy, G., Bevan, D., Senger, R., Yen, J. Y., Tanniche, I., Fisher, A. K., Gillaspy, G. E., et al. (2015). Designing metabolic engineering strategies with genome-scale metabolic flux modeling. *Clinical Epidemiology*, 7:149–160.

[177] Yim, H., Haselbeck, R., Niu, W., Pujol-Baxley, C., Burgard, A., Boldt, J., Khandurina, J., Trawick, J. D., Osterhout, R. E., Stephen, R., et al. (2011). Metabolic engineering of Escherichia coli for direct production of 1, 4-butanediol. *Nature chemical biology*, 7(7):445–452.

[178] Young, F. W. (2013). *Multidimensional scaling: History, theory, and applications*. Psychology Press.

[179] Zakrzewski, P., Medema, M. H., Gevorgyan, A., Kierzek, A. M., Breitling, R., and Takano, E. (2012). Multimeteval: comparative and multi-objective analysis of genome-scale metabolic models. *PloS one*, 7(12):e51511.

[180] Zamboni, N., Fendt, S.-M., Rühl, M., and Sauer, U. (2009). 13c-based metabolic flux analysis. *Nature protocols*, 4(6):878–892.

[181] Zhang, H.-X., Dempsey, W. P., and Goutsias, J. (2009). Probabilistic sensitivity analysis of biochemical reaction systems. *Journal of Chemical Physics*, 131(9):Art–No.

[182] Zhuang, K., Ma, E., Lovley, D. R., and Mahadevan, R. (2012). The design of long-term effective uranium bioremediation strategy using a community metabolic model. *Biotechnology and bioengineering*, 109(10):2475–2483.

[183] Zomorrodi, A. R., Suthers, P. F., Ranganathan, S., and Maranas, C. D. (2012). Mathematical optimization applications in metabolic networks. *Metabolic engineering*, 14(6):672–686.

[184] Zur, H., Ruppin, E., and Shlomi, T. (2010). iMAT: an integrative metabolic analysis tool. *Bioinformatics*, 26(24):3140–3142.