BioSystems 149 (2016) 139–153

Contents lists available at ScienceDirect

BioSystems

journal homepage: www.elsevier.com/locate/biosystems

Boolean network identification from perturbation time series data combining dynamics abstraction and logic programming

M. Ostrowski[a,1], L. Paulevé[c,1], T. Schaub[a,b], A. Siegel[d], C. Guziolowski[e,*]

[a] University of Potsdam, Potsdam, Germany
[b] INRIA, Rennes, France
[c] CNRS, Université Paris-Sud LRI-UMR 8623, Orsay, France
[d] CNRS, Université de Rennes 1, IRISA-UMR 6074, Rennes, France
[e] École Centrale de Nantes, IRCCyN UMR CNRS 6597, Nantes, France

ARTICLE INFO

Article history:
Received 8 December 2015
Received in revised form 8 July 2016
Accepted 19 July 2016
Available online 30 July 2016

Keywords:
Model identification
Time series data
Multiplex phosphoproteomics data
Boolean networks
Answer Set Programming

ABSTRACT

Boolean networks (and more general logic models) are useful frameworks to study signal transduction across multiple pathways. Logic models can be learned from a prior knowledge network structure and multiplex phosphoproteomics data. However, most efficient and scalable training methods focus on the comparison of two time-points and assume that the system has reached an early steady state. In this paper, we generalize such a learning procedure to take into account the time series traces of phosphoproteomics data in order to discriminate Boolean networks according to their transient dynamics. To that end, we identify a necessary condition that must be satisfied by the dynamics of a Boolean network to be consistent with a discretized time series trace. Based on this condition, we use Answer Set Programming to compute an over-approximation of the set of Boolean networks which fit best with experimental data and provide the corresponding encodings. Combined with model-checking approaches, we end up with a global learning algorithm. Our approach is able to learn logic models with a true positive rate higher than 78% in two case studies of mammalian signaling networks; for a larger case study, our method provides optimal answers after 7 min of computation. We quantified the gain in our method predictions precision compared to learning approaches based on static data. Finally, as an application, our method proposes erroneous time-points in the time series data with respect to the optimal learned logic models.

1. Introduction

Generic prior knowledge about canonical cell signaling networks can be retrieved from database sources. This provides a first insight on how cells respond to their environment by triggering processes such as growth, survival, apoptosis (cell death), and migration. However, little is known about the exact chaining and composition of signaling events within these networks in specific cells and in response to specific experimental perturbations, as provided by the simulations of predictive mathematical models, e.g. a set of differential equations or a set of logic rules. When building predictive models, the parameters of a model (built according to generic prior knowledge) can be fitted to the data to obtain the most plausible model for a specific cell type, if enough experimental data is available. This is normally achieved by defining an objective fitness function to be optimized. In this context, post-translational modifications, notably protein phosphorylation, play a key role in signaling. They are very useful for the training of model parameters through the use of multiplex phosphorylation assays, a recent form of high-throughput data providing information about protein-activity modifications in a specific cell type upon various experimental perturbations (clamping) (Alexopoulos et al., 2010).

Boolean logical networks (Kauffman, 1969) provide a simple yet powerful qualitative framework which has become very popular during the last decade to model signaling or regulatory networks (Wang et al., 2012). In contrast to quantitative methods which permit fine-grained kinetic analysis, qualitative approaches allow for addressing large-scale biological networks. In this context, the manual identification of logic rules underlying the system has been addressed under different hypotheses and methods (Berestovsky and Nakhleh, 2013). In particular, scalable methods restrain themselves to learning models from two time points (start; end) and assume the system has reached an early steady-state when the measurements are performed. As shown in MacNamara et al.

[*] Corresponding author.
E-mail address: Carito.Guziolowski@irccyn.ec-nantes.fr (C. Guziolowski).
[1] Co-first authors.

http://dx.doi.org/10.1016/j.biosystems.2016.07.009
0303-2647/

(2012), this assumption prevents capturing important characteristics of signaling networks such as feedback cycles.

Data-driven methods for learning causal graphs representing molecular networks have been widely proposed (Bansal et al., 2007; Smet and Marchal, 2010; Maetschke et al., 2014). We here focus on methods learning causal graphs from time series data. Some of such methods use ODE's kinetic modeling to build dynamic predictive models from time series gene expression data (Busch et al., 2008; Porreca et al., 2010). Recently, a crowdsourcing challenge was proposed to learn causal graphs for breast cancer cell lines using multiplex phosphoproteomic time series data. The results in Hill et al. (2016) show that methods based on machine learning techniques, using only data and no prior knowledge network, obtained a significant score. This study reported as well that methods using prior-knowledge outperformed methods solely based on data. Having said that, the task of evaluation of such methods is very delicate since an exhaustive experimental verification of a small-scale causal graph is not feasible. Furthermore, other types of evaluation such as verifying model predictions, do not necessarily require a causal graph model structure. The approach we propose here belongs to the category of methods that use time series phosphoproteomics data and a prior-knowledge graph to infer logical causal models that can be simulated *a-posteriori* using synchronous or asynchronous updates.

Our method infers Boolean networks (BNs) from time series datasets and it scales to the size of currently studied BNs. Given multiplex time series data from the measurement of a partial set of biological entities under different experimental perturbations, we want to identify all the BNs that have a structure compatible with a given prior knowledge interaction graph and that can reproduce all the (experimentally) observed time series. Time series data are assumed incomplete, i.e., only a subset of network components are observed, with measurements made at discrete time points and with normalized continuous values. It is possible that no BN, constrained by the prior interaction graph, reproduces all the input time series. In such a case, we introduce a fitness function to measure the distance between a trace of a BN simulation and a measured time series. Therefore, we aim to infer the BNs whose dynamics contains traces with the best fitness to all measurements.

Our approach relies on the combination of several techniques. First, we introduce a necessary condition for discretized time series data to be the trace of a BN. This provides an over-approximation of the successive reachability properties, allowing to reject BNs which cannot reproduce the time series without a costly exhaustive analysis of the dynamics. Then, we use Answer Set Programming (ASP) to enumerate BNs which approximate the best experimental data while satisfying the necessary condition on the dynamics. As a result, we obtain a set of BNs associated with traces which both satisfy the necessary condition and optimally fit with experimental data. Because of the over-approximation of reachability, a part of the returned BNs cannot reproduce the associated Boolean traces. Such false positives can be detected *a posteriori* using model-checking on the returned results.

This paper extends the results presented in Ostrowski et al. (2015) in several ways: a complete and detailed characterization of the method, illustrated step by step with a toy example; a detailed description of the ASP implementation of the inference and optimization, together with justifications for the computation of the fitness of predictions and experimental observations; and with a general benchmark, composed of three case studies, which increases the diversity of networks against which we evaluate our method.

In order to evaluate our method we used synthetic data generated from BNs of three mammalian signaling networks induced by the: (i) Epidermal Growth Factor (EGF) and Tumor Necrosis Factor alpha (TNFα), (ii) T-cell Receptor (TCR), and (iii) Epidermal Growth Factor Receptor (ERBB). These networks have between 13–40 nodes and 16–50 edges and contain multiple cycles. Our prototype implementation was able to identify efficiently all BNs satisfying the necessary condition with a rate of true positives over 78% for networks of less than 25 edges. We measured the impact of incomplete networks on the precision of learned BNs; and estimated the added-value of models identified with our method on the full time series with respect to models learned from two time points, considered as a steady state. Finally we present an application of this method to detect erroneous time-points in the time series data with respect to the learned BNs.

2. Boolean network identification

2.1. Admissible Boolean networks and multiplex time series data

Boolean networks (BNs). A BN with n components $\{1, \ldots, n\}$ consists of a tuple of n Boolean functions $F = (f_1, \ldots, f_n)$ where each function $f_i : \mathbb{B}^n \to \mathbb{B}$, $\mathbb{B} \overset{\Delta}{=} \{0, 1\}$, $i \in \{1, \ldots, n\}$, associates with each global state $x \in \mathbb{B}^n$ of the network the next value of the ith component. The value of the i-th component in x is denoted x_i.

As a toy example that is used all along the present article, let us consider the BN depicted in Fig. 1b with four components *nI*, *nJ*, *nA*, and *nB* in $\mathbb{B}$ associated to the following Boolean functions:

$$F : \begin{cases} f_{nI}(nI, nJ, nA, nB) = 0 \\ f_{nJ}(nI, nJ, nA, nB) = \neg nI \\ f_{nA}(nI, nJ, nA, nB) = nI \wedge \neg nB \\ f_{nB}(nI, nJ, nA, nB) = nJ \vee nA \end{cases}$$

Transition relation and associated semantics. The transitions between global states of the network are specified with a transition relation $\to \quad \subseteq \mathbb{B}^n \times \mathbb{B}^n$. The transitive closure of $\to$ is denoted by $\to^*$. Given $x, x\prime \in \mathbb{B}^n$, $x \to^* x'$ if and only if, either $x = x'$ or $x \to \cdots \to x'$.

Several definitions of the transition relation $\to$ can be used depending on the update schedule of the components (Aracena et al., 2009), ranging from so-called parallel (or synchronous) updates where each transition updates the value of all the components, to the asynchronous update where each transition updates the value of only one component chosen non-deterministically.

As the over-approximation results presented in this article are independent of the update schedule, we use the general definition, where any number of components can be updated during a transition: for any $x, x\prime \in \mathbb{B}^n$,

$$x \to x\prime \overset{\Delta}{\Leftrightarrow} (x \neq x\prime) \wedge (\forall i \in \{1, \ldots, n\}, x\prime_i \neq x_i \Rightarrow x\prime_i = f_i(x)). \tag{1}$$

For example, in our toy example, we have that $F(1, 1, 0, 0) = (1, 0, 1, 1)$. This means that three variables may change their value from the state $(1, 1, 0, 0)$. Therefore, we have that $(1, 1, 0, 0) \to (1, 0, 1, 1)$ is a synchronous update scheme, whereas $(1, 1, 0, 0) \to (1, 0, 0, 0)$, $(1, 1, 0, 0) \to (1, 1, 1, 0)$ and $(1, 1, 0, 0) \to (1, 0, 0, 1)$ are valid in an asynchronous update. In our framework, we consider as valid the eight transitions $(1, 1, 0, 0) \to (1, b, c, d)$ where $b, c, d \in \mathbb{B}$.

Prior knowledge network and admissible BNs. An *interaction graph* between n components is a digraph between nodes $\{1, \ldots, n\}$ where each edge is signed, i.e., either positive or negative. The interaction graph of a BN F, noted $\mathsf{IG}(F)$, has a positive (resp. negative) edge from node j to node i if and only if there exists $x, x\prime \in \mathbb{B}^n$ which are identical except on the j-th coordinate where $x_j = 0$ and $x'_j = 1$ and such that $f_i(x) < f_i(x')$ (resp. $f_i(x) > f_i(x')$). Notice that according to this definition, an interaction may contain multiple edges with different signs. The interaction graph of our toy example is depicted in Fig. 1a. Notice that in this graph *nI* is a specific node since it has no predecessor. We call it an input node.