Inferring Gene Regulatory Network Models from Time-Series Data Using Metaheuristics

José Eduardo H. da Silva*, Heder S. Bernardino*, Helio J.C. Barbosa*†,
Alex B. Vieira*, Luciana C.D. Campos*, and Itamar L. de Oliveira*
jehenriques@ice.ufjf.br, heder@ice.ufjf.br, hcbm@lncc.br,
alex.borges@ice.ufjf.br, luciana.campos@ice.ufjf.br, itamar.leite@ice.ufjf.br
*_Universidade Federal de Juiz de Fora_, Juiz de Fora, Brazil
† _Laboratório Nacional de Computação Científica_, Petrópolis, Brazil

***Abstract*—The inference of Gene Regulatory Networks (GRNs) from gene expression data is a hard and widely addressed scientific challenge with potential industrial and health-care use. Discrete and continuous models of GRNs are often used (i) to understand the process, and (ii) to predict the values of the relevant variables. Here, we propose a procedure to infer models of GRNs from data where (i) the data is binarized, (ii) a Boolean model is created using a Cartesian Genetic Programming technique, (iii) the obtained Boolean model is converted to a system of ordinary differential equations, and (iv) an Evolution Strategy defines the parameters of the continuous model. As a result, we expect to reduce the effect of noise and to improve biological interpretability. The proposed method is applied to two ODE systems that describe the circadian rhythm network dynamic, with 5 and 10 state variables. The models created by the proposed procedure are able to reproduce the behavior observed in the original data.**

I. Introduction

Systems Biology is an interdisciplinary research area that involves Biology, Chemistry, Physics, etc. This area focuses on the interaction among the components of a biological system [1]. For instance, to understand the behavior of organisms at the molecular level, one needs to know what/where/when genes are expressed. Gene expression is a complex process regulated at different levels in protein synthesis [2]. All cellular activities are controlled by their genes through a complex network that forms proteins from DNA and the gene expression depends on the relationship of the genes in this network, known as Gene Regulatory Network (GRN).

In GRNs representations, genes are the network nodes and the regulatory relationships between genes are the network edges [2]. E.g., Figure 1 shows a GRN with three genes where a direct arrow indicates the activation of the regulatory relationship between two given genes; otherwise, it indicates its inhibition.

Most GRNs present several components connected via feedback loops. Thus, computational methods for modeling and simulating GRNs are indispensable [3]. Also, the understanding of complex patterns of gene interactions represents a scientific challenge with high biotechnological and health applicability [2].

GRNs can be modeled by a continuous or discrete model. Typically, the former approach is represented by an ordinary differential equations (ODE) system and the latter uses a boolean network on your representation. Also, these models can be deterministic or stochastic. The choice of the network model is commonly based on the type of data available [2]. Although continuous models have been widely applied [2], their use is limited for modeling biological systems where the kinetic parameters are unknown. On the other hand, qualitative insights have been obtained when Boolean models are analyzed [4].

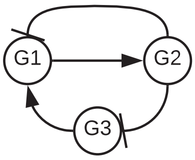


Fig. 1. A GRN Illustration composed of three genes: G1, G2, and G3. The gene G2 inhibits both G1 and G3. The genes G1 and G2 are activated by genes G3 and G1, respectively.

In this work, we propose a procedure for the GRNs inference from continuous data using evolutionary computation techniques; where: (i) the original time-series data is binarized, (ii) a Boolean model is obtained using Cartesian Genetic Programming (CGP), (iii) a continuous model in the form of an ODE system is created through the Boolean model, and (iv) the numerical coefficients of the ODE system are optimized by Evolution Strategy (ES). Thus, the proposed method produces three GRN models: a qualitative Boolean model, a continuous model in the form of an ODE system with undefined numerical coefficients, and a final model in the form of an ODE system. The Boolean models represent the interactions between the proteins and, as a consequence, allow for insights in systems biology. Also, ODE systems are accurate symbolic GRN models, providing knowledge concerning the phenomenon of interest.

More precisely, the data discretization highlights the active/inactive genes. It is possible to binarize the data using the z-score, such as in [5]. Moreover, the CGP [6], a Genetic Programming (GP) technique widely applied to the design of digital circuits, is suited to the task of inferring the Boolean model. The Boolean model is then transformed into an ODE system through the techniques presented in [7], which are based on multivariate polynomial interpolation. ES [8] is

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

typically applied to numerical optimization and is known as a good optimizer for problems in continuous search spaces [9]. ES concludes the modeling process determining the numerical coefficients of the ODE system. The use of time-series data to determine the numerical coefficients of the ODE system makes possible to reproduce the biochemical processes, and this cannot be done with Boolean models [7].

Preliminary experiments using two Circadian Rhythms, with 5 and 10 variables, evaluate the proposal. The results are promising as the proposed method is able to generate interpretable models with the behavior of the observed phenomena. Also, these problems have complex behaviors, such as oscillations, and are considered as important to be investigated in [7].

The remainder of this paper is organized as follows. Related work is presented in Section II. In Section III, the methods used here are described and the proposed procedure is defined. The computational experiments for preliminary evaluate the proposal are presented in Section IV. Finally, concluding remarks and suggestions of future works are shown in Section V.

II. Related Work

Some procedures for modeling GRNs can be found in the literature and we present here some methods related to the proposal. Boolean model is the simplest form to represent GRNs, where the variables represent the expression level of a gene. In this model, each variable of interest is defined as a discrete function that determines the next state of the biological system (protein concentration level in the next time step, for example) [2]. The inferred models provide a qualitative measure of gene regulatory mechanisms [4] and are able to represent the dynamics of several biological phenomena.

For instance, Boolean models are discrete and the choice of a discretization procedure depends on many factors, such as supervision, data sample, technology, level of discretization, and scope [5]. Commonly, the discretization of GRN data uses an alphabet of two symbols [10], [11] to represent activation and inhibition. An alphabet with three symbols $\{-1, 1, 0\}$ is also common and indicates, respectively, the activation, inhibition, and unchanging regulation. There is a trade-off between the level of discretization and the loss of information (and computational complexity) [12].

Ordinary Differential Equations [3] can be used to model the concentrations of RNAs, proteins, and other molecules, using time-dependent variables. Regulatory interactions take the form of functional and differential relationships between the variables [4]. More specifically, a gene regulation is modeled by an ODE system that expresses the production rate of a component as a function of the concentrations of the other ones. These functions are commonly named kinetic equations.

ODE systems for modeling gene regulatory networks are inferred by a GP technique in [13]. Least mean square (LMS) error was used, and accurate models were created when the proposal was applied to three networks.

Sirbu et al. [14] present a comparison of evolutionary techniques when inferring gene regulatory network models. The authors analyze several existing algorithms for models in the ODE system form. They have presented seven evolutionary algorithms and applied to both synthetic and real gene expression data from DNA microarrays. The authors conclude that pure evolutionary algorithms are useful to analyze only small-scale systems. Then, they propose hybrid methods to increase scalability and, their best results refer to the hybridization for larger networks (up to 30 genes). The authors also have shown that the convergence has a speed up by splitting the objectives and applying a multiobjective approach.

III. Methods and Proposed Approach

The current proposal involves (i) the data discretization, (ii) the inference of a Boolean model via CGP, (iii) the generation of an ODE system from the Boolean model, and (iv) the numerical coefficients optimization of the continuous model. In the following sections, we describe the methods we use in each step of the proposed approach.

A. Data Discretization

The gene expression data considered in this work is a time-series. A common discretization technique standardizes the values using z-scores with a normal distribution $\mathcal{N}(0, 1)$ [5]. The values a'_{ij} generated by this transformation are then used to calculate

$$a_{ij} = \begin{cases} 1, & a'_{ij} - a'_{i(j-1)} \geq 0 \\ 0, & a'_{ij} - a'_{i(j-1)} < 0, \end{cases} \tag{1}$$

where i is the index of the variable and j is the instant of time. According to this discretization, the value is 1 (active) when the observed values increase, and 0 (inactive) otherwise. Other procedures can be found in the literature for discretization [15], [16]. However, these works present more than two-level of discretization (future work) or require user-defined parameters. The binarized data makes it possible to create a state transition diagram and, as a result, one can create a truth table. There might occur ambiguities between transitions and, in this case, (i) the most frequent transition is chosen and, (ii) when a tie occurs, the different outputs in the possible transitions are considered “don’t care". CGP is then applied to the truth table to evolve the Boolean model.

B. Cartesian Genetic Programming (CGP)

CGP is a technique to generate programs (computer programs and other complex structures) encoded by Directed Acyclic Graphs (DAGs). In particular, CGP has been widely applied to the design of digital circuits [17]. One of the advantages of a graph-based representation is its capacity of reusing sub-components of the programs [17].

The CGP encodes DAGs by a two-dimensional matrix of nodes, with n_c columns and n_r rows. The genes are integer values and, for each node, there are inputs (called connection genes) and the operation/function that node performs (called function genes). Given a node in the matrix, the nodes at its left side can be used as inputs. To constrain the connectivity of the graph, a user-defined parameter called levels-back (lb) limits the number of columns at the left side where inputs

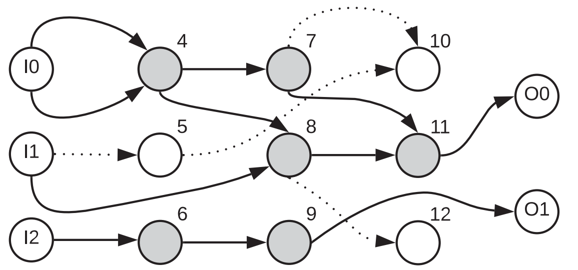


Fig. 2. Representation of a CGP's individual. The nodes labeled with integer values from 4 to 12 represent components of the genotype, where the grey ones are active nodes and the white are inactive. The dashed lines are the connections that do not interfere in the phenotype. I0, I1, and I2 are the primary inputs and O0 and O1 are the outputs.

can be selected from. A special case occurs when the $n_c = 1$ and lb= n_c, in which the genotype can represent any DAG formed by n_c nodes [17]. The function set is user-defined and problem-dependent. For example, logic functions or gates are used when designing digital circuits.

Figure 2 shows a CGP's individual with $n_c = 3$, $n_r = 4$, and lb= n_c. The gray nodes are considered active as they effectively contribute to the individual's phenotype and white nodes are inactive. The dashed lines show the connections of these inactive nodes, and the solid lines show the connections between the active nodes. The illustrative example shows the genotype with (i) the program inputs (I0, I1, and I2), (ii) the internal nodes (indexes 4 to 12) with their inputs and functions, and (iii) the outputs of the program (O0 returns the output of node 11, and O1 returns the output of node 9).

CGP uses a simple Evolution Strategy variant, $(1 + \lambda)$-ES, as its search procedure. In this optimization method, λ new candidate solutions are generated based on one parent. Originally, these new individuals were generated by a simple point mutation, but currently, other approaches are widely used, such as the single active mutation. In CGP, small populations are more efficient in obtaining feasible circuits [17] and, normally, $\lambda = 4$. Also, CGP uses elitism, keeping the best individual from the previous generation.

For the inference of Boolean models, we adapt the truth table obtained we obtain from the discretization procedure as input to CGP. The goal is to find a Combinational Logic Circuit (CLC) whose outputs depend only on the combinations of the current states, meeting the transitions provided by the truth table. We may have not all possible inputs to address the current problem since: (i) the data is a (short) observation of the phenomenon modeled, and (ii) some transitions may never occur. Thus, any transition that is not present in the original truth table is considered irrelevant (called don't care).

The circuit is evolved using two steps: (i) increasing the number of matches concerning the truth table, and (ii) reducing the number of logic gates (nodes). The first step generates a solution which models the discretized data, and the second step decreases the complexity of the model. The latter is important as a smaller number of logic gates makes it easier to obtain the continuous model.

C. Determining the Continuous Model from the Discrete One

The methodology we consider to obtain continuous models from Boolean models is based on Wittmann et al. [7]. In general, a Boolean model consists of N species, $X_1, X_2, \ldots, X_N$, and each species takes values $x_i \in \{0, 1\}$. In addition, there is a discrete update function B_i for each species $X_i(t)$, at time t, that gives its value at the instant $t + 1$, as

$$X_i(t+1) = B_i(x_{i1}(t), x_{i2}(t), ..., x_{iN_j}(t)). \tag{2}$$

The first step for obtaining a continuous model equivalent to the Boolean model is to transform each discrete variable x_i into a continuous variable $\overline{x_i} \in [0, 1]$, where concentrations are normalized in the unitary interval [7]. To do so, B_i is transformed into a continuous update function $\overline{B_i}$. The continuous update function ($\overline{B_i}$) is defined here using HillCubes. Initially, BooleCubes are obtained through a multivariate polynomial interpolation, according to

$$\overline{B_i}(\overline{x_1}, ..., \overline{x_N}) = \sum_{x_1=0}^{1} ... \sum_{x_N=0}^{1} B(x_1, ..., x_N)P(x_i, \overline{x_i}) \tag{3}$$

where B$(x_1, x_2, ..., x_n)$ represents the update function of the species X_i, and $P(x_i) = \prod_{i=1}^{N}(x_i\overline{x_i} + (1 - x_i)(1 - \overline{x_i}))$.

However, BooleCubes are not able to represent the sigmoidal shape commonly present in biological interactions. HillCubes are used here, as they can represent this "switching" behavior using Hill's sigmoid functions [7]. A Hillcube can be defined as

$$f(\overline{x}) = \frac{\overline{x}^n}{\overline{x}^n + k^n}, \tag{4}$$

where n determines the slope of the curve (interaction cooperativity) and k is a threshold.

The continuous variables $\overline{x_i}$ are replaced by the Hill's functions in $\overline{B_i}$ and the new continuous update function, called HillCube, can be defined as

$$\overline{B}_i^H(\overline{x}_{i1}, ..., \overline{x}_{iN_i}) = \overline{B}_i\left(f_{i1}(\overline{x}_{i1}), ..., f_{iX_i}(\overline{x}_{iN_i})\right). \tag{5}$$

As a HillCube never equals 1, normalization is applied:

$$\overline{B}_i^{Hn}(\overline{x}_{i1}, ..., \overline{x}_{iNi}) = \overline{B}_i^I\left(\frac{f_{i1}(\overline{x}_{i1})}{f_{i1}(1)}, ..., \frac{f_{iN_i}(\overline{x}_{iN_i})}{f_{iN_i}(1)}\right) \tag{6}$$

Finally, the normalized HillCubes are applied to

$$\dot{\overline{x_i}} = \frac{1}{\tau_i}(\overline{B}(\overline{x_{i1}}, \overline{x_{i2}}, ..., \overline{x_{iN_i}}) - \overline{x_i}) \tag{7}$$

to obtain the temporal variation of $\overline{x}_i$. This value is composed of (i) the continuous update function $\overline{B}$ (here we adopt the HillCubes), which describes the production of the species X_i and a first-order decay term, and (ii) its corresponding parameter τ_i, that can be understood as the lifetime of the species X_i.

In brief, an ODE system is determined using HillCubes by the following steps: (i) to obtain a BooleCube through the multivariate polynomial interpolation presented in Equation 3, and (ii) to transform this BooleCube into a HillCube. The model generated contains numerical coefficients (n, k, and τ) which should be determined. For this task, an Evolution Strategy approach is proposed here.

D. Evolution Strategy (ES)

Evolution Strategy (ES) [9] is an evolutionary computation method widely used for solving optimization problems involving continuous search spaces [18]. Normally, this technique self-adapts its mutation parameters, making it possible to change its behavior during the searching process. The individuals are composed of (i) decision variables, and (ii) strategy parameters. Here we used a simple ES approach in which these strategy parameters are the lengths of the mutation steps.

Algorithm 1 presents a pseudo-code of ES where $P^{(t)}$ is the set of individuals in the generation t, $p \in P^{(t)}$ is a tuple (x, Ψ), Ψ_V is the set of variation parameters, and Ψ_{Age} represents the ages of the individuals. The user-defined parameters are the number of parents μ, the amount of offspring λ, the number ρ of parents that generate offspring, and the maximum age κ a candidate solution can reach. The value of κ also defines the replacement operator. When $\kappa = 1$, the parents of the next generation will be the offspring created in the current generation. This is called the (μ, λ)-ES. One has the $(\mu + \lambda)$-ES when κ is an infinite lifetime, and the selection occurs among the best individuals considering the μ parents and the λ offspring. The mutation is performed adding a random value taken from a Multivariate Normal Distribution [9], i.e., a new individual is created by $x' = x + \Psi_V \times \mathcal{N}(0, \mathbf{I})$.

Algorithm 1 Pseudo-code of an ES. Adapted from [9].

```
Initialization of $P^{(0)}$ with $\mu$ individuals
$\forall$ p $\in$ P: p.$\Psi_{Age}$ $\leftarrow$ 1, p.f $\leftarrow$ f(p.x)
while stop criteria is not reached do
    $Q^t \leftarrow \oslash$
    for i = 1 $\rightarrow \lambda$ do
        Randomly select p parents $p_1, \dots, p_p \in P^{(t)}$
        q $\leftarrow$ Variation($p_1, \dots, p_p, p_1.\Psi_V, \dots, p_p.\Psi_V$)
        q.$\Psi_{Age}$ $\leftarrow$ 0, q.f $\leftarrow$ f(q.x)
        $Q^{(t)} \leftarrow Q^{(t)} \cup$ {q}
    end for
    $P^{(t+1)} \leftarrow$ best($\mu, Q^{(t)} \cup \{p \in P^{(t)} : p.\Psi_{Age} < k\}$)
    $\forall p \in P^{(t+1)}$: $p.\Psi_{Age} \leftarrow p.\Psi_{Age} + 1$ and update $p.\Psi_V$
    t $\leftarrow$ t + 1
end while
```

The parameters n, k and τ in the HillCubes are the numerical coefficients to be optimized. In the proposal, each candidate solution in ES contains the parameters n, k and τ of all HillCubes present in the continuous model. One can notice that the complexity of the optimization problem is directly proportional to the size of the ODE system.

To conclude the used ES and, consequently, the proposed procedure, we present the adopted objective function. We desire to determine $\mathbf{g}(\mathbf{x}, t)$ such that $\mathbf{x}'(t) = \mathbf{g}(\mathbf{x}, \mathbf{t})$ fits the observed data $(\mathbf{x}_i, t_i)$, with $i = 1, \dots, m$. A dynamic model can be evaluated by numerically integrating the ODE system $\mathbf{x}'(t) = \mathbf{g}(\mathbf{x}, t)$, corresponding to candidate $\mathbf{g}(\mathbf{x}, t)$, and comparing these values to the observed data. Although other approaches can be found in the literature, such as those using numerical derivatives [19], [20], the use of numerical integration tends to generate more accurate models. The optimization problem solved by ES is the minimization of the absolute difference between the values calculated by the numerical integration and those in the original data (1-norm). Preliminary experiments were carried out with the minimization of the quadratic difference (2-norm) but the results were worse than those obtained with the use of the 1-norm. No comparison is provided here due to the lack of space.

It is important to highlight that BooleCubes and HillCubes use normalized data $(\overline{x_i})$ [7]. Here the normalization is performed into the ODE system during the numerical integration. An example of this normalization applied to $\overline{G2}$ is

$$\dot{\overline{G2}} = \frac{1}{\tau_{G2}} \left(\frac{N(\overline{G1})^{n_{G2G1}}}{N(\overline{G1})^{n_{G2G1}} + k_{G2G1}^{n_{G2G1}}} - N(\overline{G2}) \right), \qquad (8)$$

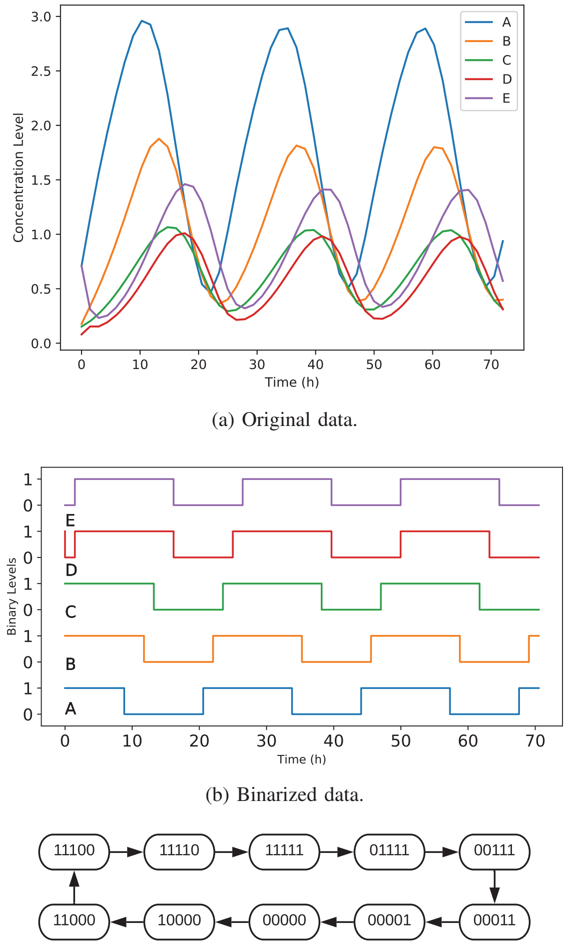
where $N(v) = \frac{v}{\max(v)}$, $G1$ and $G2$ are variables of interest, and $\max(G1)$ and $\max(G2)$ are, respectively, their maximum values in the data.

IV. Computational Experiments

The proposed method was applied to two datasets generated by two gene regulatory networks whose dynamic is described by ODE systems. Both ODE model the Drosophila's circadian rhythm network. Circadian rhythm, or circadian cycle, designates the period of approximately 24 hours on which the biological cycle of almost all living beings is based, influenced mainly by the variation of light, temperature, tides, and winds during the day. Hence, each data set corresponds to the network species concentration that is repeated in a 24-hour cycle. One of the ODE systems has five state variables (thus, five ODEs) and the other one has ten state variables. The five variables model is a GRN that has been proposed for circadian oscillations in the Drosophila PER protein and its respective gene and mRNA [21]. The ten variable model takes into account more details about the cyclic GRN describing the dynamic of circadian rhythm [22]. For more details we refer the reader to [21] and [22]. Through the MATLAB® ODE solver, we solve both ODE systems and generated a time series containing 50 points evenly spaced in a simulation time interval from 0 to 72 hours. So, no experimental data (RNASeq, for example) was used. They are artificial data obtained from real GRN models.

The parameter setting used in CGP is: 20 independent runs, maximum number of objective function evaluations equal to 50.000, n_r = 1, n_c = 100, lb= n_c, and function set Γ = {AND, OR, NOT, XOR}. As previously indicated, ES is proposed to find the best values for the numerical coefficients τ, n, and k in every equation of the ODE system. We applied ES to the circuit obtained by CGP with the smallest number of logic gates. The used parameters are: $\mu = 15$, $\lambda = 105$ (as typically $\lambda = 7 \times \mu$), and maximum number of objective function evaluations equal to 10.000. The source code of the proposed approach and supplementary material are available[1].

[1] https://github.com/ciml/



(a) Original data.

(b) Binarized data.

(c) State transition diagram.

Fig. 3. Plots and state transition diagram – 5 Variables Circadian Rhythm.

The methods were implemented using Python 3.6 and the *odeint* function of *SciPy*[2] numerically solves the initial value problem of the first order ODE system. The error of the model was defined as the absolute difference (1-norm) between the actual and predicted values. Also, $n \in \mathbb{Z}$ and $k, \tau \in \mathbb{R}$, and these variables are bounded as $1 \leq n \leq 25$, $0.1 \leq k \leq 1$ and $0.1 \leq \tau \leq 5$ [23]. In the ES, the variables are evolved in $\mathbb{R}$ and the integer part is used for the n values.

A. Circadian Rhythm with 5 Variables

The first model created here is for a circadian rhythm with 5 variables, namely, A, B, C, D, and E. The data (time series) of the 5 variables is presented in Figure 3a.

Initially, the data is binarized as described in Section III-A. The new data can be shown in Figure 3b, where 0 represents inactivation and 1 indicates activation. The values 0 and 1 were translated with respect to the ordinate axis to improve readability. Ten different states and 11 state transitions were determined using these data and the state transition diagram can be seen in Figure 3c. One expects a truth table composed of $2^5 = 32$ rows as the modeled phenomenon involves 5 variables. As indicated in Section III-B, the states not observed are considered “don’t care".

TABLE I
Boolean Expressions for the 5 variable Circadian Rhythm

Variable	Expression
A	not(E)
B	A
C	B
D	C
E	D

TABLE II
Boolean Expressions for the 10-Variable Circadian Rhythm.

Variable	Expression	Variable	Expression
A	not(J)	F	A
B	E	G	(B xor F) or E
C	(B xor F) or E	H	F
D	E	I	C and D
E	not(J)	J	I

CGP is applied to the obtained truth table to generate the discrete model. The simplest solution in terms of the number of logic gates was chosen as the Boolean Model. This chosen model can be found in Table I.

The discrete model generated by CGP is used to determine the BooleCubes through the multivariate polynomial interpolation presented in Section III-C. In the sequence, the BooleCubes are converted to HillCubes. Finally, the temporal behavior of the continuous update functions is obtained using Equation 7. Considering this normalization, the obtained model in the form of an ODE system is

$$\frac{dA}{dt} = \left(1 - \frac{N(E)^{n_{AE}}}{(N(E)^{n_{AE}} + k_{AE}^{n_{AE}})} - N(A)\right)/\tau_A \quad (9)$$

$$\frac{dB}{dt} = \left(\frac{N(A)^{n_{BA}}}{(N(A)^{n_{BA}} + k_{BA}^{n_{BA}})} - N(B)\right)/\tau_B \quad (10)$$

$$\frac{dC}{dt} = \left(\frac{N(B)^{n_{CB}}}{(N(B)^{n_{CB}} + k_{CB}^{n_{CB}})} - N(C)\right)/\tau_C \quad (11)$$

$$\frac{dD}{dt} = \left(\frac{N(C)^{n_{DC}}}{(N(C)^{n_{DC}} + k_{DC}^{n_{DC}})} - N(D)\right)/\tau_D \quad (12)$$

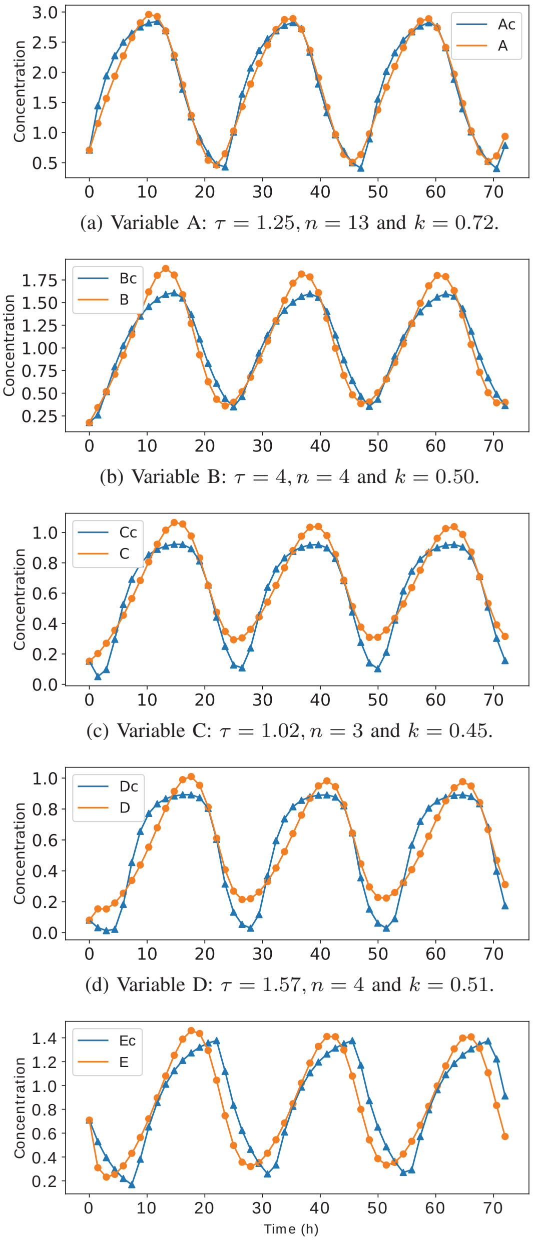
$$\frac{dE}{dt} = \left(\frac{N(D)^{n_{ED}}}{(N(D)^{n_{ED}} + k_{ED}^{n_{ED}})} - N(E)\right)/\tau_E \quad (13)$$

Finally, the numerical coefficients conclude the model and an ES is used to solve this problem. The numerical coefficients found by ES, and plots of the actual data and those values predicted by the obtained model are presented in Figure 4. This model achieved an error equal to 27.96.

B. Circadian Rhythm with 10 Variables

The original data of the circadian rhythm with 10 variables, the data discretized, and the state transition diagram generated using this data are presented in Figure 5. The variables of

[2] https://scipy.org/



(a) Variable A: $\tau = 1.25, n = 13$ and $k = 0.72$.

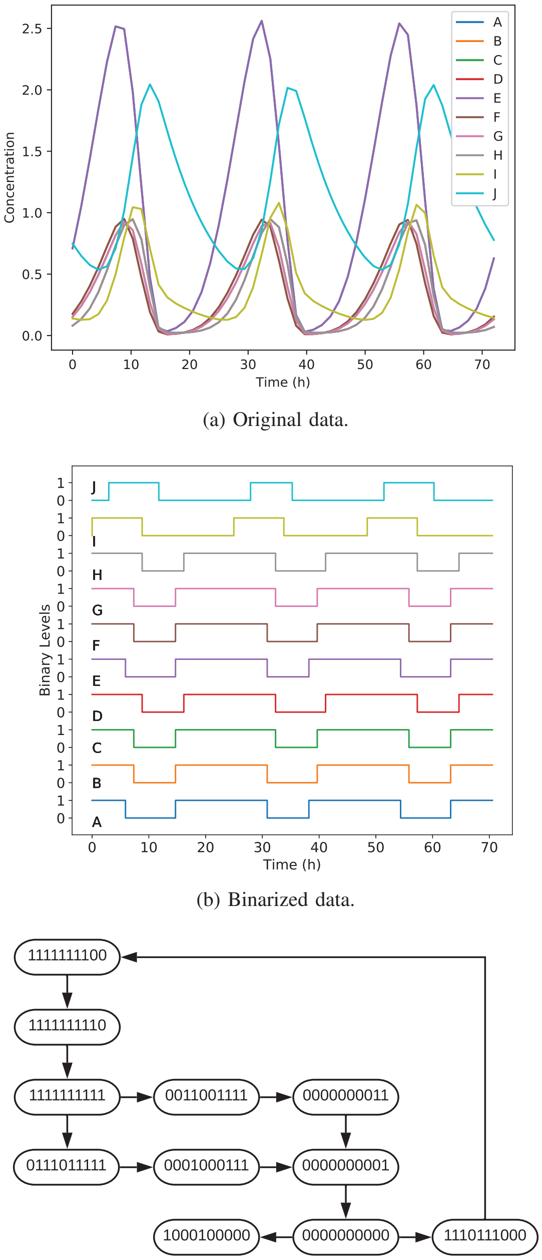
(b) Variable B: $\tau = 4, n = 4$ and $k = 0.50$.

(c) Variable C: $\tau = 1.02, n = 3$ and $k = 0.45$.

(d) Variable D: $\tau = 1.57, n = 4$ and $k = 0.51$.

(e) Variable E: $\tau = 3.43, n = 16$ and $k = 0.52$.

Fig. 4. Results for the Circadian Rhythm with 5 variables. Circle (orange) and triangle (blue) marks are the actual and the predicted values, respectively.



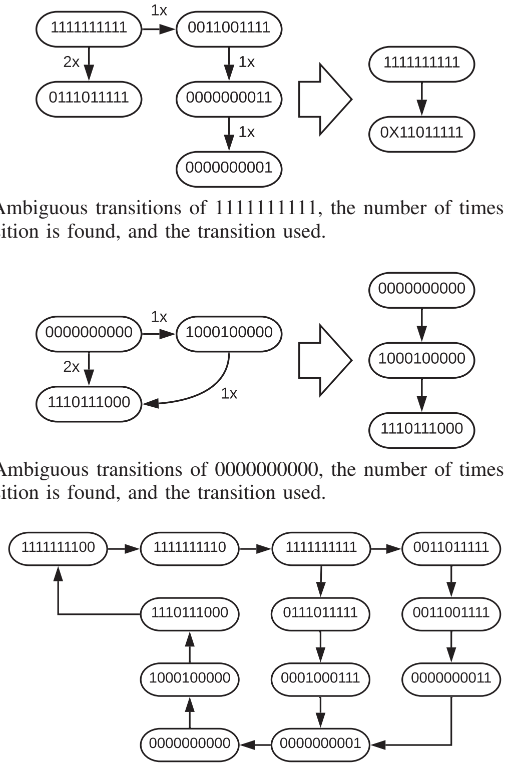
(a) Original data.

(b) Binarized data.

(c) Initial state transition diagram.

Fig. 5. Plots and initial state transition diagram – 10 Variables.

this problem are labeled as A, . . ., J. The states transition diagram is composed by 12 possible transitions (Figure 5c) and the maximum number of rows in the truth table is $2^{10} = 1024$. However, one can see in this problem that 2 states (1111111111 and 0000000000) produce more than one alternative states. As Boolean models considered here are deterministic, it is important to reduce these ambiguities in the state transition set. Also, as the data represents a sequence of activation/inactivation of the modeled phenomenon, the state transitions should form a cycle. Thus, any successor appears as a precursor of another state transition. In order to determine the unique resulting state for these 2 cases, we select the most frequent transition in the data. Figures 6a and 6b present the 2 two ambiguity, the number of times each state transition occurs, and the chosen state transition. It is important to highlight that the solution presented in Figure 6a still contains an ambiguous state transition. In this case, a don't care situation allows CGP to find the model that better represents the state transitions observed. Also, unlike the



(a) Ambiguous transitions of 1111111111, the number of times each transition is found, and the transition used.

(b) Ambiguous transitions of 0000000000, the number of times each transition is found, and the transition used.

(c) Final state transition diagram.

Fig. 6. Ambiguous cases and final state transition diagram – 10 Variables.

alternatives for the state 0000000000, both possible successors of 1111111111 exist in the state transition diagram. The final state transition diagram is presented in Figure 6c.

The states that are not present in Figure 6c and the ambiguity in the transition from state 1111111111 are treated here as irrelevant (don't care) situations in CGP. The expressions of the Boolean model obtained by the evolutionary process are presented in Table II. It is important to highlight the presence of more complex logic operators in this model, such as XOR.

The same procedures to determine the model in the form of an ODE system from the Boolean model and its numerical coefficients are applied to this problem. The error of this model is equal to 50.87. The plots of the actual and predicted values, and the numerical coefficients are presented in Figure 7. The continuous model created is not presented here due to the lack of space, but it can be found in the supplementary material.

V. Conclusions and Future Work

Systemic biology models can be used to test new hypotheses formulated using prior knowledge or data from experimentation. New hypotheses often arise in the form of a set of regulatory mechanisms. Recently, there has been a growing interest in the application of logic-based approaches in systemic biology, and advances in the estimation of gene regulatory networks (GRN) have led to a greater understanding of cellular regulation. However, additional methodological advances are still needed. We proposed here a procedure to infer models of GRN from data where (i) the data is binarized, (ii) a Boolean model is created via Cartesian Genetic Programming, (iii) a continuous model in the form of an ODE system is obtained from the Boolean model, and (iv) the numerical coefficients of the ODE system are optimized by an Evolution Strategy.

Boolean models are less susceptible to noise while ODE systems have a high degree of interpretability. Thus, combining both models may infer an accurate and symbolic GRN model, leading to interpretability. Also, it is expected that novel solutions, not commonly intelligible to humans, are explored using metaheuristics in the inference process.

In sum, our computational experiments indicate that the proposed method is capable of successfully generating GRN models. In addition, the obtained model in symbolic form can be used to extract biological knowledge. Finally, future work includes the evaluation of the proposed method in existing competitions. Moreover, we intend to model the GRNs of yeasts, which are highly explored in the literature.

In addition, we will further expand our proposal work with a higher level of discretization of gene expression data. We also intend to investigate the scalability of the proposed approach.

Acknowledgments

The authors thank CAPES, CNPq, FAPERJ, and FAPEMIG.

References

[1] R.-S. Wang, A. Saadatpour, and R. Albert, "Boolean modeling in systems biology: an overview of methodology and applications," *Physical biology*, vol. 9, no. 5, p. 055001, 2012.

[2] M. N. McCall, "Estimation of gene regulatory networks," *Postdoc journal: a journal of postdoctoral research and postdoctoral affairs*, vol. 1, no. 1, p. 60, 2013.

[3] H. De Jong, "Modeling and simulation of genetic regulatory systems: a literature review," *J. Comput. Biology*, vol. 9, no. 1, pp. 67–103, 2002.

[4] G. Sanguinetti *et al.*, "Gene regulatory network inference: an introductory survey," in *Gene Regulatory Networks*. Springer, 2019, pp. 1–23.

[5] C. A. Gallo, R. L. Cecchini, J. A. Carballido, S. Micheletto, and I. Ponzoni, "Discretization of gene expression data revised," *Briefings in bioinformatics*, vol. 17, no. 5, pp. 758–770, 2015.

[6] J. F. Miller, P. Thomson, and T. Fogarty, "Designing electronic circuits using evolutionary algorithms. arithmetic circuits: A case study," 1997.

[7] D. M. Wittmann, J. Krumsiek, J. Saez-Rodriguez, D. A. Lauffenburger, S. Klamt, and F. J. Theis, "Transforming boolean models to continuous models: methodology and application to t-cell receptor signaling," *BMC systems biology*, vol. 3, no. 1, p. 98, 2009.

[8] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies – a comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.

[9] T. Bäck, C. Foussette, and P. Krause, *Contemporary evolution strategies*. Springer, 2013.

[10] S. Watterson, S. Marshall, and P. Ghazal, "Logic models of pathway biology," *Drug discovery today*, vol. 13, no. 9-10, pp. 447–456, 2008.

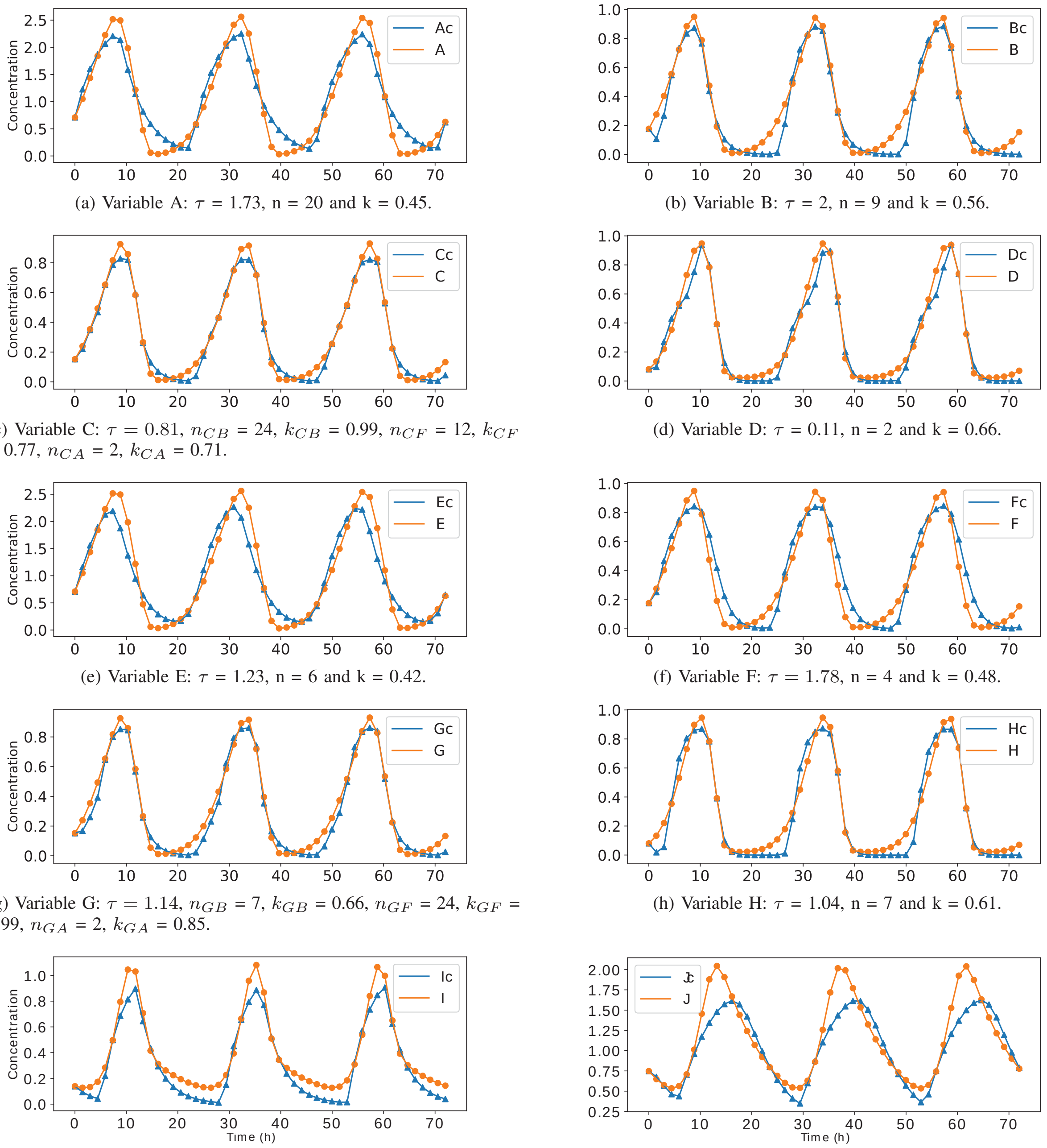
[11] D. Irons, "Logical analysis of the budding yeast cell cycle," *Journal of theoretical biology*, vol. 257, no. 4, pp. 543–559, 2009.

[12] G. Karlebach and R. Shamir, "Modelling and analysis of gene regulatory networks," *Nat. Rev. Molecular Cell Biology*, vol. 9, no. 10, p. 770, 2008.

[13] E. Sakamoto and H. Iba, "Inferring a system of differential equations for a gene regulatory network by using genetic programming," in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)*, vol. 1. IEEE, 2001, pp. 720–726.

[14] A. Sîrbu, H. J. Ruskin, and M. Crane, "Comparison of evolutionary algorithms in gene regulatory network model inference," *BMC bioinformatics*, vol. 11, no. 1, p. 59, 2010.

[15] C. Becquet, S. Blachon, B. Jeudy, J.-F. Boulicaut, and O. Gandrillon, "Strong-association-rule mining for large-scale gene-expression data analysis: a case study on human sage data," *Genome Biology*, vol. 3, no. 12, pp. research0067–1, 2002.



(a) Variable A: τ = 1.73, n = 20 and k = 0.45.

(b) Variable B: τ = 2, n = 9 and k = 0.56.

(c) Variable C: $\tau = 0.81$, n_{CB} = 24, k_{CB} = 0.99, n_{CF} = 12, k_{CF} = 0.77, n_{CA} = 2, k_{CA} = 0.71.

(d) Variable D: τ = 0.11, n = 2 and k = 0.66.

(e) Variable E: τ = 1.23, n = 6 and k = 0.42.

(f) Variable F: $\tau = 1.78$, n = 4 and k = 0.48.

(g) Variable G: $\tau = 1.14$, n_{GB} = 7, k_{GB} = 0.66, n_{GF} = 24, k_{GF} = 0.99, n_{GA} = 2, k_{GA} = 0.85.

(h) Variable H: τ = 1.04, n = 7 and k = 0.61.

(i) Variable I: $\tau = 3.47$, n_{IG} = 21, k_{IG} = 0.55, n_{IH} = 20 and k_{IH} = 0.46.

(j) Variable J: τ = 3.21, n = 3 and k = 0.17.

Fig. 7. Results for the Circadian Rhythm with 10 variables. Circle (orange) and triangle (blue) marks are the actual and the predicted values, respectively.

[16] R. G. Pensa, C. Leschi, J. Besson, and J.-F. Boulicaut, "Assessment of discretization techniques for relevant pattern discovery from gene expression data," in *Proc. of the International Conference on Data Mining in Bioinformatics*. Springer, 2004, pp. 24–30.

[17] J. F. Miller, "Cartesian genetic programming," *CGP*, pp. 17–34, 2011.

[18] A. E. Eiben, J. E. Smith *et al.*, *Introduction to evolutionary computing*. Springer, 2003, vol. 53.

[19] H. S. Bernardino and H. J. C. Barbosa, "Comparing two ways of inferring a differential equation model via grammar-based immune programming," in *Proc. of the Iberian-Latin-American Congress on Computational Methods in Engineering*, 2010.

[20] ——, "Inferring systems of ordinary differential equations via grammar-based immune programming," in *Int. Conf. on Art. I.S.* Springer, 2011, pp. 198–211.

[21] A. Goldbeter, "A model for circadian oscillations in the drosophila period protein (per)," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 261, no. 1362, pp. 319–324, 1995.

[22] J.-C. Leloup and A. Goldbeter, "A model for circadian rhythms in drosophila incorporating the formation of a complex between the per and tim proteins," *Journal of biological rhythms*, vol. 13, no. 1, pp. 70–87, 1998.

[23] J. Krumsiek, D. M. Wittmann, and F. J. Theis, "From discrete to continuous gene regulation models–a tutorial using the odefy toolbox," *Ap. of MATLAB in Science and Eng.*, p. 35, 2011.