Intelligent Data Analysis 14 (2010) 1–19
DOI 10.3233/IDA-2010-0415
IOS Press

From digital genetics to knowledge discovery: Perspectives in genetic network understanding

Guillaume Beslon[a,c,*,1], David P. Parsons[a,c,1], Jose-Maria Peña[b], Christophe Rigotti[a,c,1] and Yolanda Sanchez-Dehesa[a,c,1]
[a]*Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France*
[b]*DATSI, Universidad Politecnica de Madrid, Spain*
[c]*IXXI, Institut Rhône-Alpin des Systèmes Complexes, Lyon, F-69007, France*

Abstract. In this paper, we propose an original computational approach to assist knowledge discovery in complex biological networks. First, we present an integrated model of the evolution of regulation networks that can be used to uncover organization principles of such networks. Then, we propose to use the results of our model as a benchmark for knowledge discovery algorithms. We describe a first experiment of such benchmarking by using gene knock-out data generated from the modeled organisms.

1. Introduction

Since Jacques Monod and François Jacob discovered the lactose operon in *Escherichia coli*, genetic regulation has been recognized as one of the main control centers in cells and organisms. Influenced by the *lac* operon, biologists have long considered gene networks as sets of simple independent “reflex” regulations. However, it is now recognized that these networks are very complex and tightly linked, putting all the activities and functions in interaction with one another. A great deal of work have shown that the structure of regulation networks is far from being random, showing regularities at all scales from small motifs [4,12] to global connectivity patterns [6,43]. The understanding of these regularities challenge modern biology and two fundamental questions have emerged during the last decade: What is the origin of such complex connectivity patterns and how can we decipher the emergent dynamic behavior of such networks? In fact, these questions are tightly related since a network’s behavior is selected for by the Darwinian process that created the network itself.

The question of the behavior of regulation networks is at the heart of the so-called “systems biology” approach that has emerged ten years ago [17,21,22]. Given the complex connectivity patterns of biological networks, the objective of systems biology is to understand the global behavior of these networks. On the one hand, this global challenge can be decomposed into questions like “Can we understand the

[*] Corresponding author. E-mail: guillaume.beslon@liris.cnrs.fr.
[1] On strike while writing this article to protest against French Government’s projects on research management.

behavior of specific organisms under some specific conditions?" (e.g., carbon starvation response in *E. coli* [36]). On the other hand, one can raise the question of general principles of genetic regulation: Does the complex gene network structure correspond to a particular – i.e., integrated – way of controlling the metabolic activity of biological organisms?

Various hypotheses have been proposed to explain the structure of these networks in terms of evolutionary forces: Mutational patterns (i.e., neutral evolution), direct selective forces or indirect selection for robustness or evolvability. To cite but a few, it was proposed (*i.*) that the power-law connectivity of regulation networks could be a consequence of gene duplication-divergence [28], (*ii.*) that the modularity of regulation networks could be selected for because it allows fast adaptation to variations of the environment [20] or (*iii.*) that overrepresented Feed-Forward Loop motifs are selected for because they allow a fine tuning of the response delays in regulation networks [4].

Every engineer knows how to design large systems by building a hierarchy of modules in order to reduce the inter-dependence between modules and to rationalize the system's conception. Yet the modular organization of our engineered systems is a consequence of our intellectual limits as system designers. But evolution is *not* an engineer. It acts by trial and error, keeping the structures that are most effective, whatever their complexity and intricacy, ignoring our attempts to understand its product. Systems biology is often considered as a reverse engineering process applied to biological entities: Given the observations of the behavior of the biological system (and possibly, its response to man-made perturbations [18]), biologists are to decipher the organizational principles of the system. However, in the case of reverse engineering, it can be assumed that the observed system was conceived by an engineer, following reasonable conception rules and reasoning. In the case of biological systems, we cannot suppose evolution to be rational at all (at least in our common – i.e., human – sense). Yet, this does not mean that there are no rules in evolution: Under some specific conditions (e.g., cyclic environments) it has been shown that evolution can behave *as* an engineer and produce organized systems [2,20]. The existence of other general laws that could govern the organization of biological networks depending on external conditions is an open question. Our grand challenge is hence to identify the "language" that evolution has created for regulation networks and how it can be translated from a structural description (i.e., the set of weighted links, motifs and modules) to a functional description (the cell behavior) [41].

These questions are very difficult to tackle with real organisms, either because they require long and complex experimental setups or because results are difficult to analyze given the little knowledge available. In particular, it is difficult to trace changes in genomes and to identify selected traits in real organisms. A possible solution is then to develop in silico models of evolution – i.e., digital genetic models [1] – and to test how, within these models, organisms evolve depending on the environmental conditions. Such models have already shown to be interesting to study how organisms have their structure and complexity adapted, resulting in an increased robustness [40]. As far as regulation networks are concerned, computational evolution has been used to investigate the evolvability of networks [9] or the development of modular structures under cyclic environmental conditions [20]. One of the best-known models – namely the GRN model – has been proposed by Wolgang Banzhaf [5] and used to investigate the emergence of specific topological properties in regulatory networks [28]. More recently, Claudio Mattiussi and Dario Floreano have proposed the "Analog Genetic Encoding" framework [32] which has been latter on used to investigate the modular structure of regulation networks [31]. Other authors have used computational evolution in order to evolve small networks performing predefined functions such as oscillators [12,29], clocks [23] or switches [12].

In this paper, we propose an original computational approach to assist the understanding of complex biological networks. First, we describe the RAevol model, an integrated model of the evolution of

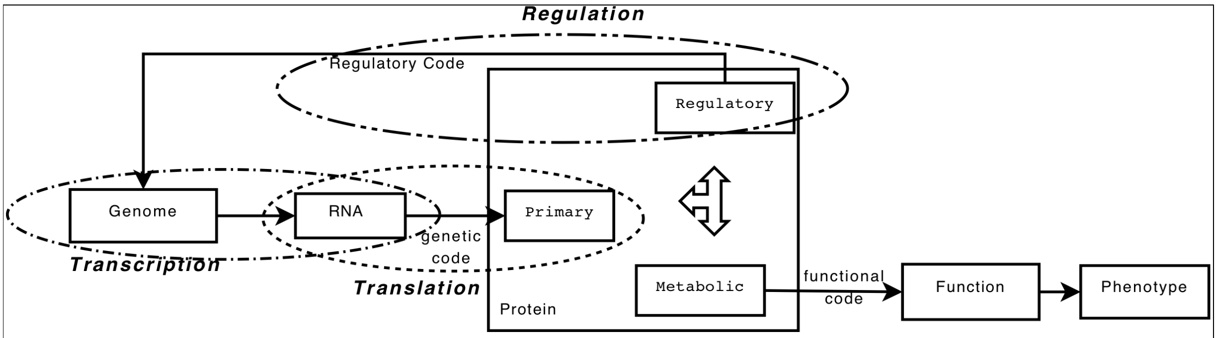

Fig. 1. Overview of the central position of proteins in the model. Proteins play different roles depending on the elements they interact with. They are translated from the genome (actually from mRNA), they can carry out a phenotypic function and they can regulate transcription activity.

regulation networks (Section 2) that can be used to uncover organization principles of such networks. Then, going from modeling to data-mining tools, we propose to use the results of our model as a benchmark in order to experiment with knowledge discovery algorithms. We present a first experiment of such benchmarking by using gene knock-out data generated from our *in silico* organisms (Section 3).

2. The RAevol model

To study the evolution of the structure of gene networks, we have developed an integrated model, RAevol (Regulatory Aevol), that extends the “Artificial Evolution” (Aevol) model previously developed in our team to study robustness and evolvability in artificial organisms [24]. In Aevol and RAevol, the structure of the genome is inspired by prokaryote DNA structure: It is organized as a circular double-strand binary string with a variable number of genes separated by non-coding sequences. The genetic sequence is translated into a set of “proteins” that interact with one another to produce a phenotype that can be more or less adapted to its environment. Although Aevol does not include any regulatory process (a gene is transcribed at a constant rate, depending only on the sequence of its promoter), it includes all the organization levels needed to build an integrated model of genetic regulation. That’s the reason why we used it as a basis for the development of the RAevol model.

2.1. Artificial chemistry in RAevol

In a model like RAevol, a complete realistic biochemichal description of organisms is obviously impossible. We hence define an “artificial chemistry” [10] that describes the organism metabolism in a mathematical language. In Raevol, this artificial chemistry is centered on the protein level[1] (Fig. 1).

Basically, proteins are elements able to perform metabolic functions: In the model we defined an abstract space of metabolic functions Ω wherein the proteins will act with a certain degree of efficiency (a real number within the interval [0, 1]). Thus, each protein P_i is described as a parametric function $f_i : \Omega \rightarrow [-1, 1]$ that can activate (if positive) or inhibit (if negative) a subset of the metabolic space with a particular efficiency $|f_i|$. The parameters of the function f_i are computed from the genetic sequence of the protein through a transcription-translation-folding process: Transcribed sequences are identified on

[1] A complete description of the model can be found in [37].

the genome by consensus promoter sequences (translation initiation) and ρ-independent-like structures (transcription termination). These sequences are transcribed into RNA sequences. RNAs are then parsed to extract the genes (that are detected thanks to a shine-dalgarno-like sequence and a START triplet, a STOP triplet being used to detect the end of the gene). These genes are then translated into the primary sequence of the corresponding protein thanks to an artificial *genetic code* which associates triplets of bits (i.e., codons) with artificial Amino-Acids. Since our genetic sequence is composed of binary "bases", we have $2^3 = 8$ different codons among which two must be reserved for the START and STOP codons. In RAevol, the metabolic activity of proteins is represented by simple symmetric triangle-shaped functions. Thus any function f_i can be fully defined by three parameters: The mean (M_i which characterizes the main metabolic function of the protein), the width (W_i which characterizes the pleiotropy of the protein) and the height (H_i which indicates the efficiency of the protein in performing its metabolic function). An artificial folding process converts each protein primary sequence (amino-acid sequence) into a specific set of values for these three parameters, thus characterizing the function of the protein. The conversion is achieved by using three interlaced variable-length binary codes (one per parameter).

However, the efficiency of a protein P_i depends not only on its primary sequence but also on its concentration in the organism at each time step t: $c_i(t)$. Thus, the metabolic activity of a given protein is also time-dependent, meaning that the metabolic functions performed by this protein will change over time. The Amino-Acid sequence of the protein is used to compute its intrinsic activity $h_i \in [-1, 1]$ (i.e., its activity when at a concentration of 1; all concentrations are expressed in arbitrary units). Then the actual activity of the protein in the organism at a specific time t is given by $H_i(t) = |h_i|.c_i(t)$, where the concentration of the protein P_i is computed through a synthesis-degradation process with a constant degradation rate ϕ and a promoter-dependent synthesis rate (see below).

Once all the proteins encoded on the genotype of the organism have been identified and characterized, the global phenotype can be computed by combining the whole set of proteins. This phenotype is the result of fuzzy set operations performed on two protein subsets: Those which activate metabolic functions (i.e., $h_i > 0$) and those which inhibit metabolic functions (i.e., $h_i < 0$). The global metabolic capabilities of an organism are the functions which are activated by a protein of this organism and not inhibited by another protein (all the fuzzy-set operations are performed using the Lukasiewicz fuzzy operators):

$$F = (\cup_i(P_i|h_i > 0)) \cap \overline{(\cup_j(P_j|h_j < 0))} \tag{1}$$

At this stage, the model fully conforms to the so-called "central dogma of molecular biology": The information flows linearly from DNA to the phenotype through the RNA and protein levels. However, with the work of Monod and Jacob in the early 60's, things turn out to be far more complex. In particular, it is now well known that proteins can regulate the RNA transcription activity, thus constituting the *genetic network* that governs the dynamic of the cell. In prokaryotic organisms, the regulatory activity of a protein is mainly caused by its direct interactions with the DNA strands. If it is able to bind to a promoter region on the DNA, it may have a regulatory activity. In RAevol, we modeled this process by introducing a new module in our artificial chemistry: The possibility of a given protein to bind to a specific promoter is determined by a "value of affinity" between the primary sequence of the former and the genetic sequence of the latter. Small amino-acid sequences, that will henceforth be refered to as regulation domains, are able to bind to specific DNA subsequences with a given affinity. If a protein contains several regulation domains, its global affinity value over the promoter will be given by the strongest. This value of affinity is used to determine the strength of the protein's regulation of the transcriptional activity of the promoter it binds to. Like in most bacterial promoters, the nature of

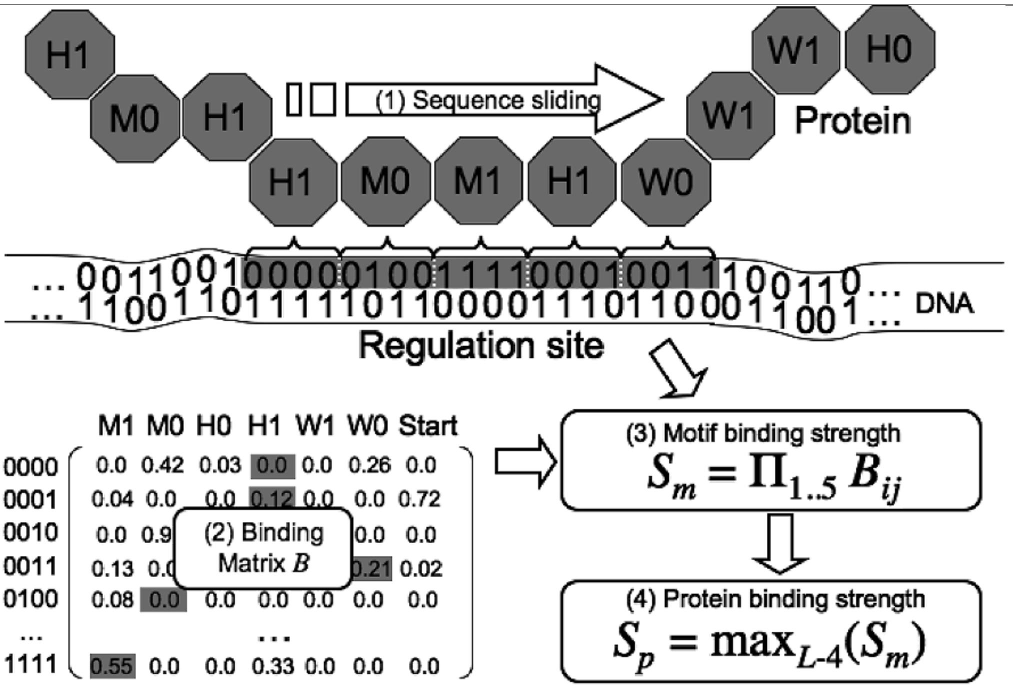

Fig. 2. Computation of the binding between TFs and regulation sites. (1) The protein primary sequence slides in front of the 20-bp regulation site and all 5-AA-long motifs are tested. (2) For each AA-subsequence pair, the binding value B_{ij} is read in a binding matrix B (see main text for the initialization of the binding matrix). (3) The binding strength, S_m, of the whole motif is the product of the five B_{ij} values and (4) the binding strength, S_p, of the whole protein is the maximum strength over the $L - 4$ motifs it contains (L being the length of the proteins primary sequence). The regulatory activity of the protein p then depends on the global strength value S_p: For a given promoter i, a protein p is aligned both upstream and downstream of the promoter resulting in two different S_p values (S_{p_up} and S_{p_down}). The upstream alignment enables us to compute the activation abilities of the protein ($A_{pi} = S_{p\_up}$) while the downstream alignment gives its inhibition abilities ($I_{pi} = S_{p\_down}$).

the regulation (activation or inhibition) depends on whether the binding occurs before (upstream) or after (downstream) the position of the first transcribed nucleotide [19]. Thus, in RAevol, a promoter is composed of three DNA subsequences: The consensus sequence (where the RNA polymerase starts the transcription process) and its two flanking regions. When bound upstream from the promoter (on its *enhancer* site), a protein enhances the transcriptional activity; on the opposite, when bound downstream (on the *operator*), it represses the activity of the polymerase, thus reducing the transcriptional activity.

The sequences that are able to bind to a specific DNA subsequence (thus constituting the possible regulation domains) are randomly determined at the beginning of the evolutionary run (through a random regulatory matrix that associates Amino-Acid subsequences with nucleotide subsequences). In RAevol, regulation domains are small 5-Amino-Acid (AA) sequences that may have an affinity with 20 base pairs (bp) DNA sequences. To compute this affinity value, we align the regulation domain with the DNA sequence and compute the local affinity of each AA with the 4-bp subsequence it faces (Fig. 2). The motif will be able to bind to the DNA sequence only if all five AA have strictly positive affinities with their corresponding DNA subsequences. Given our artificial chemistry principles, we have 7 possible Amino-Acids (START, M_0, M_1, H_0, H_1, W_0 and W_1) and $2^4 = 16$ 4-bp sequences. A 7×16 binding matrix B is defined which contains the affinity of each amino-acid with each 4-bp sequence. By choosing the initialization procedure of the regulatory matrix, we are able to choose the probability for a given motif to have a regulation activity. In the experiment presented here, B was randomly initialized (uniform distribution in [0, 1]) and subsequently filled with 75% of null values. Thus, the probability that a given motif will bind to a specific binding site is less than 0.1%. As a consequence, the probability that a 20-AA-long protein will be able to regulate a given promoter can be estimated at around 5% (probability to contain a motif that binds to either the enhancer or the operator flanking the promoter).

The overall activity of a given promoter i thus depends on: (i) the quality of its consensus sequence (measured as the hamming distance d with the consensus sequence) which gives the transcriptional ground state β_i (or "basal transcription level", see Eq. (2)), (ii) the sum of the activities of activators A_i Eq. (3) and (iii) the sum of the activities of repressors Eq. (4) [38].

$$\beta_i = 1 - \frac{d_i}{d_{\max} + 1} \tag{2}$$

where d_i is the hamming distance between the promoter i and the consensus sequence and $d_{\,max}$ is the maximum admissible hamming distance for the RNA polymerase to recognize the promoter sequence.

$$A_i(t) = \sum_j c_j(t) A_{ji} \tag{3}$$

$$I_i(t) = \sum_j c_j(t) I_{ji} \tag{4}$$

where A_{ji} (resp. I_{ji}) is the affinity of protein j on the enhancer (resp. operator) of the promoter i and $c_j(t)$ is the concentration of protein j at time t.

The general equation that describes the transcription rate over time is then given by a Hill-like function:

$$e_i(t) = \beta_i \cdot \left(\frac{\theta^n}{I_i(t)^n + \theta^n} \right) \left(1 + \left(\frac{1}{\beta_i} - 1 \right) \left(\frac{A_i(t)^n}{A_i(t)^n + \theta^n} \right) \right) \tag{5}$$

where n and θ are constant coefficients that determine the shape of the Hill-function (in the simulations presented below $d_{max} = 4$, $n = 4$ and $\theta = 0.5$).

Finally, given the transcription rate, one can compute the protein concentration (for the sake of simplicity, we assume here that the protein concentration is linearly proportional to the RNA concentration) through a synthesis-degradation rule (with a degradation rate ϕ, Eq. (6)). Thus, when a protein is regulated, its concentration is scaled up or down depending on its transcription rate. Note that this also implies that the phenotype of the organisms is now a function of time.

$$\frac{\partial c_i}{\partial t} = e_i(t) - \phi c_i(t) \tag{6}$$

The transcription activity in RAevol is of course a huge simplification of the real mechanisms of DNA-protein interactions. However, it catches the main mechanisms of genetic regulation while remaining computationally tractable, thus enabling us to study the evolution of regulation in artificial – digital – organisms. It allows for a lots of different situations that are indeed observed in real bacteria: Operon structures (more than one gene on a single transcribed sequence), overlapping genes (either on the same strand or not), gene clustering along the DNA ... It also allows for proteins that perform a metabolic activity without any regulatory activity or, on the opposite, for proteins without any metabolic activity (i.e., $\int_0^1 |f(x)| = 0$) that can nevertheless have a regulatory activity. These proteins are called "Transcription Factors" (TFs). Eventually, genes and proteins can form a *genetic network* of variable complexity.

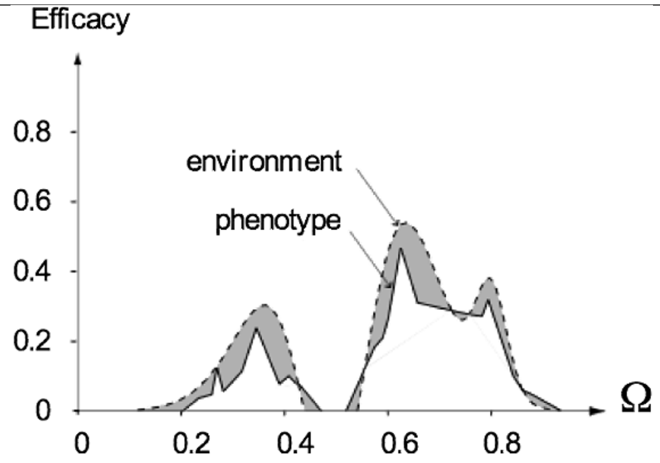

Fig. 3. Measure of an individual adaptation. Dashed curve: Environmental distribution E. Solid curve: Phenotypic distributio F (resulting function profile obtained after combining all proteins). Filled area: Metabolic error g. In RAevol, all distributions may change over time and the metabolic error can be computed in different ways (see main text for details).

2.2. *Evolutionary process*

Thanks to our artificial chemistry, we are able to map a genotype to a phenotype, the former being a circular double-strand binary sequence and the latter being a dynamic function $F(t)$ which expresses, at each time step t, the metabolism of the organism in the abstract functional space Ω. This enables us to evaluate the organism and to determine its “metabolic error” g in a given environment: The environment is described as a set of functions that have to be fulfilled by the cell in order to be able to reproduce (the lower the metabolic error, the higher the reproduction probability). The metabolic error is computed as the gap between both functions (Fig. 3).

Since the phenotype is a dynamic function, the environment may also be a dynamic function $E(t)$. Depending on the experiment one wants to do with the model, the metabolic error can be computed only once (e.g., after a transient period), at regular steps (e.g., each ten time steps), during a time interval (e.g., between $t = 10$ and $t = 20$) or after a particular environmental event. In this last case, the environment function is changed regularly or randomly and the changes are characterized by a signaling molecule introduced into the “cell” (this molecule is a manually-designed amino-acid chain. It can have a metabolic function or not, but it must contain a regulation domain in order to be able to interact with the regulation network). After a short transient period, the metabolic error is computed by comparing the phenotype to the new environmental function. In such experiment, the cell must evolve a regulatory network in order to react to the environmental perturbation. We are then able to analyze the structure of the network depending on the environmental complexity and on the evolutionary parameters. Figure 4 shows a typical phenotype evolved under such conditions.

In RAevol, the evolutionary process is based on a reproduction loop: A population of genomes (typically 1000 individuals) is initially created at generation 0. At each generation, each genome is translated into a phenotype which is evaluated. Then, a selection procedure and a biased random process (based either on the metabolic error g of the individuals, on their rank in the population ordered by decreasing g values or on the exponential of this rank) enable us to determine the number of offsprings each individual will have in the next generation. Then, all the organisms reproduce to create the next generation. In the current version of RAevol, the population size is constant and the population is completely renewed at each generation (generational algorithm).

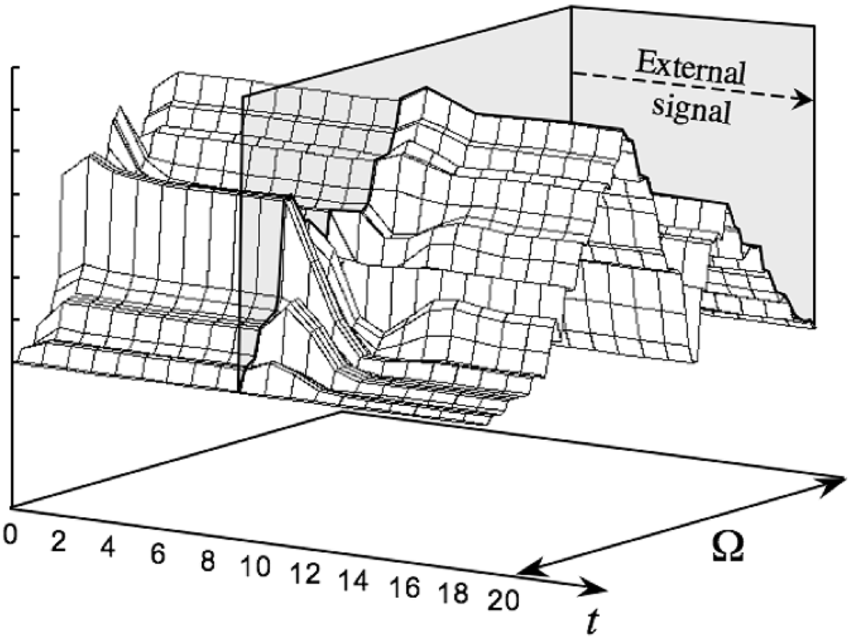

Fig. 4. At each time step t the phenotype is expressed as the efficiency of the organism in performing the metabolic functions in the abstract set Ω. Here, at time $t = 10$, an external signal is sent to the organism which reacts by modifying its metabolic profile.

The reproduction process itself is based upon error inducing DNA replication. These errors (mutations) are governed by operators that are directly inspired by prokaryotic reproduction. Genomes can undergo seven different types of mutations, the first three being point mutations and the four others, large chromosomic rearrangements:

Switch One base of the genome is randomly chosen and switched from 0 to 1 or vice versa.

InDel *In*sertion and *Del*etion are two different mutation operators. A short sequence – 1 to 6 bases – is inserted (repect. deleted) at a random position in the genome.

Translocation A randomly chosen segment of the genome is moved from its current position to a randomly chosen position (“cut-and-paste”).

Inversion A randomly chosen segment is inverted from one strand to the other and from one direction to the opposite one.

Duplication A randomly chosen segment is duplicated and reinserted at a randomly chosen position (“copy-paste”).

Deletion A randomly chosen segment is deleted from the genome.

The rate at which mutations occur, formally u (probability of mutation per base pair), is a parameter of the model, to be chosen for each particular simulation. It can be uniform (for all type of mutations) or it can be chosen specifically for each one of them. Mutations affect the genome but can be neutral. They can change the size of genome, the number of genes or the function of the proteins. Indirectly, they can modify the regulatory network’s topology by either duplicating/deleting genes or promoter regions. Finally, they can modify the affinities between transcription factors and regulatory regions by changing either the promoter sequences or the regulation domain in the proteins’ primary sequence.

2.3. *In silico experimental evolution with RAevol*

The typical use of digital evolutionary models is quite close to experimental evolution procedures [11]: Populations of organisms are initialized and left to evolve in controlled conditions (i.e., controlled parameters). By observing the products of the evolutionary process in different conditions and by comparing them, we can unravel the direct or indirect pressures that constrain the structure of organisms.

Using first Aevol, we had performed numerous *in silico* evolutionary experiments that had enabled us to uncover an indirect mechanism that regulates the balance between robustness and variability in evolving organisms [30]. In particular, we had been able to show that the amount of non-coding DNA that accumulate in a genome is inversely proportional (in a log-log space) to the mutation rate. This is the consequence of an indirect pressure exerted by the mutational burden brought by chromosomic rearrangement on non-coding sequences [25–27]. More recently, we also tested the effect of the mutation rate on the RAevol model and found that not only is the genetic sequence under the influence of the mutation rate but that the regulation network is also under the same pressure: Both the number of nodes and the overall connectivity of the regulation network clearly scale down when the mutation rate increases [7,8]. Moreover, as far as the number of nodes is concerned, this scaling follows a power-law distribution, whose exponent is larger for Transcription Factors (i.e., proteins that have a regulation activity in the network) than for metabolic elements (i.e., proteins that directly contribute to the metabolism of the organism). This is very similar to what is observed in comparative genomics studies among the prokaryotic kingdom [33,39].

Eventually, our main objective is to use RAevol to understand how the regulation network evolves depending on environmental conditions. Thus, while all these previous experimental setups were using static environments, we tried to evolve digital organisms in a more demanding dynamic environment. In the experiments presented here, organisms must adapt to a variable environment: During the first ten time steps, the ideal metabolic function is the one displayed on Fig. 3 (a three lobe function). Then, during the ten following steps, the rightmost lobe was removed. The metabolic error g is then computed as the mean between the metabolic error at $t = 10$ and the one at $t = 20$. When the environment is changed, a signaling protein is sent to the cell to trigger the change in the activity of the genetic regulation network. The organism must thus evolve a regulation network with the appropriate dynamic in order to react efficiently to this signal.

We let 3 different populations of 1000 organisms evolve for 40 000 generations under medium mutation rates ($u = 1.10^{-5}$ per mutation type per base pair) and a mild selective pressure (later on, experiments are labeled S1, S2 and S3). During the evolutionary process, the organisms progressively acquire new genes and connect them in order to fulfill the regulation task they are selected for (Fig. 5). All the simulations proceed qualitatively in a similar way, evolving quickly in the first stage of evolution (rapid gene recruitment) then slowing down the process of gene acquisition while optimizing gene and promoter sequences. At the end of the evolutionary process, we discovered that the organisms had acquired some specific proteins that contribute to the regulation process while having no metabolic activity. We call these proteins pure Transcription Factors (TFs).

After 40 000 generations, we focus on the best individual of each experiment. All three experiments show complex regulation networks with 51 (S1), 34 (S2) and 58 (S3) genes respectively. These genes are connected by hundreds of links, but while the number of genes is only slightly variable from one simulation to the other, the number of connections strongly differs: Network S1 has 328 connections, network S2, 153 and S3, 908. Note that there is no direct correlation between the size of the network and the metabolic error: S2, being the smallest network, is better than S1 which is much larger (Fig. 5).

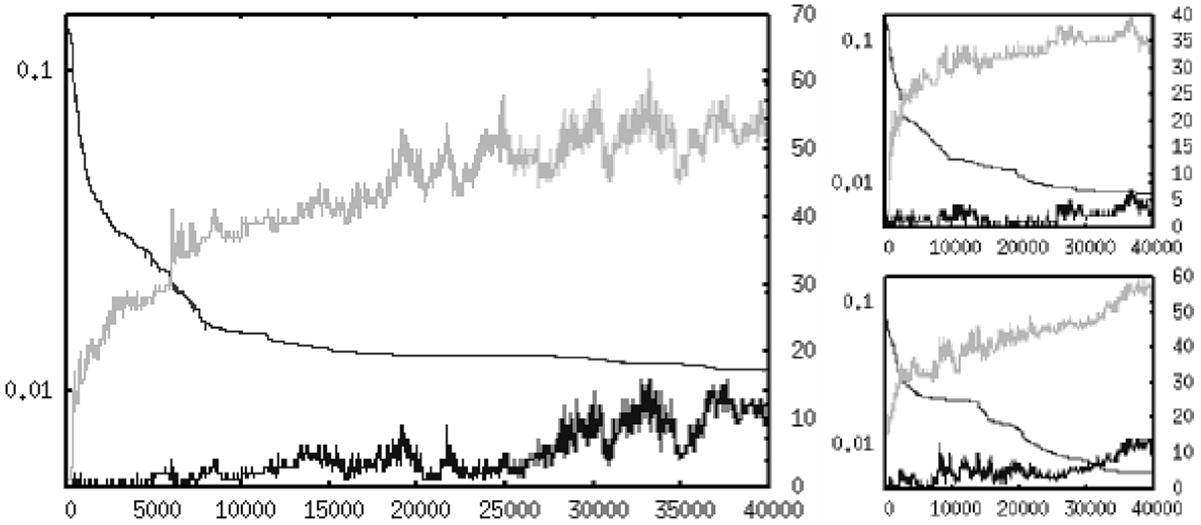

Fig. 5. Evolution of the organisms during 40 000 generations. Left: Simulation S1. Right: Simulations S2 (top) and S3 (bottom). Left axis: Metabolic error (black decreasing line). Right axis: Number of genes (soft grey) and genes which do not contribute to the metabolic activity (dark grey). The increasing number of non-metabolic genes in the second stage of the evolutionary process is characteristic of the recruitment of pure Transcription Factors (TFs).

Figure 6 (squared part) shows the variation of protein concentration during the lifetime of the best final individual in S1. One can easily see that, at time $t = 10$ (i.e., when the signaling protein is provided to the "cell") and $t = 20$ (i.e., when this signal is switched off), the protein concentrations quickly change to stabilize on new values. When looking at the phenotype (Fig. 7, squared part), we see that the signaling protein triggers a reorganization of the phenotype, one lobe of the metabolic function vanishing from $t = 10$ to $t = 20$.

When looking at the genetic networks created by evolution (Fig. 8), it clearly appears that the dynamic behavior of the network is impossible to decipher directly from the network structure. Thus, when analyzing the results of our experiments, we face the same problem as practitioners do with real biological networks: We need automatic mining algorithms to help us understand the structure of the networks and link it to their dynamic behaviors. Going one step beyond, we decided to perform, with our *in silico* organisms, the same experiments biologists do with real bacteria: We generated mutant variants in which single genes are invalidated one at a time (KO-mutants, Section 3.1) and measured their transcriptional activity. The resulting dynamic data can then be analyzed to understand the structure of the regulation network (Sections 3.2 and 4). Following this idea, we propose to use them as benchmarks, available for the scientific community to test knowledge discovery algorithms.[2]

3. Mining digital data

In this section, we present a simple data mining method we developed and evaluated on the artificial data generated by RAevol. The main purpose of this method is to help the expert get some insights about cell regulation processes. Using protein concentration data coming from realistic in silico models

[2] All data are available on Internet: http://liris.cnrs.fr/guillaume.beslon/IDAj.data/.

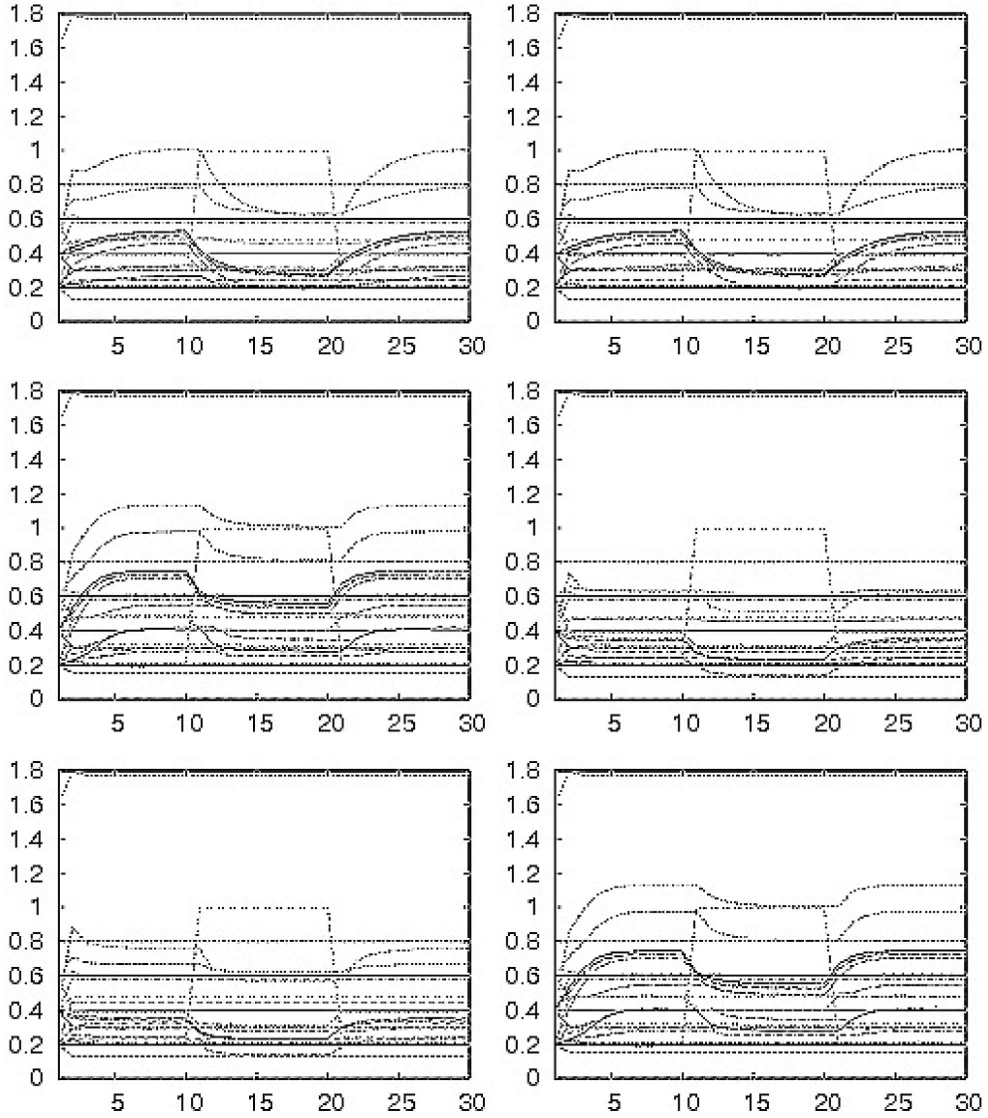

Fig. 6. Variations of protein concentrations over time in the wild-type organism (squared) and in five KO mutants: From top to bottom and left to right: Wild type, KO of gene 1, KO of gene 17, KO of gene 19, KO of gene 20 and KO of gene 34. x axis: Organism's live time (time steps). y axis: Protein concentrations (arbitrary units). Displayed mutants have been chosen because the show clear differences when compared to the wild-type.

of regulation networks, we can quickly collect some evidence about the possible effectiveness of the method. Such an evaluation aims to help us decide if the method is sufficiently promising to undertake further developments and more complex applications on real data.

3.1. Gene knock-outs in digital models

Gene Knock-Out (KO) [13] is a widely used technique in molecular biology. It provides geneticists with an insight into complex mechanisms, focusing on the contribution of a particular gene or set of genes.

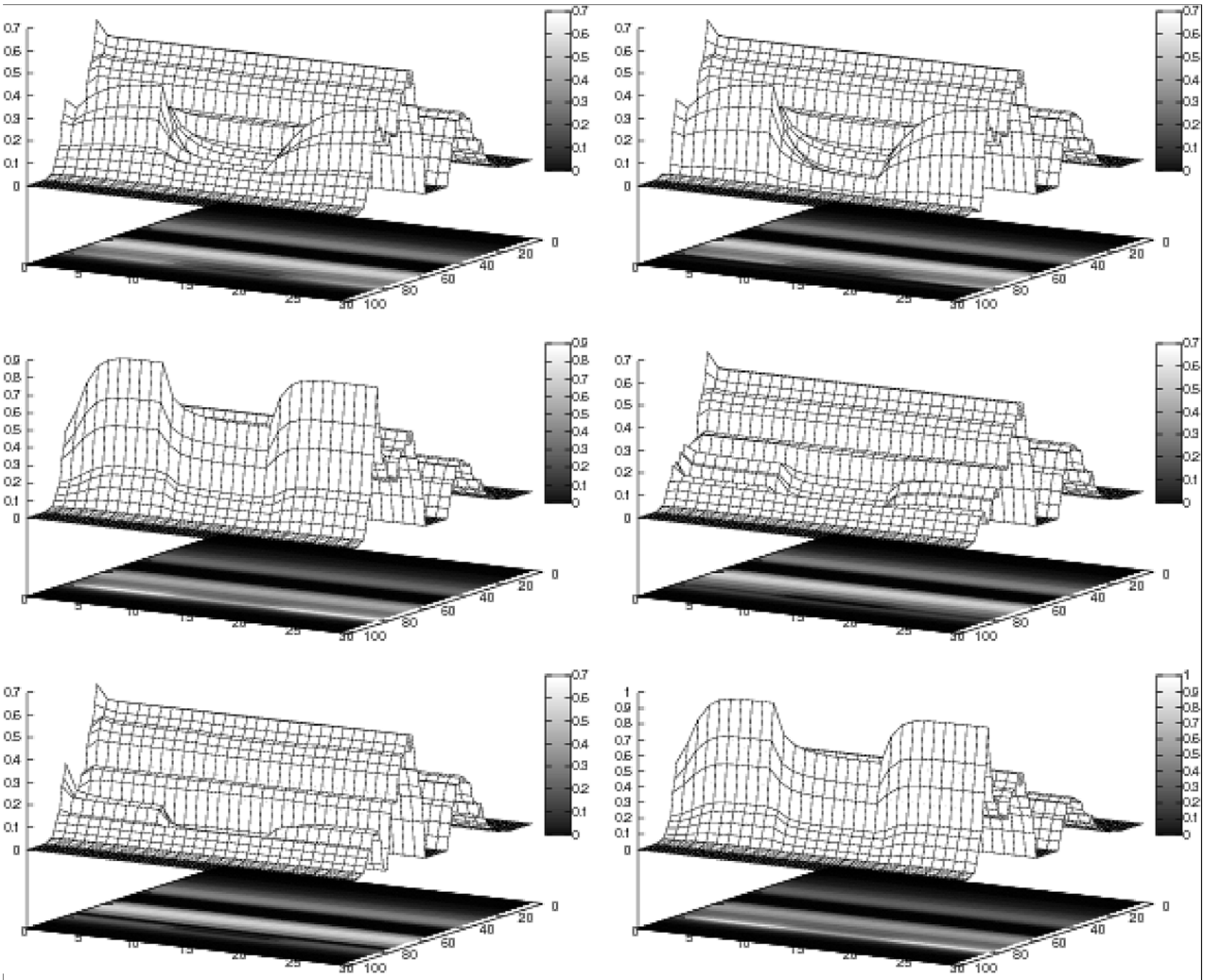

Fig. 7. Phenotypes of the wild-type organism (squared) and of five KO mutants. x axis: Time life of the organisms (time steps). y axis: Functional space. z axis: Efficiency of the organism in performing the function. Mutants are the same as those shown on Fig. 6 (KO of genes 1, 17, 19, 20 and 34).

It consists in producing a mutant lineage in which a targeted gene is invalidated (knocked-out), hence preventing the corresponding protein from being produced by the organism under study. This mutant organism is then compared to the original one (the “wild-type”) with the objective of understanding the role of the knocked-out gene in the organism.

Using KO mutants, both the direct phenotypic contribution of a given protein and its role in the regulation network can be studied. To study the, network, one needs to focus on transcriptome data that gives information about the expression of genes. Systematically knocking out every gene within a genome allows the geneticists to carry out broad comparative studies that could shed light on the complex structure of gene networks [15]. Indeed knock-outs can be used to create perturbations on a gene network in order to help inferring its hidden structure [16]. However, systematic knock out of genes in any genome, even the smallest, yields a vast amount of data that is very difficult to process by hand. It is hence necessary to develop data mining tools that can help analyse knock out data [14].

In silico models have the advantage of providing us with access to any piece of data we would need

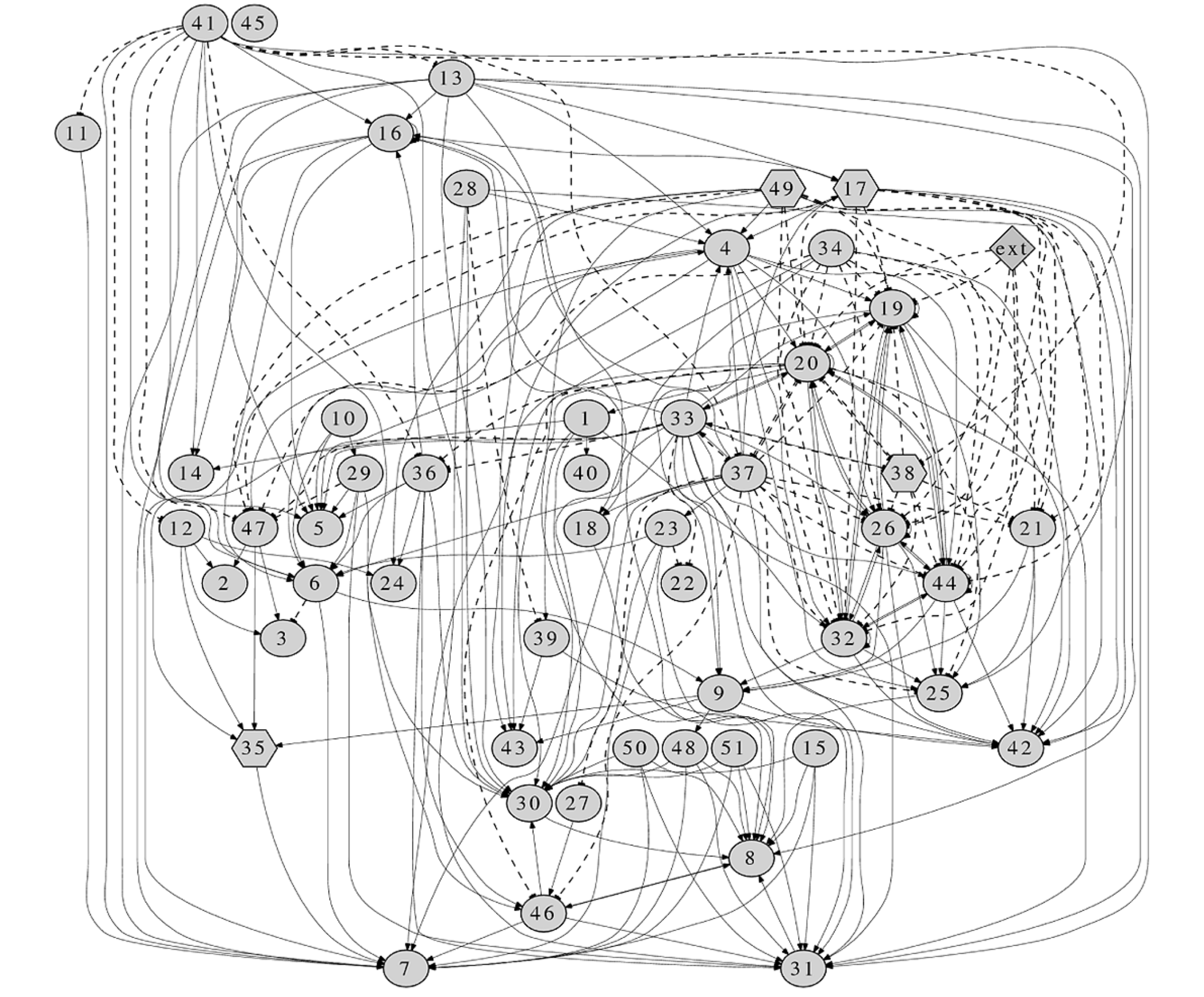

Fig. 8. Structure of the regulatory network after 40 000 generations (experiment S1). The network contains 51 genes, 4 of which being pure transcription factors (hexagons, genes 17, 35, 38, 49), the remaining having all a metabolic activity (ellipses). The signaling protein (dark grey diamond) is connected to the network through a complex connectivity pattern whose behavior is quite impossible to decipher manually. Solid lines represent activation links. Dashed lines represent inhibition links.

in order to understand the system's behaviour, even after the experiment is finished. They are hence particularly suited to be used as a benchmark for data mining methods. We have simulated a systematic knock-out process on an evolved RAevol organism, generating time series representing the concentration of proteins over time for each gene's knock-out. Figures 6 and 7 show some examples of mutant behaviors represented by variations of protein concentrations (Fig. 6) and of the phenotype (Fig. 7). This set of data was then used to understand the underlying gene regulation network thanks to the mining algorithm described below.

3.2. *Mining the KO sequences*

The objective of this method is to provide the expert with some information about genes having similar KO effects. More precisely, the method aims to exhibit groups of genes whose KO lead to similar changes in protein concentration values.[3] This information is obviously useful if we do not know the regulation network, to suggest genes that are involved in the same regulation process. Such groups are also interesting when we know the regulation network (or at least part of it), since in this case these groups can help the expert to focus on genes acting together in the regulation network, and then to gain some understanding of the underlying regulation processes. Here, we are facing the latter case since we perfectly know the network's structure (Fig. 8). Note that the data produced by RAevol can be used in both cases. Computer scientists as well as bioinformaticians are invited to test inference algorithms on the datasets provided on-line (see section "availability" below).

The data describe a set of experiments, where each experiment corresponds to the KO of one single gene, for which we recorded the concentration of all proteins during the life of the cell. We also have at hand an additional experiment used as a reference, that is an experiment where no KO is performed, and that provides the protein concentrations in the original cell (wild-type experiment).

In each experiment, all the concentrations are recorded at the same time and this measure is repeated m times, at a regular time interval. Let the genes be numbered from 1 to n and let us denote exp_i the data obtained from the KO of gene i. Then exp_i can be represented as a $n \times m$ matrix of concentration values $c_{j,t}$, where $c_{j,t}$ denotes the concentration of protein j at time t. Similarly, the wild type experiment denoted exp_{wt} is also represented as a $n \times m$ matrix of concentration values. The global mining process contains three main steps:

- discretization of the concentration values;
- identification of the main concentration values;
- extraction of groups of genes, such that, within a group, the same main concentration values disappear when a KO is performed on the genes of the group.

3.2.1. *Discretizing and finding the main concentration values*

Discretization is performed using a function f_d which maps, by intervals, the concentration values to integer values called the concentration levels. The bounds of the intervals of concentration values used by f_d are set after visual inspection of the distribution of the values. Each element $c_{j,t}$ of each matrix exp_i (and of exp_{wt}) is then replaced by a discrete label of the form $(j, f_d(c_{j,t}))$. Notice that this label encodes the gene number j and the concentration level $f_d(c_{j,t})$.

Let α be a given percentage, then we retain as *main concentration values* only the labels $(j, f_d(c_{j,t}))$ that appear in the wild type experiment exp_{wt} and in at least α percent of the KO experiments $exp_1, \ldots, exp_n$. The set of main concentration values is denoted $\mathcal{M}$.

3.2.2. *Finding groups of genes*

Let p be the number of main concentration values (i.e., the size of $\mathcal{M}$). Then a unique identifier in $\{1, \ldots, p\}$ is associated to each element $x \in \mathcal{M}$, and is denoted $id(x)$.

Let L be a $p \times n$ Boolean matrix defined as follows. An element $l_{k,i}$ (row k column i) of L is equal to 1 if the element $y \in \mathcal{M}$ having identifier $id(y) = k$ is not present in the discretized version of matrix exp_i. Otherwise $l_{k,i}$ is set to 0. Thus an element $l_{k,i}$ is simply equal to 1 if the k^{th} main concentration value has disappeared in the KO of gene i. Notice that in this method we do not take into account the time stamps of occurrence of the main concentration values.

[3] Note that, in RAevol, the protein concentrations are expressed in arbitrary units and are linearly proportional to the mRNA concentrations. Thus the protein concentrations and the expression values are used indifferently in the mining experiments.

Group	measure $\mathcal{A}$	support
4 19 20 26 32 38 44	119	17
17 34 37 38 49	115	23
4 19 20 26 32 44	108	18
17 34 38 49	100	25
17 34 49	84	28

Fig. 9. Top five groups of genes w.r.t. measure $\mathcal{A}$.

Let σ be a given integer, we extract the σ-frequent closed sets [35] from the Boolean matrix L. A σ-frequent set in a Boolean matrix L is a set S of column indexes (i.e., a set of gene numbers) such that the support of S is at least σ, where the support of S is the number of rows in L having 1 values in all column indexes in S. Such a set S is called a closed set if there is no superset of S having the same support as S.

Intuitively, in the method applied here, a σ-frequent closed sets S is a set of genes such that the KO of these genes have in common at least σ missing main concentration values and are in some sense locally maximal (since no superset of S have the same missing main concentration values).

We are interested in sets of genes containing a large number of genes and having a large support (sharing many missing main concentration values). Thus, we used as a ranking criterion a measure $\mathcal{A}$ combining both aspects. Let G be a group of genes, $\mathcal{A}$ is defined by: $\mathcal{A}(G) = |G| \times support(G)$. The sets of genes are sorted by decreasing value of $\mathcal{A}$.

Finally, we remove the sets of size less than 2, since we want groups of genes not singletons.

4. Results

In this section, we present some results of the mining process. The expression data used here corresponds to the regulation network obtained in simulation S1 and depicted Fig. 8. There are 51 genes, on which we performed KO experiments, leading to 51 matrices exp_i. Then we built the exp_{wt} matrix corresponding to the protein concentration in the original cell (i.e., without KO). From visual inspection of the distribution of concentration values we chose the following discretization bounds: 0.05, 0.12, 0.175, 0.225, 0.275, 0.325, 0.375, 0.425, 0.475, 0.525, 0.575, 0.625, 0.675, 0.725, 0.775, 0.825, 0.875, 0.925, 0.975, 1.025, 1.075, 1.125, 1.175, 1.225, 1.275, 1.325, 1.375, 1.425, 1.475, 1.525, 1.575, 1.625, 1.675, 1.725. The first discretization interval contains the concentrations strictly lesser than 0.05, the second intervals contains the values greater or equal to 0.05 and strictly lesser than 0.12, and so on, up to the last interval that contains the values greater or equal to 1.725.

The parameter α was set to 75% to select as main concentration values, only the discrete labels appearing in the wild type experiment and in at least 75% of the KO experiments matrices exp_i. We obtained 105 main concentration values. We built the corresponding Boolean matrix L, on which we ran a frequent closed set extraction tool. The σ threshold was set to 5, in order to collect groups of genes such that the KO of the genes in a group have at least 5 missing main concentration values in common. This resulted in 75 groups of genes that we ranked according to the $\mathcal{A}$ measure. The top five groups are given Fig. 9.

Among these groups, the third one is a subset of the first, and the fourth and fifth are subsets of the second one. Thus, we focus on the first and second groups of genes (the highest $\mathcal{A}$ values): $\{4, 19, 20, 26, 32, 38, 44\}$ ($\mathcal{A} = 119$) and $\{17, 34, 37, 38, 49\}$ ($\mathcal{A} = 115$).

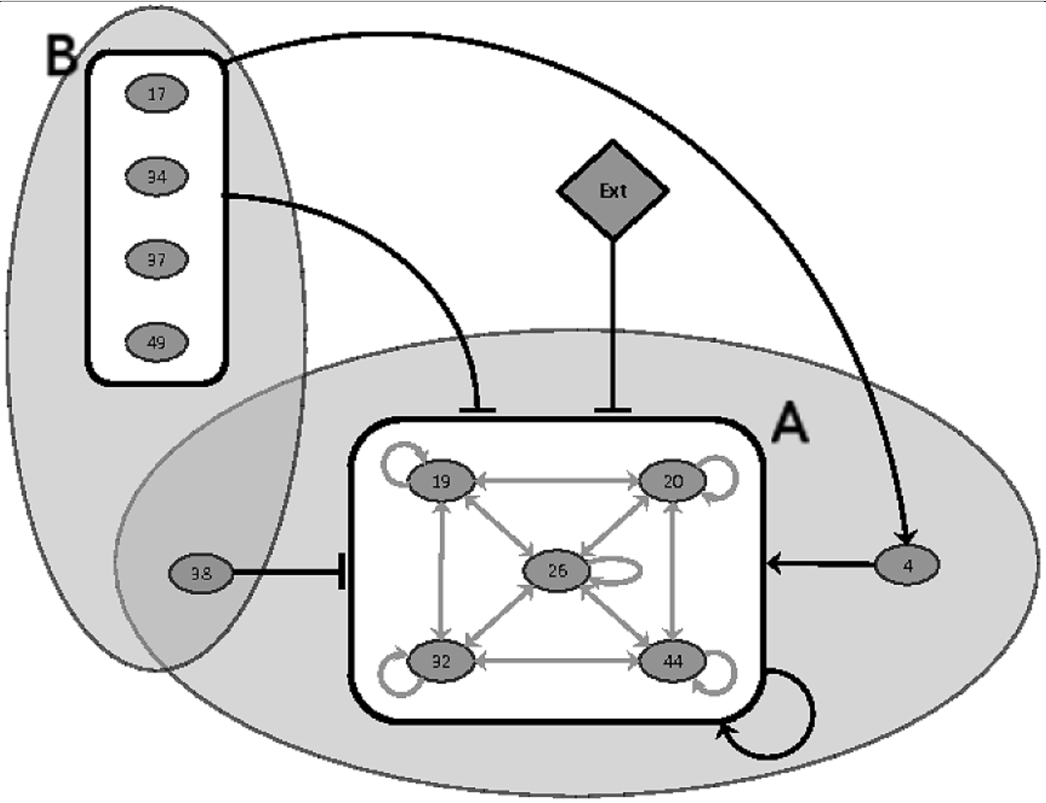

Fig. 10. Sketch view of the network of experiment S1 (see Fig. 8 for the full network). The two groups identified by the mining algorithm are depicted by the grey ellipses. Module A (bottom) is composed of five genes forming a positive-feedback-loop motif. Module B (left) is composed of four genes that are not connected with one another: These genes fulfill the same role in the network although they are not interconnected. Gene 38 can be considered as belonging to the same module even though it does not activate gene 4. The external signal (grey diamond) triggers the activity of module A. Vee-Arrows: Activations; Tee-Arrows: Inhibitions.

Using these two groups as entry-points into the otherwise very difficult network, we were able to fairly easily understand the way this network works and how it manages to respond to the appearance of the external signal. Figure 10 shows a sketch of the network in a way that is meaningful to a human being. It is basically made up of two functional modules that we were able to distinguish from the groups given by our mining process. The first module (module A), whose proteins all code for a metabolic activity in the leftmost lobe of the environment (i.e., the lobe that has to be turned off when the external signal appears), displays a positive-feedback-loop-like motif, which is known to adopt a bistable behaviour [3]. This module is directly inhibited by the external protein as well as by the second module (module B). Module B has a double negative effect on module A: Not only does it down-regulate A but it also phenotypically inhibits the functions performed by A. The positive regulation of module B on module A through gene number 4 seems incoherent but it probably helps the organism to fine-tune the expression of the genes of module A.[4] It is important to note that, while we might have been able to detect module A

[4] This gene and the two modules form an incoherent Feed-Forward Loop – iFFL. Such a loop is known to accelerate state switch [4]. However, since in our case the network has ten time steps to switch, it is not clear whether the iFFL has been selected to decrease the switch delay when the signal is perceived or simply to tune precisely the module activity.

"by hand", we would surely not have discovered module B since its constituting nodes are not connected with one another. Thus we are facing a functional module, whose unity is important for the "cell", but that does not correspond to any structural module in the network. Such a module would obviously have been missed by any link-pruning algorithm [34]. The emergence of such modules during the evolution process, as well as its discovery by our KO-mining algorithm, was a surprise. We now plan to reproduce similar experiments to test whether or not this type of structure emerges repeatedly.

5. Conclusion

Understanding complex structures such as bacterial regulatory networks is a challenging objective. Obviously, such an objective cannot be efficiently tackled with a single, isolated, tool or within a single, isolated, discipline. In order to complement direct analysis of biological data (genomics and comparative genomics, transcriptomics, systems biology, etc.), we propose an *in silico* model of the evolution of regulation networks. This model can be used to understand how the historical process of evolution can account for the complex structures observed in real organisms, both at the genomic and transcriptomic levels. This model, called RAevol, is an integrated model which realistically accounts for all the organization levels of a real bacterial organism. In particular, the regulatory network is only indirectly submitted to the evolutionary operators (mutation and selection): Mutations act on the genomic level (which indirectly encodes the regulation network) and selection is based on the global metabolic activity of the organisms (which is indirectly encoded by the regulation network). In other words, the structure of the regulation network is never directly manipulated nor evaluated. Thus, although indirectly constrained, the networks evolve freely and progressively acquire complex structures that we can analyze later on in the light of the evolutionary conditions under which they appear.

The experiments we present here show that, although the organisms were facing quite simple tasks, they developed complex regulation networks whose connectivity is impossible to decipher manually (Fig. 8). Surprisingly, we had to face the same problems practitioners face when they analyze real biological data. This is the reason why we decided to use the artificially generated data as a benchmark for data-mining and inference algorithms. Following this idea, we generated knock-out mutants of one of the evolved organisms and used a motif-detection algorithm to identify groups of genes on the basis of their KO-similarities. Interestingly, we discovered two main groups that, when analyzed further, gave us a sketchy grasp of the network structure and behavior (Fig. 10). The structure we discovered during the mining process shows the great potential of using computational evolution as benchmark generators. Indeed, the computational evolutionary process is able to surprise us by creating such unexpected structures while they can hardly be created by traditional benchmarking tools based on random graph generation.

We now plan to use RAevol as an *in silico* experimental evolution platform in order to go further in the understanding of the evolutionary dynamics of regulation networks. Simultaneously, we will produce various datasets that will be available for the scientific community for, e.g., testing data-mining algorithms, phylogeny reconstruction or network inference. Since we are able to generate networks of variable complexity [8], we can produce collections of benchmarks of different sizes and complexity. We are confident that both directions are to help practitioners to better take-up the challenge of systems biology.

6. Availability

RAevol software is currently in alpha version, supported by the LIRIS-Turing group at INSA-Lyon. For any request, please contact directly the authors. All the data of the three simulations (genomes, networks, protein variations for the wild type and for all KO mutants) are available on the website of the authors: http://liris.cnrs.fr/guillaume.beslon/IDAj.data/.

Acknowledgment

The authors want to thank Carole Knibbe and all the members of the Turing team and the BSMC group for fruitful discussions. The BSMC group provides us with the computing resources and we would like to warmly thank Fabien Chaudier for his invaluable help in this domain. This work has been funded by the Rhône-Alpes Institute of Complex Systems (IXXI), the french ANR MDCO Bingo2 2008-2010 project and the Spanish Ministry of Education (project number TIN2007-67148).

References

[1] C. Adami, Digital genetics: unravelling the genetic basis of evolution, *Nat Rev Genet* **7**(2) (2006), 109–118.

[2] U. Alon, Biological networks: the tinkerer as an engineer, *Science* **301**(5641) (2003), 1866–1867.

[3] U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits*, Chapman & Hall/CRC, 2006.

[4] U. Alon, Network motifs: theory and experimental approaches, *Nat Rev Genet* **8**(6) (2007), 450–461.

[5] W. Banzhaf, Artificial regulatory networks and genetic programming, in: *Riolo and Bill Worzel*, L. Rick, ed., Genetic Programming Theory and Practice, chapter 4, Kluwer, 2003, pp. 43–62.

[6] A.-L. Barabási and Z.N. Oltvai, Network biology: understanding the cell's functional organization, *Nat Rev Genet* **5**(2) (2004), 101–113.

[7] G. Beslon, D. Parsons, Y. Sanchez-Dehesa, J.-M. Pena and C. Knibbe, Scaling Laws in Bacterial Genomes: A Side-Effect of Selection of Mutational Robustness, *To appear in BioSystems*, 2010.

[8] G. Beslon, Y. Sanchez-Dehesa, D.P. Parsons, J.-M. Pe na and C. Knibbe. Scaling laws in digital organisms, In *Proceedings of Information Processing in Cells and Tissues* (*IPCAT'2009*), Ascona, Switzeland, 5–9 April 2009, pp. 111–114.

[9] A. Crombach and P. Hogeweg, Evolution of evolvability in gene regulatory networks, *PLoS Comp Biol* **4**(7) (2008), e1000112.

[10] P. Dittrich, J. Ziegler and W. Banzhaf, Artificial chemistries-a review, *Artif Life* **7**(3) (2001), 225–275.

[11] S.F. Elena and R.E Lenski, Evolution experiments with microorganisms: the dynamics and genetic bases of adaptation, *Nat Rev Genet* **4**(6) (2003), 457–469.

[12] P. François and V. Hakim, Design of genetic networks with specified functions by evolution in silico, *Proc Natl Acad Sci USA* **101**(2) (Jan 2004), 580–585.

[13] L.A. Galli-Taliadoros, J.D. Sedgwick, S.A. Wood and H. Korner, Gene knockout technology: a methodological overview for the interested novice, *Journal of Immunological Methods* **181**(1) (1995), 1–15.

[14] F. Geier, J. Timmer and C. Fleck, Reconstructing gene-regulatory networks from time series, knock-out data, and prior knowledge, *BMC Systems Biology* **1** (2007), 11.

[15] Z. Gu, L.M. Steinmetz, X. Gu, C. Scharfe, R.W. Davis and W.-H. Li, Role of duplicate genes in genetic robustness against null mutations, *Nature* **421**(6918) (2003), 63–66.

[16] M. Hecker, S. Lambeck, S. Toepfer, E. van Someren and R. Guthke, Gene regulatory network inference: data integration in dynamic models-a review, *BioSystems* **96**(1) (2009), 86–103.

[17] T. Ideker, T. Galitski and L. Hood, A new approach to decoding life: systems biology, *Annual Review of Genomics and Human Genetics* **2** (2001), 343–372.

[18] T. Ideker, L. Winslow and D. Lauffenburger, Bioengineering and systems biology, *Annals of Biomedical Engineering* **34**(2) (2006), 257–264.

[19] S.C. Janga and J. Collado-Vides, Structure and evolution of gene regulatory networks in microbial genomes, *Research in Microbiology* **158**(10) (2007), 787–794.

[20] N. Kashtan and U. Alon, Spontaneous evolution of modularity and network motifs, *Proc Natl Acad Sci USA* **102**(39) (2005), 13773–13778.

[21] H. Kitano, Computational systems biology, *Nature* **420**(6912) (2002), 206–210.

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

[23] J.F. Knabe, C.L. Nehaniv and M.J. Schilstra, Genetic regulatory network models of biological clocks: Evolutionary history matters, *Artificial Life* **14**(1) (2008), 135–148.

[24] C. Knibbe, Structuration des génomes par sélection indirecte de la variabilité mutationnelle, une approche de modélisation et de simulation, *PhD thesis, INSA-Lyon*, 2006, p. 177.

[25] C. Knibbe, A. Coulon, O. Mazet, J.-M. Fayard and G. Beslon, A long-term evolutionary pressure on the amount of noncoding dna, *Mol Biol Evol* **24**(10) (2007), 2344–2353.

[26] C. Knibbe, J.-M. Fayard and G. Beslon, The topology of the protein network influences the dynamics of gene order: from systems biology to a systemic understanding of evolution, *Artif Life* **14**(1) (2008), 149–156.

[27] C. Knibbe, O. Mazet, F. Chaudier, J.-M. Fayard and G. Beslon, Evolutionary coupling between the deleteriousness of gene mutations and the amount of noncoding sequences, *J Theor Biol* **244**(4) (2007), 621–630.

[28] P.D. Kuo, W. Banzhaf and A. Leier, Network topology and the evolution of dynamics in an artificial genetic regulatory network model created by whole genome duplication and divergence, *BioSystems* **85** (2006), 177–200.

[29] A. Leier, P.D. Kuo, W. Banzhaf and K. Burrage, Evolving noisy oscillatory dynamics in genetic regulatory networks, in: *Proceedings of the 9th European Conference on Genetic Programming, volume 3905 of Lecture Notes in Computer Science*, P. Collet, M. Tomassini, M. Ebner, S. Gustafson and Anikó Ekárt, eds, Budapest, Hungary, Springer, 10–12 April 2006, pp. 290–299.

[30] R.E. Lenski, J.E. Barrick and C. Ofria, Balancing robustness and evolvability, *Plos Biol* **4**(12) (2006), e428.

[31] D. Marbach, T. Schaffter, C. Mattiussi and D. Floreano, Generating Realistic In Silico Gene Networks for Performance Assessment of Reverse Engineering Methods, *Journal of Computational Biology* **16**(2) (2009), 229–239.

[32] C. Mattiussi and D. Floreano, Analog Genetic Encoding for the Evolution of Circuits and Networks, *IEEE Transactions on Evolutionary Computation* **11**(5) (2007), 596–607.

[33] N. Molina and E. van Nimwegen, The evolution of domain-content in bacterial genomes, *Biol Direct* **3** (2008), 51.

[34] M.E.J. Newman and M. Girvan, Finding and evaluating community structure in networks, *Phys Rev E* **69**(2) (2004), 026113.

[35] N. Pasquier, Y. Bastide, R. Taouil and L. Lakhal, Efficient mining of association rules using closed itemset lattices, *Information Systems* **24** (1999), 25–46.

[36] D. Ropers, H. de Jong, M. Page, D. Schneider and J. Geiselmann, Qualitative simulation of the carbon starvation response in escherichia coli, *BioSystems* **84**(2) (2006), 124–152.

[37] Y. Sanchez-Dehesa, D.P. Parsons, J.-M. Pe na and G. Beslon, Modelling evolution of regulatory networks in artificial bacteria, *Math Model Nat Phenom* **3**(2) (2008), 27–66.

[38] K. Struhl, Fundamentally different logic of gene regulation in eukaryotes and prokaryotes, *Cell* **9**(1) (1999), 1–4.

[39] E. van Nimwegen, Scaling laws in the functional content of genomes, *Trends Genet* **19**(9) (2003), 479–484.

[40] C.O. Wilke, J.L. Wang, C. Ofria, R.E. Lenski and C. Adami, Evolution of digital organisms at high mutation rates leads to survival of the fittest, *Nature* **412**(6844) (2001), 331–333.

[41] D. Wolf, Motifs, modules and games in bacteria, *Current Opinion in Microbiology* **6**(2) (2003), 125–134.

[42] E. Yeger-Lotem, S. Sattath, N. Kashtan, S. Itzkovitz, R. Milo, R.Y. Pinter, U. Alon and H. Margalit, Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction, *Proc Natl Acad Sci USA* **101**(16) (2004), 5934–5939.

[43] X. Zhu, M. Gerstein and M. Snyder, Getting connected: analysis and principles of biological networks, *Genes & Development* **21**(9) (2007), 1010–1024.