Theoretical Computer Science 599 (2015) 102–117

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Analysis and design of molecular machines

C. Angione [a,*,1], J. Costanza [b,1], G. Carapezza [c], P. Lió [a,*], G. Nicosia [c,*]

[a] *Computer Laboratory, University of Cambridge, UK*
[b] *Italian Institute of Technology, Milan, Italy*
[c] *Department of Mathematics and Computer Science, University of Catania, Italy*

ARTICLE INFO

Article history:
Received 18 February 2013
Received in revised form 13 December 2014
Accepted 17 January 2015
Available online 22 January 2015

Keywords:
Pareto optimality
E. coli modelling
Turing machine
Molecular machine
Biological complexity
Petri nets
Register machines
Von Neumann architectures
Trade-off genetic strategies
Flux-balance analysis

ABSTRACT

Biologically inspired computation has been recently used with mathematical models towards the design of new synthetic organisms. In this work, we use Pareto optimality to optimize these organisms in a multi-objective fashion. We infer the best knockout strategies to perform specific tasks in bacteria, which involve concurrent maximization/minimization of multiple functions (codomain) and optimization of several decision variables (domain). Furthermore, we propose and exploit a mapping between the metabolism and a register machine. We show that optimized bacteria have computational capability and act as molecular Turing machines programmed using a Pareto optimal solution. Finally, we investigate communication between bacteria as a means to evaluate their computational capability. We report that the density and gradient of the Pareto curve are useful tools to compare models and understand their structure, while modelling organisms as computers proves useful to carry out computation using biological machines with specific input–output conditions, as well as to estimate the bacterial computational effort for specific tasks.

1. Introduction

In the last decades, computer science and mathematics have been widely used to understand the behavior of biological systems or to analyze high throughput data. There are different methods to represent a biological system, for instance by using a system of equations. In metabolic systems, each variable represents the variation of a metabolite concentration in a compartment, in a dynamic or steady state, where the metabolite concentration depends on material fluxes that enter or leave the compartment. Each flux can be also modelled by using kinetics parameters. Usually, these systems contain a large number of equations (differential or algebraic) and solving the problem analytically is often very hard, leading to the increasing use of numerical methods. Additionally, the advent of high throughput data in medicine requires computational techniques for data mining. Biologically inspired computation has been used to infer mathematical models, parameter values, or to capture states and transitions at the molecular level [1].

Metabolic engineering consists of optimizing genetic and regulatory processes within cells to increase the cell production of certain substances. The in silico analysis is the first step for designing a new synthetic organism. *Escherichia coli* is

[*] Principal corresponding authors.
E-mail addresses: claudio.angione@cl.cam.ac.uk (C. Angione), jole.costanza@iit.it (J. Costanza), carapezza@dmi.unict.it (G. Carapezza), pl219@cam.ac.uk (P. Lió), nicosia@dmi.unict.it (G. Nicosia).
URLs: http://www.cl.cam.ac.uk/~pl219/ (P. Lió), http://www.dmi.unict.it/nicosia (G. Nicosia).
[1] These authors contributed equally to this work.

http://dx.doi.org/10.1016/j.tcs.2015.01.030
0304-3975/© 2015 Elsevier B.V. All rights reserved.

one of the most studied organisms in biology, as well as in synthetic biology and metabolic engineering. Researchers and biotechnologists focused their efforts on the study of its metabolic network, since it is simple and its strain is easy to manipulate in laboratory. In particular, in the last ten years, Palsson and colleagues have published several works about the *E. coli* network and its modelling. In 2000, Edwards and Palsson [2] published the first genome-scale metabolic network of the K12-MG1655 *E. coli*, composed of 627 reactions, 438 metabolites and 660 genes. By considering the steady-state, the flux balance analysis (FBA) is used to solve the system with a linear programming approach, obtaining the flux distribution in the metabolic network. The fluxes distribution depends on the environment, for instance the glucose feed or the presence of oxygen, and on genetic manipulations (knockouts). More recently, the genome-scale reconstruction of this organism has been augmented to include 904 genes, 625 metabolites and 931 reactions [3], then 1260 genes, 1039 metabolites and 2077 reactions [4] and finally 1366 genes, 1136 metabolites and 2251 reactions [5]. FBA is a useful framework in that it allows to understand the behavior of large networks and perform the knockout analysis at a low computational cost [6]. FBA-based approaches reveal more efficient than other mathematical modelling techniques, such as those based on ordinary differential equations [7], at tackling genome-scale metabolic networks, which have been extensively used to characterize energy production in cells [8], and to design synthetic pathways in silico (e.g. for production of biofuels [9]).

In this research work, we analyze two genome-scale metabolic networks of *E. coli*. By using a multi-objective optimization method called Genetic Design by Multi-objective Optimization (GDMO) [6], we maximize several pairs of biological functions, such as acetate production and biomass formation. The biomass reaction is scaled so that the flux through it is equal to the exponential growth rate of the organism. In the optimization procedure, we search for the best genetic strategies that maximize the selected objectives. The results are represented in the Pareto curve. The area under the curve, 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 compare different models or different organisms.

Further, 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 Boolean string found by GDMO represents the optimal genetic knockout strategy and also the 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.

The remainder of the article is organized as follows. In Section 2, we review Pareto optimality and the main ideas underlying GDMO [6]. Then, we report a comparison between two different models used to represent the *E. coli* network. In the FBA framework, we adopt the GDMO algorithm to optimize genetic manipulations in order to maximize several biological functions. We also perform the sensitivity and robustness analyses [10] for the two models, and rank nutrient metabolites according to their influence on the output (the distribution of fluxes). Additionally, based on Pareto genetic strategies, we infer neutral, trade off and destructive manipulations. In Section 3, we introduce a relation between computation and metabolism explained through a formalism that associates the structure of any bacterium with a von Neumann architecture. In Section 4, we discuss this mapping thinking of the metabolism as a Minsky register machine with universal computational capability. In Sections 5–6, we discuss the effect that various events (e.g. motility, communication, gene duplication) may have on the computation performed by a bacterium. We also remark the changes occurring in the computation capability as a consequence of a duplication event followed by a mutation.

2. Optimization of gene sets

A Pareto front is the set of points in a given objective space such that there does not exist any other point that dominates them in all the objectives. It is obtained as a result of a multi-objective optimization technique needed when a system (a given phenotype) cannot be optimal at all the tasks it performs, and particularly when tasks are in contrast with each other [11]. For instance, given the task of optimizing an organism, the Pareto front allows to maximize or minimize two or more target metabolites, thus obtaining new optimal strains specialized in many aims simultaneously.

Formally, given r objective functions $f_1, \dots, f_r$ to maximize/minimize, the problem of optimizing in a multi-objective fashion can be rephrased as the problem of finding a vector x^* that satisfies all the constraints and optimizes the vector function $f(x) = (f_1(x), f_2(x), \dots, f_r(x))^\mathsf{T}$, where x is the variable (vector) to be optimized in the search space. Without loss of generality, in the definition all the functions are maximized (however, minimizing a function f_i is equivalent to maximizing $-f_i$). The output of a multi-objective routine is a set of Pareto optimal points, which constitute the *Pareto front*. A solution x^* in the search space X is *Pareto optimal* if $\nexists x \in X$ such that $f(x)$ *dominates* $f(x^*)$, or more formally if

$$\nexists x \in X \quad \text{s.t.} \quad f_i(x) > f_i\big(x^*\big), \quad \forall i = 1, \dots, r, \tag{1}$$

where f is the vector of r objective functions that have to be maximized in the objective space. Since the multiple targets f_i are usually in conflict with each other, the term *optimizing* means finding all the solutions that represent a trade-off for the designer.

The many-objective Pareto optimality is a useful and powerful tool to understand the phenotype of organisms in different environmental conditions and genetic strategies. By adopting a trade-off strategy, an organism is able to maximize/minimize simultaneously several biotechnological targets, e.g. the output of the computation it carries out. In the Pareto fronts and