BIOINFORMATICS ORIGINAL PAPER

Vol. 26 no. 9 2010, pages 1239–1245
doi:10.1093/bioinformatics/btq120

Systems biology

Advance Access publication March 19, 2010

Identification of genetic network dynamics with unate structure

Riccardo Porreca[1], Eugenio Cinquemani[2,*], John Lygeros[1] and Giancarlo Ferrari-Trecate[3]

[1]Institut für Automatik, ETH Zürich, 8092 Zürich, Switzerland, [2]INRIA Grenoble-Rhône-Alpes, Montbonnot, 38334 Saint-Ismier Cedex, France and [3]Dipartimento di Informatica e Sistemistica, Università degli Studi di Pavia, 27100 Pavia, Italy

Associate Editor: Trey Ideker

ABSTRACT

Motivation: Modern experimental techniques for time course measurement of gene expression enable the identification of dynamical models of genetic regulatory networks. In general, identification involves fitting appropriate network structures and parameters to the data. For a given set of genes, exploring all possible network structures is clearly prohibitive. Modelling and identification methods for the a priori selection of network structures compatible with biological knowledge and experimental data are necessary to make the identification problem tractable.

Results: We propose a differential equation modelling framework where the regulatory interactions among genes are expressed in terms of unate functions, a class of gene activation rules commonly encountered in Boolean network modelling. We establish analytical properties of the models in the class and exploit them to devise a two-step procedure for gene network reconstruction from product concentration and synthesis rate time series. The first step isolates a family of model structures compatible with the data from a set of most relevant biological hypotheses. The second step explores this family and returns a pool of best fitting models along with estimates of their parameters. The method is tested on a simulated network and compared with state-of-the-art network inference methods on the benchmark synthetic network IRMA.

Contact: eugenio.cinquemani@inria.fr

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

Received on September 30, 2009; revised on February 4, 2010; accepted on March 16, 2010

1 INTRODUCTION

Identification of genetic regulatory networks aims at inferring the regulatory mechanisms of gene expression from experimental data. Various modelling approaches have been employed with success for the reconstruction of the network of interactions among genes (see e.g. Bansal *et al.*, 2007; Gardner and Faith, 2005; Markowetz and Spang, 2007, for recent reviews). In particular, Boolean activation rules (Kauffman, 1969) have been proposed as a natural framework for the description and reconstruction of gene activation rules, whereas Bayesian networks (Nachman *et al.*, 2004; Nagarajan *et al.*, 2004; Segal *et al.*, 2003) have been utilized to capture statistical relationships in the expression of a network of genes.

The development of experimental techniques for the quantitative monitoring of gene expression over time is paving the way for the learning of gene expression dynamics. Kinetic modelling, where the interactions among genes are encoded into the structure of a set of gene expression rates (see de Jong, 2002, and references therein), provides an accurate description of the time evolution of gene expression. Unfortunately, an overwhelming variety of gene activation functions must be explored in order to reconstruct the network structure and dynamics from the data. A way around this complexity is to quantify genetic interactions by universal approximators. In Jaeger *et al.* (2004), the activation level of each gene is quantified by a saturated linear combination of the concentrations of the network proteins. This enables the reconstruction of direction and sign (inhibition versus activation) of the interactions among genes. Unlike in Boolean networks, the interplay among controlling genes in the regulation of the controlled gene (the ‘logics’ of the network) is not modelled explicitly. Similar insight into regulatory interactions near equilibria is provided by the linearization methods (Bansal *et al.*, 2007; Cinquemani *et al.*, 2009; Gardner *et al.*, 2003; Zavlanos *et al.*, 2008). Glass and Kauffman (1973) suggested to model switch-like regulatory interactions by step functions. This results into a very simple piecewise linear model of gene product concentration kinetics (constant synthesis rate plus degradation) that can be fitted to experimental data (Cinquemani *et al.*, 2008; Drulhe *et al.*, 2008; Porreca *et al.*, 2008) quite efficiently. For many activation functions, however, the approximation by step functions is rather coarse and may prevent the use of this framework for identification.

In this work, we address the identification of kinetic models of gene regulatory networks from time course gene expression data. Our primary interest is the reconstruction of the network of interactions and logics behind gene expression control. We propose a modelling framework where the synthesis rate equations reflect the structure of a class of gene activation rules known as unate functions (Aracena, 2008). In the context of Boolean modelling, unate functions capture all interactions where each gene acts exclusively as an inducer or as an inhibitor for the expression of another gene. Based on biochemical analysis, Grefenstette *et al.* (2006) argue that unate functions provide a comprehensive modelling framework for genetic regulatory networks. Unate functions include hierarchically (or nested) canalizing functions (Jarrah *et al.*, 2007), a class of Boolean functions describing the majority of the known gene activation rules (Kauffman *et al.*, 2004; Nikolajewa *et al.*, 2007; Szallasi and Liang, 1998). Canalizing models are used for gene network inference by Laubenbacher and Stigler (2004), Raeymaekers (2002)

*To whom correspondence should be addressed.

and Akutsu *et al.* (2000). We establish properties of the rate equations with unate structure and exploit them to set up an identification procedure in two steps. The first step selects a restricted family of gene interaction patterns consistent with the experimental data. The second step explores kinetic models with compatible structure and returns a pool of kinetic models of lowest complexity that explain the data in a statistical sense. This procedure ensures that a reduced number of models need to be fitted to the data, leading to substantial computational savings. Still, the number of model structures compatible with the data may be enormous in practice. To cope with this, we show by a relevant example how the method can be adapted to confine the search of model structures to the most relevant biological hypotheses. We assume that gene product concentration and synthesis rate measurements are available simultaneously at discrete time instants. Various direct of indirect methods providing this data exists (see for instance Brown and Lostroh, 2008; Ronen *et al.*, 2002). The performance of the method is first tested on a simulated system and then compared with state-of-the-art on the network IRMA (*In vivo* assessment of Reverse-engineering and Modeling Approaches), a synthetic network designed by Cantone *et al.* (2009) as a benchmark for gene network inference algorithms.

2 METHODS

2.1 Kinetic models with unate structure

Consider a network with n genes. In the context of Boolean networks, the activation status of gene i, with $i=1,\dots,n$, is encoded by a binary variable X_i that is 1 if the gene is active and 0 otherwise. The laws governing the activation of gene i are captured by a Boolean rule $B_i(X):\{0,1\}^n \to \{0,1\}$, with $X=(X_1,\dots,X_n)$. In practice, B_i depends only on the entries of X corresponding to the genes that control the expression of gene i. Grefenstette *et al.* (2006) argued on the basis of biochemical reaction modelling that virtually all regulatory interactions can be described by unate functions. Also known as sign-definite functions, these are Boolean functions that are either non-decreasing or non-increasing in each of the input variables. Every unate function can be written in conjunctive normal form by an expression where each variable X_i appears either in a positive form (X_i itself) or in a negative form (the negation of X_i, indicated by $\neg X_i$), but not both. That is, it can be written as

$$B_i(X)=\bigwedge_{l=1}^{n_i} T_l(X), \qquad T_l(X)=\bigvee_{j\in J_l} \tilde{X}_j, \tag{1}$$

where '$\wedge$' and '$\vee$' stand for conjunction ('and') and disjunction ('or'), respectively, each J_l is a non-empty set of pairwise different indices from $\{1,\dots,n\}$, and each variable $\tilde{X}_j$ is uniquely defined as either X_j or $\neg X_j$. By convention, a conjunction of $n_i=0$ terms is equal to 1. A theoretical investigation of the properties of Boolean regulatory networks based on unate activation functions is developed by Aracena (2008). Unate functions include, among others, the class of hierarchically canalizing functions (HCF; Aracena, 2008). According to Kauffman *et al.* (2004) and Szallasi and Liang (1998), HCF capture a large class of the known regulatory interactions among genes and are intimately related with the stability properties of the network.

Let $x_i \in \mathbb{R}_{\geq 0}$ denote the concentration of the product of gene i, and let $x=\left(x_1,\dots,x_n\right)$. We consider ordinary differential equation models describing the evolution of x as follows (de Jong, 2002): for $i=1,\dots,n$,

$$\dot{x}_i = g_i(x)-\gamma_i(x), \tag{2}$$

where $g_i(x)\geq 0$ and $\gamma_i(x)\geq 0$ are the synthesis and the degradation rates of the product of gene i. The gene network identification methods that we will discuss rely on sample observations of x and corresponding $g_i(x)$. As long as these data are available, the decay rate function $\gamma_i(x)$ does not enter the problem and will be ignored. We focus on non-linear models of $g_i(x)$ of the type

$$g_i(x)=\kappa_{0,i}+\kappa_{1,i} b_i(x), \tag{3}$$

where $\kappa_{0,i}\in\mathbb{R}_{\geq 0}$ and $\kappa_{1,i}\in\mathbb{R}_{\geq 0}$ are constants and $b_i(x):\mathbb{R}^n_{\geq 0}\to[0,1]$ quantifies the regulatory effects of the gene products on the expression of gene i via Hill activation functions (Keller, 1995; Yang *et al.*, 2007):

$$\sigma^+(x_j)=\frac{x_j^d}{x_j^d+\eta^d}, \qquad \sigma^-(x_j)=1-\sigma^+(x_j)=\frac{\eta^d}{x_j^d+\eta^d},$$

where $d\geq 1$ is a cooperativity coefficient and $\eta>0$ is a threshold parameter. We propose a modelling framework where the unate structure of gene activation functions is reflected into the algebraic structure of $b_i(x)$. Given an activation function (1), we obtain the corresponding $b_i(x)$ by the following transformation rules. For parameters d and η possibly depending on i, we replace each X_i by $\sigma^+(x_i)$. Given any two functions $\tau(x)$ and $\tau'(x)$ representing the Boolean expressions $T(X)$ and $T'(X)$, we encode $\neg T(X)$ by $1-\tau(x)$ and $T(X)\wedge T'(X)$ by $\tau(x)\cdot\tau'(x)$. With these rules, Equation (1) is transformed into (Supplementary Material)

$$b_i(x)=\prod_{l=1}^{n_i}\tau_l(x), \qquad \tau_l(x)=1-\prod_{j\in J_l}\left(1-\sigma^{\pm}(x_j)\right), \tag{4}$$

where $\sigma^{\pm}(x_j)=\sigma^+(x_j)$ if $\tilde{X}_j=X_j$ and $\sigma^{\pm}(x_j)=\sigma^-(x_j)$ if $\tilde{X}_j=\neg X_j$. By convention, a product of zero terms, i.e. $n_i=0$, is equal to 1. According to Plahte *et al.* (1998), this is the algebraic counterpart of (1) if x_j low is interpreted as $X_j=0$ and x_j high is interpreted as $X_j=1$. We will refer to (2–4) as a kinetic model with unate structure.

2.2 Hierarchies of consistent models

We show that kinetic models with unate structure possess monotonicity properties that are independent of the model parameters and of the decay rates $\gamma_i(x)$. These properties can be exploited in order to accept or reject families of model structures based on a qualitative analysis of experimental data. Since the results apply equally to any fixed index i, in the remainder of the section we will drop index i from g_i. We define the *sign pattern* of (3–4) to be the n-tuple $p=(p_1,\dots,p_n)\in\{-1,0,+1\}^n$, where, for $j=1,\dots n$,

$$p_j=\begin{cases} 0, & \text{if } j\notin J_l,\ l=1,\dots,n_i,\\ 1, & \text{if } \sigma^{\pm}(x_j)=\sigma^+(x_j),\\ -1, & \text{if } \sigma^{\pm}(x_j)=\sigma^-(x_j).\end{cases}$$

Note that many different functions of the form (4) share the same sign pattern. We will write $g(x|p)$ in place of $g(x)$ to specify a synthesis rate with sign pattern p. The *complexity* $C(p)$ of a sign pattern p is defined as the number of non-zero entries of p, and is equal to the number of effective inputs of $g(x|p)$. Given any sign pattern p with $C(p)>0$ and any two concentration vectors x^1 and x^2, it holds that (Supplementary Material)

$$\left[p_j(x_j^2-x_j^1)\geq 0,\ j=1,\dots,n\right]\Rightarrow\left[g(x^2|p)-g(x^1|p)\geq 0\right]. \tag{5}$$

In words, if all the elements of x move in the direction of growth (defined by p) of the corresponding sigmoid in $g(x|p)$, then $g(x|p)$ is bound to increase. In view of identification, consider a set of m concentration measurements x^k and corresponding synthesis rates $g^k=g(x^k)$, with $k=1,\dots,m$. A sign pattern p is called *inconsistent* with the data if there exist two data points (x^k,g^k) and (x^l,g^l), with $k,l\in\{1,\dots,m\}$, for which (5) is violated, i.e.

$$\left[p_j(x_j^k-x_j^l)\geq 0,\ j=1,\dots,n\right] \text{ and } \left[g^k-g^l<0\right]. \tag{6}$$

A pattern that is not inconsistent is called *consistent* with the data. A sign pattern p' is called a *subpattern* of p (and p is a superpattern of p') if all its non-zero entries are equal to the corresponding entries of p. We indicate this fact by the notation $p' \sqsubseteq p$. It is easily seen that in this case $C(p') \leq C(p)$ and subpatterns of inconsistent sign patterns are also inconsistent. Conversely, superpatterns of consistent sign patterns are also consistent. Let us compute a set $\bar{P}$ as follows.

Computation of $\bar{P}$: set $\bar{P} = \emptyset$. For all indices $k, l \in \{1, \ldots, m\}$:

(I) If $g^k - g^l < 0$, define the sign pattern $\bar{p} = (\bar{p}_1, \ldots, \bar{p}_n)$ by setting $\bar{p}_j = \text{sign}(x_j^k - x_j^l)$, with $j = 1, \ldots, n$, and include $\bar{p}$ in $\bar{P}$.

[By convention $\text{sign}(0) = 0$]. It follows from above that all sign patterns in $\bar{P}$ are inconsistent. Moreover, it is shown in the Supplementary Material that every inconsistent sign pattern p is a subpattern of at least one pattern $\bar{p} \in \bar{P}$. It is possible to determine a set P^* of minimal consistent sign patterns, such that every consistent pattern is a superpattern of at least one sign pattern in P^*. In light of these properties, we will denote the hierarchy of consistent sign patterns by $\mathscr{H}(P^*)$. Given $\bar{P}$, the set P^* of minimal consistent sign patterns can be computed by the following procedure.

Computation of P^*: define $\bar{\ell} = \max\{C(\bar{p}) : \bar{p} \in \bar{P}\}$. Initialize $P^* = \emptyset$. For increasing values of complexity $\ell = 0, \ldots, \min\{n, \bar{\ell} + 1\}$:

(II) Generate all patterns p of complexity ℓ. For each such p,

(III) Check if p is consistent by verifying that there is no $\bar{p} \in \bar{P}$ such that $p \sqsubseteq \bar{p}$. If this is the case,

(IV) Check if p is minimal consistent by verifying that there is no $p^* \in P^*$ such that $p^* \sqsubseteq p$. If this is the case, include p in P^*.

The complexity of the computation of $\bar{P}$ is $\mathscr{O}(m^2)$. In the current implementation, the complexity of the computation of P^* from $\bar{P}$ is $\mathscr{O}(3^{\bar{\ell}})$, with $\bar{\ell} \leq n$. The correctness of the algorithm is proven in the Supplementary Material. In practice, the procedure will be applied to noisy data. The necessary extensions are presented in the next section.

2.3 Structure and parameter identification

We exploit the properties of kinetic models with unate structure to perform model identification in two steps. In a first step, data is used to select a hierarchy $\mathscr{H}$ of consistent sign patterns. In a second step, we fit models of increasing complexity with sign pattern in $\mathscr{H}$ in order to build a pool $\mathscr{P}$ of models explaining the data with sufficient accuracy. The method requires simultaneous measurements of gene product concentrations and synthesis rates at m time instants $t_1, \ldots, t_m$, and a quantification of the statistics of the measurement noise. We assume that data obeys the following measurement model: for $k = 1, \ldots, m$,

$$\tilde{x}_i^k = x_i^k + e_i^k, \qquad \tilde{g}_i^k = g_i^k + \epsilon_i^k, \tag{7}$$

where $\tilde{x}_i^k$ and $\tilde{g}_i^k$ are noisy observations of $x_i^k = x_i(t_k)$ and $g_i^k = g\big(x(t_k)\big)$, respectively, while e_i^k and ϵ_i^k are mutually uncorrelated Gaussian random variables with zero mean and variance $v_e(x_i^k) = \text{var}(e_i^k)$ and $v_\epsilon(g_i^k) = \text{var}(\epsilon_i^k)$ possibly depending on x_i^k and g_i^k. We assume that the functions $v_e(x_i)$ and $v_\epsilon(g_i)$ are known. This measurement model includes, for instance, additive noise models, where v_e and v_ϵ are positive constants, and multiplicative noise models (see e.g. Kreutz *et al.*, 2007), where $v_e(x_i)$ and $v_\epsilon(g_i)$ are linear functions of x_i^2 and g_i^2.

2.3.1 *Outline of the algorithm*

Given a sign pattern p, let $S(p)$ be a set of admissible structures for a model $g(x|p)$. Under the assumption that $g(x|p)$ has a unate structure, each element $s \in S(p)$ determines the family of index sets $J_1, \ldots, J_{n_i}$ of Equation (4) and, in accordance with p, the sign of the sigmoids $\sigma^{\pm}(x_j)$, with $j = 1, \ldots, n$. For a model (4) with structure s, the parameters $\kappa_{0,i}$, $\kappa_{1,i}$, and the cooperativity and threshold parameters of all the sigmoids in the model will be collectively denoted by θ. Given data $(\tilde{x}^k, \tilde{g}_i^k)$, with $k = 1, \ldots, m$, and values $N > 0$ and $\alpha \in (0, 1)$ specified by the user, we perform identification by the following algorithm, executed separately for each gene $i = 1, \ldots, n$. (The definition of several new quantities, including N and α, will be discussed shortly.)

Algorithm 1 Two-step identification.

Step 1. (Selection of consistent model structures)

I. Set $\bar{P} = \emptyset$. For all indices $k, l \in \{1, \ldots, m\}$, if $\tilde{g}_i^k - \tilde{g}_i^l < -N\sigma_{g_i}^{k,l}$ then define $\bar{p} = (\bar{p}_1, \ldots, \bar{p}_n)$ by

$$\bar{p}_j = \begin{cases} -1, & \text{if } \tilde{x}_j^k - \tilde{x}_j^l \leq -N\sigma_{x_j}^{k,l}, \\ 1, & \text{if } \tilde{x}_j^k - \tilde{x}_j^l \geq N\sigma_{x_j}^{k,l}, \\ 0, & \text{otherwise}, \end{cases} \qquad j = 1, \ldots, n,$$

and include $\bar{p}$ in $\bar{P}$.

II–IV. Execute the computation of P^* from the resulting $\bar{P}$, as described in Section 2.2.

Step 2. (Identification of best consistent models) Set $\mathscr{P} = \emptyset$. Define $\ell^* = \min\{C(p^*) : p^* \in P^*\}$. For $\ell = \ell^*$ to n:

V. Generate patterns p such that $C(p) = \ell$ and $p^* \sqsubseteq p$ for some $p^* \in P^*$. For each such p, execute VI.

VI. For all $s \in S(p)$, fit the model $g_i(\cdot)$ with sign pattern p and structure s by solving the nonlinear regression problem

$$\delta = \min_{\theta} \sum_{k=1}^{m} w_k \big(\tilde{g}_i^k - g_i(\tilde{x}^k)\big)^2. \tag{8}$$

If $\delta < \tau(\alpha)$, include the fitted model in $\mathscr{P}$.

VII. If $\mathscr{P} \neq \emptyset$ return $\mathscr{P}$ and exit.

2.3.2 *Discussion of the algorithm*

Step 1 computes the minimal consistent sign patterns. The procedure is an adaptation of Steps I–IV of Section 2.2 to allow for the presence of measurement noise. For the various indices $l, k \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$, we employ standard statistical hypothesis testing for the mean of Gaussian random variables in order to evaluate the signs of $g_i^k - g_i^l$ and $x_j^k - x_j^l$, on the basis of noisy data. For a given $N > 0$ (typically integer), the hypothesis $g_i^k - g_i^l < 0$ is accepted if and only if $\tilde{g}_i^k - \tilde{g}_i^l < -N\sigma_{g_i}^{k,l}$, where $\sigma_{g_i}^{k,l}$ is the standard deviation of $\tilde{g}_i^k - \tilde{g}_i^l$. On the basis of (7), we make the approximation $\sigma_{g_i}^{k,l} = \sqrt{v_\epsilon(\tilde{g}_i^k) + v_\epsilon(\tilde{g}_i^l)}$. Likewise, for $\sigma_{x_j}^{k,l} = \sqrt{v_e(\tilde{x}_j^k) + v_e(\tilde{x}_j^l)}$, we accept the hypothesis $x_j^k - x_j^l < 0$ (i.e. set $\bar{p}_j = -1$) if $\tilde{x}_j^k - \tilde{x}_j^l \leq -N\sigma_{x_j}^{k,l}$, the hypothesis $x_j^k - x_j^l > 0$ ($\bar{p}_j = 1$) if $\tilde{x}_j^k - \tilde{x}_j^l \geq N\sigma_{x_j}^{k,l}$ and the hypothesis $x_j^k - x_j^l = 0$ ($\bar{p}_j = 0$) otherwise. The result of Step 1 is a set of minimal consistent sign patterns P^*. The smaller N, the smaller $\mathscr{H}(P^*)$, at the price of an increased probability of excluding the true sign pattern from $\mathscr{H}(P^*)$.

In Step 2, we seek models with structure compatible with $\mathscr{H}(P^*)$ that explain the data with sufficient accuracy. The search is conducted by increasing levels of complexity ℓ, starting from the simplest models ($\ell = \ell^*$) with structure compatible with $\mathscr{H}(P^*)$ onwards, and is stopped at the level of complexity where at least one good model is found. The parameters θ of a candidate model $g(\cdot)$ with sign pattern p and structure s are estimated by regression (8). The choice of the weights w_k is determined by the statistics of the fitting errors $\tilde{g}_i^k - g_i(\tilde{x}^k)$. Under the null hypothesis that structure and parameters of g_i are correct, $\tilde{g}_i^k - g_i(\tilde{x}^k)$ is approximately Gaussian with mean 0 and covariance $\sigma^2(x^k, g_i^k) = G(x^k)^T \text{diag}\big(v_e(x_1^k), \ldots, v_e(x_n^k)\big) G(x^k) + v_\epsilon(g_i^k)$, where $G(x)$ is the column vector given by $G(x) = \partial g_i(x)/\partial x$. Therefore, the choice $w_k = \big(\sigma^2(\tilde{x}^k, \tilde{g}_i^k)\big)^{-1}$ guarantees that the error residuals are appropriately weighted by the inverse of the corresponding noise levels. In addition, under the same null hypothesis, this choice approximately yields $\delta \sim \chi^2(m - |\theta|)$ (chi-square distributed with $m - |\theta|$ degrees of freedom, see the Supplementary Material). This fact is used to set the model acceptance threshold $\tau(\alpha)$. For a suitable $\alpha \in (0,1)$, we choose $\tau(\alpha) = F^{-1}_{m-|\theta|}(\alpha)$, where F^{-1}_m is the inverse of a chi-square distribution with m degrees of freedom. Then, with confidence level α, we reject the hypothesis that a model $g_i(\cdot)$ with residual fitting error δ is a satisfactory description of the data if $\delta \geq \tau(\alpha)$, otherwise we accept the model and stop the iterations of the identification procedure at the current level of complexity. This mechanism favours simple models over complicated ones, reducing the risk of overfitting: the search of acceptable models proceeds by increasing levels of complexity ℓ and halts at the level for which at least one model is found. Several models of the same complexity may be accepted. If no good model exists, the procedure terminates at the maximum level of complexity n returning the empty set $\mathscr{P} = \emptyset$. Alternatively, a smaller upper bound to the complexity of the models searched can be placed by the user in the light of biological considerations.

For the special circumstance where $b_i(x) \equiv 0$ (no effective inputs), it is convenient to introduce a preprocessing step that checks if a constant model explaining the data well enough exists. We do this by a standard chi-square test with confidence level α:

Step 0. (Check for trivial dynamics) For $w_1 = \cdots = w_m = 1$, solve regression (8) with respect to $g_i(x) \equiv \kappa_{0,i} = \theta$. If $\delta < v_\epsilon(\kappa_{0,i}) \cdot F^{-1}_{m-1}(\alpha)$, then return the model $g_i(x) = \kappa_{0,i}$ and exit. Otherwise execute Algorithm 1.

2.3.3 Restriction of the search space. In practice, searching all unate structures $S(p)$ associated with a sign pattern p is prohibitive, especially for large values of complexity $C(p)$. A possible remedy is to confine the search to sign patterns having limited complexity (i.e. only a limited number of genes may have a direct regulatory effect on the expression of another gene). Another sensible approach is to reduce the set of model structures $S(p)$ associated with each sign pattern p. Based on a priori information on the nature of the interactions, separately for every gene $i = 1,\ldots,n$, identification can be focused on the most relevant models of regulation.

For example, Nikolajewa *et al.* (2007) note that many gene activation rules are unate functions (in fact HCF) in one of the following forms:

$$B_i(X) = \begin{cases} \tilde{X}_{j_1} \wedge \tilde{X}_{j_2} \wedge \tilde{X}_{j_3} \wedge \cdots \wedge \tilde{X}_{j_\ell} & \text{or} \\ [\tilde{X}_{j_1} \vee \tilde{X}_{j_2}] \wedge \tilde{X}_{j_3} \wedge \cdots \wedge \tilde{X}_{j_\ell}, \end{cases} \tag{9}$$

where ℓ is the number of effective inputs of $B_i(X)$ and $j_1,\ldots,j_\ell$ are pairwise different indices from the set $\{1,\ldots,n\}$. Both expressions are in the form (1). Following Section 2.1, the algebraic counterpart of (9) becomes

$$b_i(x) = \begin{cases} \sigma^{\pm}(x_{j_1})\sigma^{\pm}(x_{j_2})\sigma^{\pm}(x_{j_3})\cdots\sigma^{\pm}(x_{j_\ell}) & \text{or} \\ \left[1-\left(1-\sigma^{\pm}(x_{j_1})\right)\left(1-\sigma^{\pm}(x_{j_2})\right)\right]\times \\ \qquad\qquad \sigma^{\pm}(x_{j_3})\cdots\sigma^{\pm}(x_{j_\ell}), \end{cases} \tag{10}$$

For the purpose of exemplification, in our identification experiments we will restrict ourselves to models in form (10), which will be referred to as kinetic models with $\mathscr{S}$-structure,

2.3.4 Performance indices. In order to provide an evaluation of our identification method, we propose suitable indices of performance for Steps 1 and 2 based on repeated identification experiments. Let p^* and s^* denote the sign pattern and the structure of the true model. Let $\mathscr{H}^r$ and $\mathscr{P}^r$, with $r = 1,\ldots,M$, denote the hierarchy of consistent sign patterns and the pool of identified models, respectively, computed by the r-th of M identification experiments. We shall write $s \in \mathscr{P}^r$ to denote that $\mathscr{P}^r$ contains a model with structure s. For Step 1, we define the reliability index $R = |\{r : p^* \in \mathscr{H}^r\}|/M$ ($|\cdot|$ denotes set cardinality) and the selectivity index

$$S = 1 - \frac{1}{MR} \sum_{r:\, p^* \in \mathscr{H}^r} \frac{|\{p \in \mathscr{H}^r :\, C(p) \leq C(p^*)\}|}{|\{p :\, C(p) \leq C(p^*)\}|}.$$

$R \in [0,1]$ is the relative frequency of p^* falling in the set of consistent sign patterns: the larger R, the more reliable the procedure. When p^* is deemed consistent, $S \in [0,1]$ counts the number of patterns that need not be explored in Step 2 thanks to Step 1, relative to the number of patterns that would be explored in Step 2 in absence of Step 1, under the assumption that the model acceptance criterion in Step 2 is perfect. Therefore, $S \in [0,1]$ quantifies the computational savings provided by Step 1: the larger S, the more significant the saving. For Step 2, we define the accuracy index $A = |\{r : s^* \in \mathscr{P}^r\}|/M$ and the dispersion index $D = \left(\sum_{r:s^* \in \mathscr{P}^r} |\mathscr{P}^r|\right)/MA$. $A \in [0,1]$ is the relative frequency of s^* being found in the pool of identified models: the higher A, the more effective the identification. When s^* is found in the pool of identified models, $D \geq 1$ counts how many models are included in the pool on an average: the smaller the D, the more accurate the results.

3 RESULTS AND DISCUSSION

In this section, we will discuss the identification of two regulatory networks by the algorithm in Section 2. The first is an *in silico* network specifically designed for testing the performance of the identification method. The second is IRMA, a synthetic network engineered by Cantone *et al.* (2009) in *Saccharomyces cerevisiae* cells and proposed as a benchmark for the comparison of reverse engineering algorithms. The identification algorithm was implemented in MATLAB and optimization (8) was performed by the standard MATLAB procedure `fmincon`.

3.1 Performance test on a repressilator

We considered a network of six genes where three genes form a core control loop in which a single gene represses the expression of the next gene. This portion of the network is usually called *repressilator* and was first synthesized by Elowitz and Leibler (2000). It induces oscillations in the product concentration and activation levels of each of the genes. The expression of the remaining three genes is activated according to various functions of the product concentrations of the core genes. The network is governed by the following equations (for a graphical representation of the network refer the Supplementary Material):

$$\dot{x}_1 = \kappa_{0,1} + \kappa_{1,1}\sigma^{-}(x_3) - \gamma_1 x_1, \tag{11}$$

$$\dot{x}_2 = \kappa_{0,2} + \kappa_{1,2}\sigma^{-}(x_1) - \gamma_2 x_2, \tag{12}$$

$$\dot{x}_3 = \kappa_{0,3} + \kappa_{1,3}\sigma^{-}(x_2) - \gamma_3 x_3, \tag{13}$$

$$\dot{x}_4 = \kappa_{0,4} + \kappa_{1,4}\sigma^{-}(x_1)\sigma^{+}(x_2) - \gamma_4 x_4, \tag{14}$$

$$\dot{x}_5 = \kappa_{0,5} + \kappa_{1,5}[1 - \sigma^{+}(x_2)\sigma^{-}(x_3)] - \gamma_5 x_5, \tag{15}$$

$$\dot{x}_6 = \kappa_{0,6} + \kappa_{1,6}[1 - \sigma^{+}(x_2)\sigma^{+}(x_3)]\sigma^{+}(x_1) - \gamma_6 x_6. \tag{16}$$

The parameters and initial conditions for this system are reported in the Supplementary Material. In particular, the cooperativity coefficients of the sigmoids are assumed known and equal to 2.1 (Elowitz and Leibler, 2000).

We attempted identification of this system with 90 equally spaced data points over a time interval such that the product concentrations of the core genes complete three full oscillations. Measurements $\tilde{x}^k_i$ and $\tilde{g}^k_i$ were artificially corrupted by Gaussian noise samples according to the observation model (7), with $v_e(x^k_i) = (\sigma_e x^k_i)^2$ and $v_\epsilon(g^k_i) = (\sigma_\epsilon g^k_i)^2$, for the different noise levels $\sigma_e = \sigma_\epsilon = 0.01, 0.03, 0.05, 0.07$. This corresponds to noise roughly within 3%, 10%, 15% and 20% of the actual values of x^k_i and g^k_i. The performance of Algorithm 1 (with $N = 6$ and $\alpha = 0.95$) for the various noise levels and all genes is conveyed by the scores on the performance indices R, S, A and D (Table 1). These were computed

Table 1. Identification performance for the repressilator network

		$\sigma_e, \sigma_\epsilon$	0.01	0.03	0.05	0.07
Gene 1	Step 1	R	1	1	1	1
		S	0.92	0.92	0.92	0.91
	Step 2	A	0.90	0.92	0.91	0.89
		D	1	1	1	1
Gene 2	Step 1	R	1	1	1	1
		S	0.92	0.92	0.92	0.91
	Step 2	A	0.93	0.92	0.89	0.89
		D	1	1	1	1
Gene 3	Step 1	R	1	1	1	1
		S	0.92	0.92	0.92	0.92
	Step 2	A	0.93	0.93	0.93	0.92
		D	1	1	1	1
Gene 4	Step 1	R	1	1	1	1
		S	0.94	0.92	0.87	0.65
	Step 2	A	0.94	0.94	0.93	0.89
		D	1	1	1.02	1.44
Gene 5	Step 1	R	1	1	1	1
		S	0.94	0.74	0.53	0.48
	Step 2	A	0.95	0.94	0.91	0.83
		D	1	1	1.79	4
Gene 6	Step 1	R	1	1	1	1
		S	0.79	0.65	0.57	0.43
	Step 2	A	0.89	0.92	0.85	0.42
		D	1	1.02	2.76	2.74

as described in Section 2.3.4 on the basis of $M=100$ identification runs with the same system evolution, but with different random outcomes of the noise. Each run (MATLAB V.7 R.14) took on an average roughly 5 min on a Windows XP workstation with Pentium 3.20 GHz processor and 2.00 GB RAM. Computational time ranged from ∼2 s for the identification of g_3 to ∼4 min for the identification of g_6. Step 1 always performs very reliably, i.e. index R is constantly equal to 1. This is expected since the choice $N=6$ makes Step 1 conservative, i.e. the probability of declaring the true sign pattern inconsistent is negligible. The selectivity S generally decreases with the increase of the noise level. Even for high noise level, however, Step 1 is able to save the exploration of ∼50% of the sign patterns from the iterative identification procedure in Step 2; note that the total number of patterns that would be explored in absence of Step 1 is 232 for the most complicated case of Gene 6. The accuracy A of Step 2 is very high. The true model structure is included in the pool of identified models in >80% of the cases except for Gene 6. In this case, which corresponds to the most complicated synthesis rate function, the accuracy drops drastically with the increase of noise. On average, pools of less than four models (index D) are returned and need to be discriminated on the basis of biological knowledge or dedicated experiments. Note that the data fit of all models produced by Algorithm 1 is satisfactory (according to the acceptance test in Step 2.VI of Algorithm 1), even if the correct model is not found. Scores D in Table 1 reveal that when the true structure is among the models found it is also often the only model found. Intuitively, this means that frequently the best fit model (usually associated to the correct structure) is the only one that passes the acceptance test among models of the same complexity. Finally, for correctly identified model structures, parameter estimates turned out to be quite accurate in all numerical experiments (results not shown).

3.2 Performance assessment on the IRMA network

A graphical representation of the network of interactions IRMA, comprising five genes, is depicted in Figure 1a. Time series of gene product concentrations and corresponding standard errors were obtained in Cantone *et al.* (2009) by averaging different experimental replicates (Supplementary Material) under two growth-medium conditions termed *switch-on* and *switch-off*. In particular, 15 and 20 data points collected every 20 and 10 min are available for the switch-on and switch-off experiments, respectively.

Data in both conditions were used to assess the performance of different state-of-the-art techniques, ranging from ordinary differential equation (ODE) models to Bayesian and information theoretic approaches. In order to quantify the performance of reverse engineering algorithms, Cantone *et al.* (2009) considered the *unsigned* directed graph produced by each method and compared it with the unsigned version of the graph in Figure 1 computing the positive predictive value (PPV) = TP/(TP + FP) (TP, true positive arcs; FP, false positive arcs) and the Sensitivity (Se) = TP/(TP + FN) (FN, false negative arcs). The use of unsigned graphs does not make any distinction between activatory and inhibitory interactions. According to Cantone *et al.* (2009), the ODE-based TSNI algorithm (Bansal and di Bernardo, 2007) was able to achieve the best performance in the context of reverse engineering from time series data, and hence it will be used for comparison in our study. The networks produced by TSNI are reported in Figure 1b along with their performance measures.

Given the availability of time series concentration data only and due to the lack of *in vivo* measurements of synthesis rates, we generated the latter data using the mathematical model proposed by Cantone *et al.* (2009, Supplementary Results) where just three out of five kinetic models have a unate structure. We then applied Algorithm 1 with $N=2$ and $\alpha=0.95$ to the combination of the *in vivo* concentration and *in silico* synthesis data. The latter were corrupted by an artificial measurement noise according to (7) with $(\sigma_\epsilon g_i^k)^2$. Three values of σ_ϵ (0.07, 0.1 and 0.3) were considered. Roughly, they correspond to noise contributions within 20%, 30% and 90% of the data values. The value $N=2$ was chosen to make the sign pattern selection effective given the highly noisy data. For each value of σ_ϵ, 100 noisy dataset were produced and, for each dataset, a single reconstructed network was obtained by selecting the best fit model among the pool returned by Algorithm 1. The average performance of the method was evaluated by computing mean and standard error of the PPV and Se values for the 100 reconstructed networks, as shown in Table 2. In order to provide a visual inspection of the results, Figure 1c shows as representative networks the ones inferred more frequently for switch-on and switch-off time series.

Results in Table 2 show that Algorithm 1 succeeds in reconstructing correct interactions for the two lower noise levels, outperforming TSNI also. There is a decay of performance for increasing values of σ_ϵ, particularly relevant for the highest noise level. The good performance of the proposed technique is confirmed by the representative networks in Figure 1c, in particular with respect to the few false positive interactions. Interestingly, our method is also accurate in inferring the sign (activation/inhibition) of the

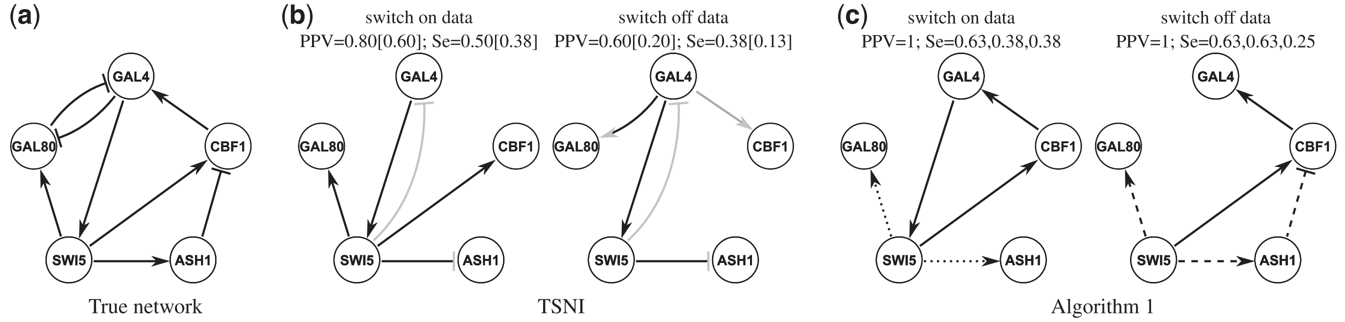

Fig. 1. (**a**) True network of interactions in IRMA. Results obtained by (**b**) the TSNI algorithm (Cantone *et al.*, 2009) and by (**c**) Algorithm 1. Grey arcs (respectively, grey-end markers) denote incorrect direction (respectively, sign) of the inferred interactions. Values of PPV and Se for the signed directed graph, when different from the unsigned case, appear in square brackets. The three values of Se in (c) refer to increasing noise levels, while dashed and dotted arcs denote interactions inferred only for $\sigma_\epsilon < 0.3$ and $\sigma_\epsilon < 0.1$, respectively.

Table 2. Average performance (standard errors in parentheses) on the IRMA datasets for different noise levels

	Switch-on data		Switch-off data	
σ_ϵ	PPV	Se	PPV	Se
0.07	0.98 (0.07) [0.98 (0.07)]	0.53 (0.08) [0.53 (0.08)]	0.91 (0.12) [0.88 (0.13)]	0.58 (0.07) [0.56 (0.08)]
0.1	0.95 (0.10) [0.94 (0.11)]	0.46 (0.08) [0.46 (0.08)]	0.85 (0.14) [0.80 (0.14)]	0.51 (0.09) [0.48 (0.09)]
0.3	0.67 (0.23) [0.64 (0.24)]	0.29 (0.10) [0.27 (0.10)]	0.58 (0.25) [0.52 (0.25)]	0.25 (0.11) [0.22 (0.11)]

Indices PPV and Se are reported for both the signed (in square brackets) and unsigned (without square brackets) directed graph.

interactions. Indeed, PPV and Se values computed with respect the *signed* graph in Figure 1a (i.e. an arc is false positive if it has either the wrong direction or the wrong sign) and shown in Table 2 are very similar to their unsigned counterparts. Moreover, all arcs in Figure 1c have the correct sign. Conversely, there is a significant performance decay for the TSNI algorithm, especially for the switch-off data where only one out of the five reconstructed interactions has correct direction and sign. This analysis reveals that the proper use of concentration and synthesis rate data can significantly improve results obtained by methods based on concentration data only, such as the algorithms considered in Cantone *et al.* (2009).

4 CONCLUDING REMARKS

We presented a kinetic modelling framework for genetic networks based on the unate structure of the regulation functions typically encountered in Boolean network modelling. We exploited monotonicity properties of the models in this class to devise a model identification procedure. Assessment on an artificial repressilator system and on the benchmark network IRMA revealed that the method performs well and outperforms state-of-the-art reconstruction methods provided product synthesis rates and concentration time series are both available. The fact that our method, compared with most reverse engineering algorithms, provides a pool of accepted models opens new perspectives in the analysis of the results. As an example one can assign confidence measures on the interactions according to their frequency of appearance in the pool of accepted models. For the case where only concentration time series are available, we are currently working on the extension of the method based on non-parametric or semiparametric estimation for the missing data. First results based on bootstrapping suggest that the approach is still feasible in this case.

ACKNOWLEDGEMENTS

The work of E. Cinquemani and J. Lygeros was supported in part by the SystemsX.ch research consortium under the project YeastX.

Conflict of Interest: none declared.

REFERENCES

Akutsu,T. *et al.* (2000) Algorithms for identifying Boolean networks and related biological networks based on matrix multiplication and fingerprint function. *J. Comput. Biol.*, **7**, 331–344.

Aracena,J. (2008) Maximum number of fixed points in regulatory Boolean networks. *Bull. Math. Biol.*, **70**, 1398–1409.

Bansal,M. and di Bernardo,D. (2007) Inference of gene networks from temporal gene expression profiles. *IET Syst. Biol.*, **1**, 306–312.

Bansal,M. *et al.* (2007) How to infer gene networks from expression profiles. *Mol. Syst. Biol.*, **3**, 78.

Brown,D. and Lostroh,C.P. (2008) Inferring gene expression dynamics from reporter protein levels. *Biotechnol. J.*, **3**, 1437–1448.

Cantone,I. *et al.* (2009) A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. *Cell*, **137**, 172–181.

Cinquemani,E. *et al.* (2008) Stochastic dynamics of genetic networks: modelling and parameter identification. *Bioinformatics*, **24**, 2748–2754.

Cinquemani,E. *et al.* (2009) *Local Identification of Piecewise Deterministic Models of Genetic Networks*, Vol. N.5469 of *LNCS Series*. Springer, Berlin/Heidelberg, Germany, pp. 105–119.

de Jong,H. (2002) Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol.*, **9**, 69–105.

Drulhe,S. *et al.* (2008) Reconstruction of switching thresholds in piecewise-affine models of genetic regulatory networks. *IEEE Trans. Automat. Control*, **53**, 153–165.

Elowitz,M. and Leibler,S. (2000) A synthetic oscillatory network of transcriptional regulators. *Nature*, **403**, 335–338.

Gardner,T. and Faith,J. (2005) Reverse-engineering transcription control networks. *Phys. Life Rev.*, **2**, 65–88.

Gardner,T. *et al.* (2003) Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, **301**, 102–105.

Glass,L. and Kauffman,S. (1973) The logical analysis of continuous, nonlinear biochemical control networks. *J. Theor. Biol.*, **39**, 103–129.

Grefenstette,J. *et al.* (2006) An analysis of the class of gene regulatory functions implied by a biochemical model. *BioSystems*, **84**, 81–90.

Jaeger,J. *et al.* (2004) Dynamic control of positional information in the early *Drosophila* embryo. *Nature*, **430**, 368–371.

Jarrah,A.S. *et al.* (2007) Nested canalyzing, unate cascade, and polynomial functions. *Physica D.*, **233**, 167–174.

Kauffman,S. (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, **22**, 437–467.

Kauffman,S. *et al.* (2004) Genetic networks with canalyzing boolean rules are always stable. *Proc. Natl Acad. Sci. USA*, **101**, 17102–17107.

Keller,A.D. (1995) Model genetic circuits encoding autoregulatory transcription factors. *J. Theor. Biol.*, **172**, 169–185.

Kreutz,C. *et al.* (2007) An error model for protein quantification. *Bioinformatics*, **23**, 2747–2753.

Laubenbacher,R. and Stigler,B. (2004) A computational algebra approach to the reverse engineering of gene regulatory networks. *J. Theor. Biol.*, **229**, 523–537.

Markowetz,F. and Spang,R. (2007) Inferring cellular networks: a review. *BMC Bioinformatics*, **28** (Suppl. 6), S5.

Nachman,I. *et al.* (2004) Inferring quantitative models of regulatory networks from expression data. *Bioinformatics*, **20** (Suppl. 1), i248–i256.

Nagarajan,R. *et al.* (2004) Modeling genetic networks from clonal analysis. *J. Theor. Biol.*, **230**, 359–373.

Nikolajewa,S. *et al.* (2007) Boolean networks with biologically relevant rules show ordered behavior. *BioSystems*, **90**, 40–47.

Plahte,E. *et al.* (1998) A methodological basis for description and analysis of systems with complex switch-like interactions. *J. Math. Biol.*, **36**, 321–348.

Porreca,R. *et al.* (2008) Structural identification of piecewise-linear models of genetic regulatory networks. *J. Comput. Biol.*, **15**, 1365–1380.

Raeymaekers,L. (2002) Dynamics of boolean networks controlled by biologically meaningful functions. *J. Theor. Biol.*, **218**, 331–342.

Ronen,M. *et al.* (2002) Assigning numbers to the arrows: Parameterizing a gene regulation network by using accurate expression kinetics. *Proc. Natl Acad. Sci. USA*, **99**, 10555–10560.

Segal,E. *et al.* (2003) Module networks: Identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genet.*, **34**, 166–176.

Szallasi,Z. and Liang,S. (1998) Modeling the normal and neoplastic cell cycle with "realistic Boolean genetic networks": their application for understanding carcinogenesis and assessing therapeutic strategies. *Proc. Pac. Symp. Biocomput.*, **3**, 66–76.

Yang,H. *et al.* (2007) An analytical rate expression for the kinetics of gene transcription mediated by dimeric transcription factors. *J. Biochem.*, **142**, 135–144.

Zavlanos,M. *et al.* (2008) Identification of stable genetic networks using convex programming. In *Proceedings of the American Control Conference*, IEEE Inc., Piscataway, Washington, USA, pp. 2755–2760.