



Target Control in Logical Models Using the Domain of Influence of Nodes

Gang Yang^{1*}, Jorge Gómez Tejeda Zañudo^{1,2,3*} and Réka Albert^{1,4*}

¹ Department of Physics, Pennsylvania State University, University Park, PA, United States, ² Department of Medical Oncology, Dana-Farber Cancer Institute, Boston, MA, United States, ³ Eli and Edythe L. Broad Institute of MIT and Harvard, Cambridge, MA, United States, ⁴ Department of Biology, Pennsylvania State University, University Park, PA, United States

OPEN ACCESS

Edited by:

Matteo Barberis,
University of Amsterdam, Netherlands

Reviewed by:

Osbaldo Resendis-Antonio,
Universidad Nacional Autónoma de
México, Mexico

María Suarez-Diez,
Wageningen University & Research,
Netherlands

Denis Thieffry,
École Normale Supérieure, France

*Correspondence:

Gang Yang
yanggangthu@gmail.com
Jorge Gómez Tejeda Zañudo
jgtz@phys.psu.edu
Réka Albert
rza1@psu.edu

Specialty section:

This article was submitted to
Systems Biology,
a section of the journal
Frontiers in Physiology

Received: 04 January 2018

Accepted: 13 April 2018

Published: 08 May 2018

Citation:

Yang G, Gómez Tejeda Zañudo J and
Albert R (2018) Target Control in
Logical Models Using the Domain of
Influence of Nodes.
Front. Physiol. 9:454.
doi: 10.3389/fphys.2018.00454

Dynamical models of biomolecular networks are successfully used to understand the mechanisms underlying complex diseases and to design therapeutic strategies. Network control and its special case of target control, is a promising avenue toward developing disease therapies. In target control it is assumed that a small subset of nodes is most relevant to the system's state and the goal is to drive the target nodes into their desired states. An example of target control would be driving a cell to commit to apoptosis (programmed cell death). From the experimental perspective, gene knockout, pharmacological inhibition of proteins, and providing sustained external signals are among practical intervention techniques. We identify methodologies to use the stabilizing effect of sustained interventions for target control in Boolean network models of biomolecular networks. Specifically, we define the domain of influence (DOI) of a node (in a certain state) to be the nodes (and their corresponding states) that will be ultimately stabilized by the sustained state of this node regardless of the initial state of the system. We also define the related concept of the logical domain of influence (LDOI) of a node, and develop an algorithm for its identification using an auxiliary network that incorporates the regulatory logic. This way a solution to the target control problem is a set of nodes whose DOI can cover the desired target node states. We perform greedy randomized adaptive search in node state space to find such solutions. We apply our strategy to *in silico* biological network models of real systems to demonstrate its effectiveness.

Keywords: target control, Boolean network, biological network, domain of influence, logical modeling, network dynamics

1. INTRODUCTION

In cellular systems various molecular species, such as DNA, RNA, proteins and small molecules, interact in diverse ways. The totality of these interactions gives rise to cellular functions. The relationship between molecular interacting systems and cellular functions is studied in the new emerging field of systems biology (Alon, 2006; Palsson, 2006). A promising systems biology methodology is to represent the molecular interacting system as a network, construct a dynamic model of the information propagation on this network, and identify the cellular functions with long-term behaviors of the dynamic model (Palsson, 2006; Newman, 2010; Wang et al., 2012; Barabási and Pósfai, 2016). Various types of dynamical models of biological networks have been built to integrate related experimental results and to reveal the underlying mechanisms of complex

diseases such as cancers, and predict beneficial interventions. Quantitative, mechanistic models, generally using systems of ordinary or partial differential equations, can be highly accurate and provide quantitative information (e.g., response time to a signal, or fold-changes of protein concentrations) (Tyson et al., 2003, 2011; Alon, 2006; Iyengar et al., 2012). These models' widespread use is limited by the scarcity of high-quality quantitative data, such as kinetic and temporal information about individual nodes in the network. Logical models using discrete variables, such as Boolean network models, have the advantage of being scalable and not requiring detailed knowledge of kinetic parameters (Morris et al., 2010; Wynn et al., 2012; Saadatpour and Albert, 2013; Laubenbacher et al., 2014; Abou-Jaoudé et al., 2016; Bloomingdale et al., 2018; Zañudo et al., 2018). An abundance of recent literature has shown that logical models can capture the emergent behaviors of real biological systems, they can generate predictions that are validated by follow-up experiments and they can predict successful intervention strategies (Li et al., 2004; Espinosa-Soto et al., 2004; Mendoza, 2006; Saez-Rodriguez et al., 2007; Naldi et al., 2010; Miskov-Zivanov et al., 2013; Steinway et al., 2015; Albert et al., 2017; Gómez Tejeda Zañudo et al., 2017). For example, logical models of signaling networks that underlie hallmarks of cancer identified the key mechanisms that yield cancer phenotypes and predicted therapeutic interventions that disrupt these phenotypes; many of these predictions were validated experimentally (Grieco et al., 2013; Cohen et al., 2015; Méndez-López et al., 2017; Khan et al., 2017; Kim et al., 2017). Discrete and quantitative models are often consistent in capturing the response repertoire of biological networks (e.g., their potential bistability or response to perturbations) (Kraeutler et al., 2010; Steinway et al., 2016).

Analysis of a logical model entails the determination of the attractors (long-term behaviors) of the system and of the initial states that converge into each attractor (the basins of attraction). Among other uses, this information is used to identify therapeutic interventions as interventions that make a disease attractor unreachable or unstable (Samaga et al., 2010; Abou-Jaoudé et al., 2015; Kim et al., 2017). Attractor identification can be accomplished by simulations of the system's trajectories, determination of all allowed state transitions, or by formal methods such as model checking (Klärner and Siebert, 2015; Abou-Jaoudé et al., 2016), process hitting (Paulevé et al., 2012), or Groebner bases (Laubenbacher et al., 2014). The state space of logical models is finite, but its size scales exponentially with the number of nodes, and thus its full mapping is impossible for systems with many elements. Methods that determine the attractor repertoire of logical models without state space exploration provide a desirable complement to dynamical methods. For example, it was shown that the presence or absence of positive and negative feedback loops in the interaction network puts bounds on the type and number of attractors; e.g., a necessary condition of multistability is the existence of a positive feedback loop (Thomas and D'Ari, 1990; Paulevé and Richard, 2012).

Network control has recently become a popular research topic as it reflects our interest to not only understand an interacting system, but also intervene in it and modify its outcomes (Motter,

2015; Liu and Barabási, 2016). Network control is a broad subject; different underlying models, different control goals and different possible interventions can be considered (Liu and Barabási, 2016). Various control strategies have been designed for both continuous dynamical systems (Liu et al., 2011; Cornelius et al., 2013; Mochizuki et al., 2013; Wells et al., 2015; Wang et al., 2016; Zañudo et al., 2017) and discrete ones (Murrugarra and Dimitrova, 2015; Zañudo and Albert, 2015; Murrugarra et al., 2016; Yang et al., 2016). Of particular interest are the methods that do not require knowledge of the detailed dynamics and parameters of the system, but instead are largely based on the structure of the interaction network and generic assumptions about the functional form of the dependences among variables. In electric circuits modeled by a system of linear ordinary differential equations, it is possible to use graph theoretical methods to identify the set of nodes whose external control can drive the system to any state from any initial condition (Lin, 1974; Liu et al., 2011). For systems with non-linear dynamics, attractor control, that is, to drive the system to one of its natural attractors from any initial condition, has been achieved in several modeling frameworks. Among these, two methods are based on the control of feedback loops: feedback vertex control for ordinary differential equation models (Mochizuki et al., 2013) and stable motif control for logic (Boolean) models (Zañudo and Albert, 2015). However, in biological systems it is not necessary and often not practical to control every component of the system. A more realistic problem is target control, where we assume that the state of the system is characterized by a subset of components and the control goal is to drive these components into desired states. The target control problem has been considered in systems with linear dynamics by Gao et al. (2014), who identified sets of nodes which, if put under suitable (potentially time-varying) external control, drive the target nodes into the desired state.

Despite recent progress in molecular biology, quantitatively manipulating the level of a chemical species is still a challenging problem for experimentalists. Thus any control strategy involving applying time-dependent, variable signals to a target is hard to implement in real systems. However, gene knockout, pharmacological inhibition of proteins and providing sustained external signals have been robustly implemented and demonstrated to be effective intervention strategies (Hopkins and Groom, 2002; Nicholl, 2008; Shalem et al., 2014). Thus we choose our intervention options to be maintaining a sustained state (either absence or abundant activity) in order to make the solution more practical. The effect of such interventions to achieve target control in Boolean network models was previously considered by Klamt et al. (2006) and Samaga et al. (2010). Klamt et al. used the interaction network and regulatory logic to identify the effect of interventions, and determined minimal intervention sets by systematic consideration of all single interventions and combinations of interventions. Samaga et al. made the search for interventions more efficient by using filtering strategies based on the interaction network (e.g., if a candidate intervention source has only negative paths to a target node, then an activating intervention of the source is not useful for activation of the target node) and by grouping equivalent interventions (e.g., if activating a node is sufficient for activating a direct neighbor, then these

interventions are equivalent and only one of them needs to be considered).

Here we propose an alternative and complementary intervention prediction method that uses heuristics based on the system-wide influence of the intervention due both to the connectivity and regulatory logic of the modeled system. Specifically, we base these heuristics on each node's domain of influence (DOI), which identifies which other nodes will adopt a fixed state following an intervention that maintains a sustained state of this node, regardless of the system's initial state. While in general determining the DOI of a node requires exploration of the state space, here we introduce the related concept of logical domain of influence (LDOI) of a node, which can be determined based on the interaction network and the regulatory logic. Specifically, the LDOI is defined on the so-called expanded network introduced in (Albert and Othmer, 2003; Wang and Albert, 2011), which is similar in spirit to a logical interaction hypergraph (Klamt et al., 2006). We use the size and internal consistency of the logic domain of influence (LDOI) to inform a greedy randomized adaptive search to identify the sets of nodes whose DOI can cover the desired target node state (combination).

In the following, we give background information on the Boolean modeling framework and relevant previously-developed concepts such as the expanded network and stable motifs. Then we define the DOI and LDOI of a node or multiple nodes and analyze their properties, such as their internal consistency (or lack thereof) and relationship to dynamic attractors. We then define our target control problem and describe our DOI-based target control strategy using greedy randomized adaptive search in node state space. We finally illustrate the effectiveness of our target control strategy in random ensembles and four *in silico* biological network models.

2. MATERIALS AND METHODS

2.1. Background on Boolean Network Models of Biological Systems

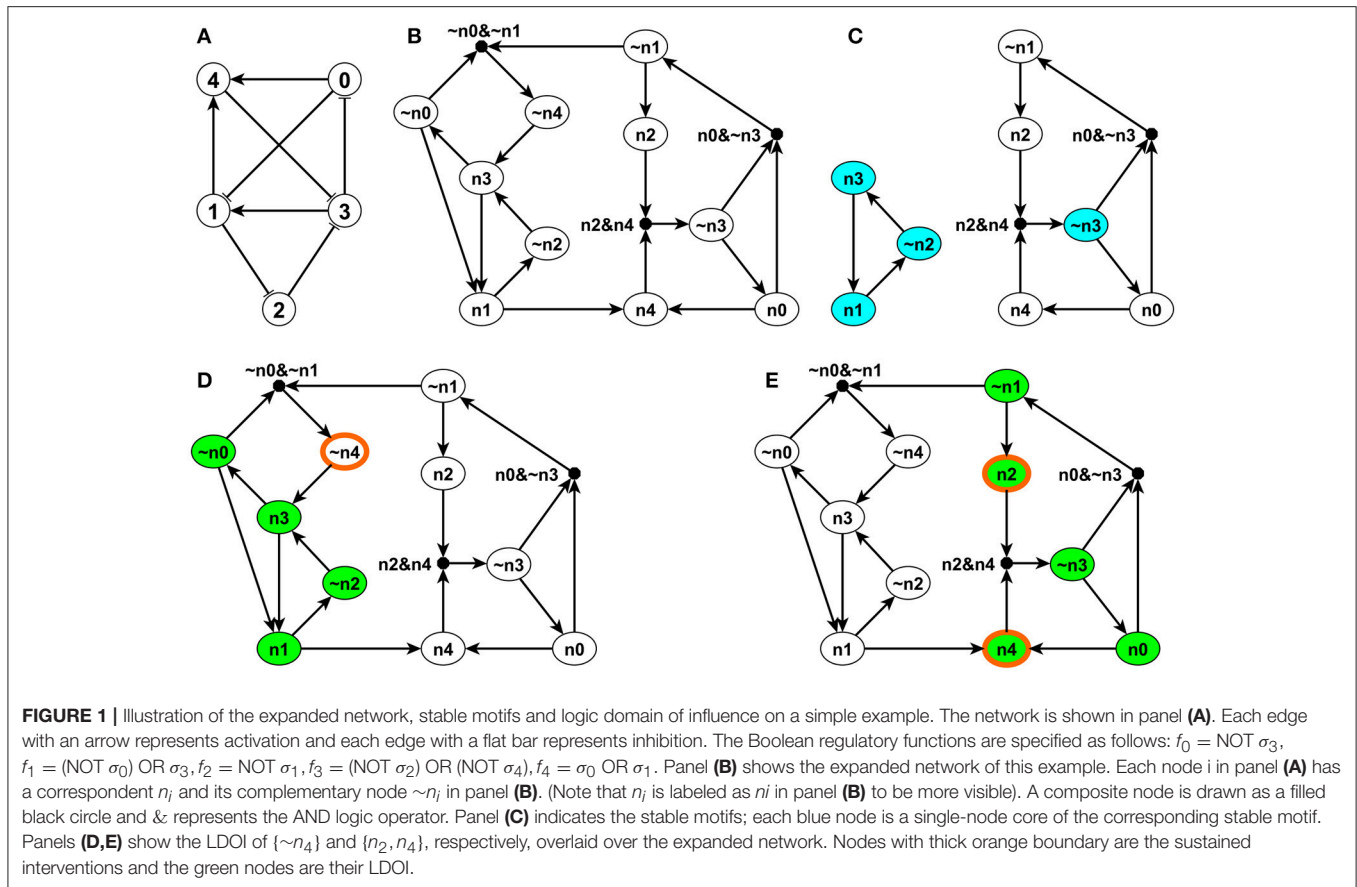
A dynamical model of a biological system starts with the construction of a network (graph) consisting of nodes (also called vertices) that represent the system's elements and edges that specify the pairwise relationships between nodes. In biological networks at the molecular level, nodes are molecular species such as small molecules, RNA, protein, and edges indicate interactions and regulatory relationships. In discrete dynamic (also called logical) models, each node i is characterized by a discrete state variable σ_i , and the vector $(\sigma_1, \dots, \sigma_n)$ represents the state of the system (Morris et al., 2010; Wynn et al., 2012; Saadatpour and Albert, 2013; Laubenbacher et al., 2014; Abou-Jaoudé et al., 2016; Bloomingdale et al., 2018; Zaňudo et al., 2018). The state of the system can be followed in continuous time or at discrete time intervals. In discrete time models, the activity of each node σ_i is described by a regulatory function $\sigma_i(t + \tau_i) = f_i(\sigma_{i_1}(t), \dots, \sigma_{i_k}(t))$, where i_1, \dots, i_k are the regulating nodes of i and τ_i is a discrete time delay. The regulatory functions f cannot be constant functions (i.e., cannot yield the same output

regardless of the state of the regulators). In models describing signal transduction networks the external signals are represented with source nodes whose regulatory functions depend only on their own state, usually sustaining this state: $\sigma_i(t + \tau_i) = \sigma_i(t)$.

Here we focus on discrete time Boolean network models, where node states are binary, 1(ON) or 0(OFF), and the regulatory function is specified by a truth table or using the Boolean operators AND, OR, NOT (Kauffman, 1969; Glass and Kauffman, 1973). This is motivated by the fact that biological species are frequently observed to demonstrate switch-like behaviors and have highly nonlinear regulations; thus the node state 1 means the molecular species is above a threshold concentration or activity and thus it is able to regulate its targets, and the node state 0 means it is below a threshold concentration or activity and is thus ineffective (Bornholdt, 2008; Wang et al., 2012). Depending on the updating scheme, the time trajectory of the system is simulated deterministically or stochastically. A simple deterministic updating scheme is synchronous updating, where $\tau_i = 1$ for every node (Wang et al., 2012). In this scheme, the system will deterministically evolve from a specific initial state into an attractor, which can be a steady state (fixed point) or a limit cycle, which consists of several states that repeat regularly. Steady states can be interpreted as cell types; limit cycles may correspond to a cell cycle or circadian rhythms. In general asynchronous updating, a commonly used stochastic updating scheme, a random node is selected to be updated at each time step (Glass, 1975). This type of update is motivated by the fact that different biological processes have various timescales, and often the timescales of specific processes are not known (Papin et al., 2005). While limit cycles depend on the specific chosen updating regime, fixed points (steady states) do not depend on the updating scheme (Klemm and Bornholdt, 2005). Stochastic update may lead to attractors that involve irregular repetition of a set of states, called complex attractors.

2.2. The Expanded Network and Its Use in Identifying the Attractor Repertoire of a Boolean Network

The possible combinatorial effect of multiple incoming regulators of a node is important, however, it is not explicitly represented by a regular interaction network. This motivated researchers to develop a concept called the expanded network, which integrates the original network with the regulatory rules of each node (Albert and Othmer, 2003; Wang and Albert, 2011). We illustrate the expanded network with the example in **Figure 1**, which consists of five nodes, node 0, 1, 2, 3, and 4 with the regulatory functions $f_0 = \text{NOT } \sigma_3$, $f_1 = (\text{NOT } \sigma_0) \text{ OR } \sigma_3$, $f_2 = \text{NOT } \sigma_1$, $f_3 = (\text{NOT } \sigma_2) \text{ OR } (\text{NOT } \sigma_4)$, $f_4 = \sigma_0 \text{ OR } \sigma_1$. First, we denote each original node i by n_i in the expanded network, and we introduce a complementary node for each original node in the system to represent the negation (deactivation) of the original node, denoted by $\sim n_i$ (Wang and Albert, 2011). As the NOT function is a unary operator, all the NOT functions are replaced by the negated state of the respective node (i.e., its complementary node) in each Boolean regulatory function. Edges are introduced in the expanded network to



represent the thus-transformed regulatory functions so that every edge represents a positive regulatory relationship in the expanded network. For example, $f_0 = \text{NOT } \sigma_3$ implies the rule for the original node n_0 as $f_{n_0} = \text{NOT } n_3 = \sim n_3$, and thus a corresponding edge is drawn from $\sim n_3$ to n_0 in the expanded network. The Boolean regulatory function for the complementary (negated) node is the logical negation of the regulatory function of the original node. In this example, $f_{\sim n_0} = \text{NOT } (\text{NOT } n_3) = n_3$ and thus a corresponding edge is drawn from n_3 to $\sim n_0$ in the expanded network.

Second, to differentiate OR rules from AND rules when multiple edges point toward the same target node, we introduce a composite node for each set of edges that are linked by an AND function (Wang and Albert, 2011). In order to uniquely determine the edges of the expanded network, the regulatory functions need to be specified in disjunctive normal form, that is, a disjunction of conjunctive clauses (in other words, grouped AND clauses separated by OR clauses). For example, $(A \text{ AND } B) \text{ OR } (A \text{ AND } C)$ is in a disjunctive normal form, while its equivalent form $A \text{ AND } (B \text{ OR } C)$ is not. The desired disjunctive normal form can be formed by a disjunction of all conditions that give output 1 in the Boolean table and then simplified to the disjunction of prime implicants (Blake canonical form) by the Quine-McCluskey algorithm (McCluskey, 1956). Now we add a composite node for each AND clause

in the Boolean regulatory function, denoted by a filled black circle in **Figure 1B**. We add edges from the non-composite nodes of the expanded network that form this clause to this composite node. For example, the composite node $\sim n_0 \& \sim n_1$ in the left upper part of **Figure 1B** represents the expression $(\text{NOT } n_0) \text{ AND } (\text{NOT } n_1)$. The expanded network has edges from $\sim n_0$ to the composite node and from $\sim n_1$ to the composite node. This composite node expresses the regulatory function of the complementary node $\sim n_4$, namely, $f_{\sim n_4} = \text{NOT } f_{n_4} = \text{NOT } n_0 \text{ AND } \text{NOT } n_1 = \sim n_0 \text{ AND } \sim n_1$. To reflect this, the expanded network contains an edge from this composite node to $\sim n_4$. Now the benefit of introducing complementary and composite nodes is evident: one can read all the regulatory functions from the topology of the expanded network. The NOT rule is indicated by a complementary node, the AND rule is indicated by a composite node with multiple regulators, while all the other edges represent independent activation (parts of an OR function). Moreover, the expanded network also incorporates the negations of the regulatory functions. Thus, for each node i , the expanded network reflects the condition that needs to be satisfied in order for $\sigma_i = 1$ (in the incoming edges of the node n_i) and the condition that needs to be satisfied in order for $\sigma_i = 0$ (in the incoming edges of the node $\sim n_i$).

As the expanded network encapsulates the regulatory logic that determines the network dynamics, it can serve as a basis

for attractor analysis. One approach is through analyzing the stable motifs of the expanded network (Zañudo and Albert, 2013). A stable motif is defined as the smallest strongly connected component (SCC) satisfying the following two properties: (1) The SCC cannot contain both a node and its complementary node and (2) If the SCC contains a composite node, it must also contain all of its input nodes (Zañudo and Albert, 2013). The first requirement guarantees that the SCC does not contain any conflict in node states and the second requirement guarantees that all the conditional dependence is satisfied and the SCC is self-sufficient in maintaining each node state inside the stable motif. Thus the stable motif represents a group of nodes that can sustain their states irrespective of outside nodes' states. The corresponding node states implied by the stable motif can be directly read out: an original node represents the ON (1) state and a complementary node represents the OFF (0) state (Zañudo and Albert, 2013). For example, in the left part of **Figure 1C**, node $n_1, \sim n_2$, and n_3 form a stable motif, representing that node 1 and node 3 are ON and node 2 is OFF. There is a strong correspondence between stable motifs and the attractors of the system. Specifically, there is a one-to-one correspondence between a sequence of stable motifs and a fixed point or a partial fixed point (a part of a complex attractor). A partial fixed point is defined as a true subset of all the nodes whose respective state remains unchanged after being updated regardless of the states of the nodes excluded from this subset (Zañudo and Albert, 2013).

2.3. The Domain of Influence of a Sustained Node State

We define the DOI of an intervention that maintains a sustained node state as all the node states that will be stabilized (i.e., attain a stationary value) in the long term under the influence of this intervention for all initial conditions in any updating regime. Mathematically, $\mathcal{D}(\sigma_i = \tilde{\sigma}_i) = \{\sigma_j = \tilde{\sigma}_j : \sigma_j(t) = \tilde{\sigma}_j \text{ as } t \rightarrow \infty \text{ for any } (\sigma_1(t=0), \dots, \sigma_k(t=0)) \text{ when } \sigma_i(t) = \tilde{\sigma}_i \text{ for any } t > 0\}$, where $\sigma_i(t) = \tilde{\sigma}_i$ is the intervention, $\tilde{\sigma}_i = 0$ represents knockout or suppression and $\tilde{\sigma}_i = 1$ represents sustained activation, $\tilde{\sigma}_j$ represents a node state fixed by the intervention, and $(\sigma_1(t=0), \dots, \sigma_k(t=0))$ represents the initial condition of all the nodes of the system. We do not include the intervention node state $\sigma_i = \tilde{\sigma}_i$ in its own DOI, unless the node is sufficient to maintain the corresponding node state in the long term even in the absence of a sustained intervention. Notice that there is one-to-one correspondence between a node state $\sigma_i = \tilde{\sigma}_i$ and a non-composite node n^{ex} in the expanded network: $\sigma_i = 1$ corresponds to a normal node n_i in the expanded network and $\sigma_i = 0$ corresponds to a negation node $\sim n_i$. Thus we use the two notations interchangeably, that is, $\sigma_j = 1 \in \mathcal{D}(\sigma_i = 1)$ is equivalent to $n_j \in \mathcal{D}(n_i)$ and $\sigma_j = 0 \in \mathcal{D}(\sigma_i = 0)$ is equivalent to $\sim n_j \in \mathcal{D}(\sim n_i)$.

The DOI of a node is difficult to calculate because it entails determining the common part of all attractors of a dynamical system to identify the nodes whose states stabilize due to the considered intervention. As an alternative to this computationally hard problem, we define a related concept called the LDOI of an intervention that maintains a sustained

node state. The LDOI consists of all the node states that, for any initial condition, are stabilized by the first update of the corresponding node in an updating regime that preserves the level order (breadth first search order) of the expanded network. An updating regime preserves the level order if all the nodes in the n th level are updated at least once before updating any node in the $(n+1)$ th level (see details in Supplementary Material 2.1). We denote the LDOI of a node state σ_i as $\mathcal{LD}(\sigma_i = \tilde{\sigma}_i)$. We define the LDOI of an empty set to be an empty set, $\mathcal{LD}(\emptyset) = \emptyset$. This is consistent with the definition as an updating order preserving the level order starting from a null set can start from any node, and a node will not achieve a stationary state upon its very first update for all initial conditions unless its regulatory function is a constant. Source nodes remain in their initial state, which nevertheless will be different for different initial conditions.

2.4. Determining the Logical Domain of Influence of a Sustained Node State

We propose to find the LDOI of a node state by doing a modified breadth first search (BFS) on the expanded network (see the pseudocode in Supplementary Material section 1.1). In order to find the LDOI of $\sigma_i = \tilde{\sigma}_i$, we start the search from the corresponding node n_i on the expanded network if $\sigma_i = 1$ or we start the search from the complementary node $\sim n_i$ if $\sigma_i = 0$. If we meet another non-composite node, we add this node to the LDOI; if we meet a composite node, we add this composite node only if all of its parent nodes (i.e., regulators) are already part of the LDOI. This is due to the fact that any edge from a node to a non-composite node represents a sufficient relationship and any edge from a node to a composite node represents a necessary relationship. We keep searching on the expanded network until no new nodes can be added to the LDOI. For example, in **Figure 1B**, one can readily see that $\mathcal{LD}(\sigma_1 = 1) \equiv \mathcal{LD}(n_1) = \{n_4, \sim n_2, n_3, n_1, \sim n_0\}$ following the described search procedure. The first difference from a normal BFS to find a connected component starting from a node is that we put an extra rule for including a composite node. Another subtle difference is that we do not include the starting point unless we visit this starting point again in our search process.

During the search process, there is a possibility that we meet the negation of the starting point. This reflects the possibility that a node state can indirectly lead to the opposite state through a negative feedback loop. This outcome represents a conflict with the original intervention. We do not add this node to the LDOI because we assume that the intervention can sustain the original node state, thus the opposite state is not reachable. This truncation of the LDOI to avoid including the negation of the starting node state ensures that the LDOI will not contain a node which is the negation of an already visited node. Mathematically, if a non-composite node $n_i^{ex} \in \mathcal{LD}(n_j^{ex})$, then n_j^{ex} is sufficient to activate n_i^{ex} , i.e., the long-term logical rule for n_i^{ex} can be expressed in the form $n_i^{ex} = n_j^{ex}$ OR \dots ; this implies $\sim n_i^{ex} = \sim n_j^{ex}$ AND \dots , i.e., $\sim n_j^{ex}$ is necessary to activate $\sim n_i^{ex}$. Thus any conflict between n_i^{ex} and $\sim n_i^{ex}$ will occur after the conflict between n_j^{ex} and $\sim n_j^{ex}$ during the search process. This

truncation of the LDOI is the third difference compared with a normal BFS.

For example, in the network of **Figure 1D**, the LDOI of the complementary node $\sim n_4$ includes nodes $n_3, \sim n_0, n_1, \sim n_2$ following the search procedure. From n_1 one can also reach node n_4 , which is the negation of the considered intervention. Thus we stopped this branch of searching based on our truncation rule. Since there are no more nodes that can be added, we conclude that $\mathcal{LD}(\sim n_4) = \{n_3, \sim n_0, n_1, \sim n_2\}$.

Our LDOI search procedure is equivalent to doing a simulation on the expanded network. If we update the system corresponding to the BFS order of the expanded network starting from the intervention node (i.e., we update node i if we visited n_i^{ex} on the expanded network), all the updated nodes are guaranteed to stabilize in the corresponding visited state on the expanded network, i.e., as in the LDOI of that node. In the example of **Figure 1**, as discussed above, $\mathcal{LD}(n_1) = \{n_4, \sim n_2, n_3, n_1, \sim n_0\}$. If we update the nodes in the order 4, 2, 3, 1, 0, each node will stabilize in the state as in $\mathcal{LD}(n_1)$. We note that this does not put a restriction on the updating regime: if we update the system in an arbitrary order, each node in the LDOI of the given sustained intervention will attain a stationary state in the first update after all of its regulators included in the LDOI have been updated once. For example, if we fixed the node 1 to be ON and we perform rounds of update of the nodes in the order 0, 1, 2, 3, 4, nodes 2, 3, and 4 will be stabilized in the first round of updating, while nodes 0 and 1 will be stabilized in the second round.

The difference between the LDOI and DOI is that LDOI requires the nodes to be stabilized when being updated for the first time, while DOI just requires the nodes to be stabilized in finite time. Thus one can see that the LDOI of a node will be a subset of the DOI of a node. In many cases the two concepts give the same result. Two exceptions are illustrated in **Figure 2**. In both cases the DOI of an intervention contains more nodes than the LDOI of this intervention. This is because certain nodes may stabilize not because of the influence of the intervention but because of the collective effect of two inconsistent feedback loops or because of a stable motif stabilized by an oscillation. In the network of **Figure 2A**, the three regulators of node B are independent and the network includes both a positive and a negative feedback loop. To analyze the LDOI of $A = 1$, taking the feedback effect of C and D on B into consideration, the regulatory function of B is simplified into $\sigma_B(t + \tau_B) = \sigma_B(t - \tau_C) \text{ OR NOT } (\sigma_B(t - \tau_D))$, where τ_i is the discrete time delay for node i , as introduced in section 2.1. This regulatory function admits a constant solution $\sigma_B = 1$ regardless of the values of the time delays (Saadatpour et al., 2010; Azuma et al., 2014). It may additionally admit an oscillatory solution for strict relationships among the time delays. In the cases where there is no oscillatory solution, for example in the cases where only one node can change state at a time, $\mathcal{D}(A) = \{B, C, \sim D\}$, as the stabilization of B leads to the stabilization of C and D as well. However, $\mathcal{LD}(A) = \emptyset$ as the activation of the composite node requires nodes $A, \sim C, \sim D$ on the expanded network shown in **Figure 2B** and thus we cannot add the composite node to the LDOI of node A. In the example shown in **Figure 2C**, the two regulators are independent for node B, $\mathcal{D}(C) = \{B\}$ as the

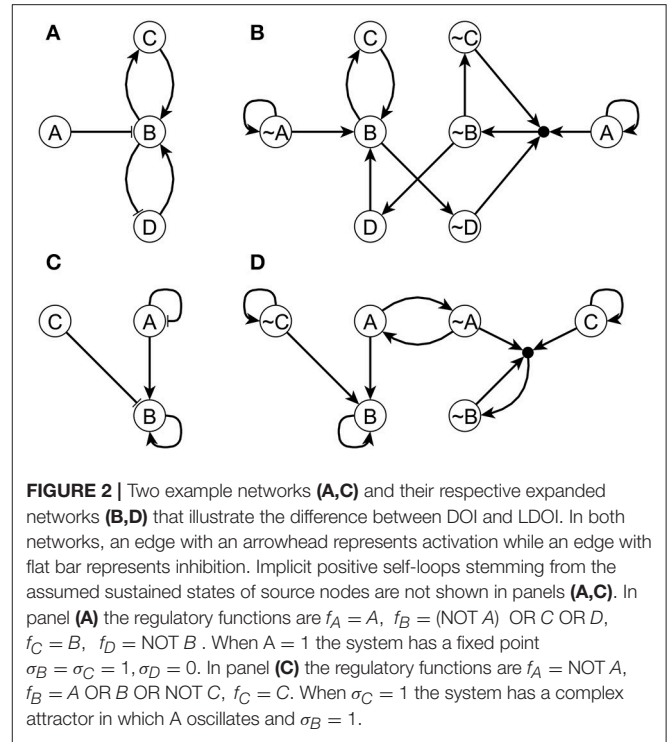


FIGURE 2 | Two example networks (A,C) and their respective expanded networks (B,D) that illustrate the difference between DOI and LDOI. In both networks, an edge with an arrowhead represents activation while an edge with flat bar represents inhibition. Implicit positive self-loops stemming from the assumed sustained states of source nodes are not shown in panels (A,C). In panel (A) the regulatory functions are $f_A = A$, $f_B = (\text{NOT } A) \text{ OR } C \text{ OR } D$, $f_C = B$, $f_D = \text{NOT } B$. When $A = 1$ the system has a fixed point $\sigma_B = \sigma_C = 1, \sigma_D = 0$. In panel (C) the regulatory functions are $f_A = \text{NOT } A$, $f_B = A \text{ OR } B \text{ OR NOT } C$, $f_C = C$. When $\sigma_C = 1$ the system has a complex attractor in which A oscillates and $\sigma_B = 1$.

negative feedback loop of node A will make A oscillate, but B will stabilize into the ON state after the first time that A visits the ON state and activates B, while $\mathcal{LD}(C) = \emptyset$ for the same reason as in the last example.

2.5. Properties of the Logical Domain of Influence of a Sustained Node State

In order to further illustrate the concept of LDOI, we discuss a few of its properties and its relationship with established concepts in Boolean dynamics. The LDOI of a node state is mathematically equivalent to the three-valued logical steady state that results when this node state is fixed (Klamt et al., 2006; Samaga et al., 2010). Here the three values are 0, 1, and unknown (for nodes who do not attain a stationary state solely due to the original node's fixed state). The LDOI of a node state is also equivalent to the set of nodes, and corresponding states, that are identified using network reduction techniques [i.e., by iteratively substituting the fixed node state(s) into the regulatory function(s) of target nodes] (Bilke and Sunnesson, 2001; Naldi et al., 2012; Saadatpour et al., 2013). Previous analysis (Samaga et al., 2010; Saadatpour et al., 2013) identified that if node i has a single outgoing edge, and is a sufficient regulator of its sole target node, j , the LDOI of the ON state of i contains the LDOI of the ON state of j . Here we study in general the possible inclusion relationship between the logic domains of influence of two node states $\sigma_i = \tilde{\sigma}_i$ and $\sigma_j = \tilde{\sigma}_j$ in the case when $\sigma_j = \tilde{\sigma}_j \in \mathcal{LD}(\sigma_i = \tilde{\sigma}_i)$ or $n_j^{ex} \in \mathcal{LD}(n_i^{ex})$ in the expanded network notation, where n_i^{ex} and n_j^{ex} represent any non-composite node in the expanded network. In a directed graph, if node n_j is a reachable from node n_i , all descendants of n_j will also be reachable from n_i ; indeed one can

easily prove this by contradiction. However, due to the special properties of the expanded network and the truncation of the LDOI, this inclusion relationship $\mathcal{LD}(n_j^{ex}) \subseteq \mathcal{LD}(n_i^{ex})$ is not generally true for the expanded network. It is possible that $n_j^{ex} \in \mathcal{LD}(n_i^{ex})$, however, $\sim n_i^{ex} \in \mathcal{LD}(n_j^{ex})$. In this case, by definition of the LDOI, we won't allow the negation of a node state to be part of the LDOI of a node state. For example, $n_1 \in \mathcal{LD}(\sim n_4)$, however, $n_4 \in \mathcal{LD}(n_1)$. Thus $\mathcal{LD}(n_1) \not\subseteq \mathcal{LD}(\sim n_4)$.

If we add an additional restriction on the two nodes, this inclusion relationship will hold the same way as for descendants in a directed graph. To be specific, the *first key property of the LDOI* is, if the node state $\sigma_i = \tilde{\sigma}_i$ and $\sigma_j = \tilde{\sigma}_j$, corresponding to the two non-composite node n_i^{ex} and n_j^{ex} on the expanded network, are both included in the same (partial) fixed point and $n_j^{ex} \in \mathcal{LD}(n_i^{ex})$, the LDOI of n_j^{ex} will be a subset of the LDOI of n_i^{ex} , i.e., $\mathcal{LD}(n_j^{ex}) \subseteq \mathcal{LD}(n_i^{ex})$. (Recall that a partial fixed point is a subset of nodes whose respective state remains unchanged after being updated regardless of the states of the nodes excluded from this subset.) The reason why the inclusion relationship holds is that node states in a (partial) fixed point stabilize in the long term, thus they will not lead to a situation with opposing behavior $n_j^{ex} \in \mathcal{LD}(n_i^{ex})$ and $\sim n_i^{ex} \in \mathcal{LD}(n_j^{ex})$. This restriction can be weakened to only require that node state n_i^{ex} is in a (partial) fixed point. The reason is that if $n_j^{ex} \in \mathcal{LD}(n_i^{ex})$ and n_i^{ex} is in a (partial) fixed point, then n_j^{ex} must also be in the same (partial) fixed point, or be a node whose state stabilizes due to the nodes in the partial fixed point. Also, as one or more stable motifs are part of a (partial) fixed point, the conclusion will be true if one replaces “(partial) fixed point” by “stable motif” in the above statement. In the example of **Figure 1**, as nodes $n_1, \sim n_2$ and n_3 form a stable motif and its corresponding (partial) fixed point is $(\sigma_1, \sigma_2, \sigma_3) = (1, 0, 1)$ as shown in **Figure 1C**, which also lead to the stabilization of the remaining two nodes as $\sigma_0 = 0$ and $\sigma_4 = 1$, thus $n_3 \in \mathcal{LD}(n_1)$ implies that $\mathcal{LD}(n_3) \subseteq \mathcal{LD}(n_1)$. In fact, $\mathcal{LD}(n_3) = \mathcal{LD}(n_1) = \{n_4, \sim n_2, n_3, n_1, \sim n_0\}$. Also $n_4 \in \mathcal{LD}(n_1)$ implies that $\mathcal{LD}(n_4) \subseteq \mathcal{LD}(n_1)$. Note that only n_1 is part of the stable motif or partial fixed point in the latter example, n_4 is not.

As stable motifs represent generalized positive feedback loops of the Boolean network (Zafundo and Albert, 2013), we explore the relationship between stable motifs and the LDOI of a node state. The *second key property of LDOI* is, if the LDOI of a node state contains this node state itself, the LDOI contains a stable motif. As the LDOI of a node state only contains the node state itself if we meet this node during the search process on the expanded network, this indicates the existence of a positive feedback loop, which is the intuition why this proposition holds. (A sketch of proof from the dynamical standpoint is included in Supplementary Material section 2.2). For example, $n_1 \in \mathcal{LD}(n_1)$ implies that there exists a stable motif contained in $\mathcal{LD}(n_1)$, indeed, $SM_1 = \{n_1, \sim n_2, n_3\} \subseteq \mathcal{LD}(n_1)$.

2.6. The Domain of Influence of a Node State Set

Now we generalize the concept of DOI of a single node state to DOI of a node state set (i.e., a set of nodes, each in a sustained state). We define the DOI of a node state set as all the node

states that can be stabilized in the long term by the given set of node states under all initial conditions in any updating regime. Mathematically, $\mathcal{D}(\{\sigma_i = \tilde{\sigma}_i\}) = \{\{\sigma_j = \tilde{\sigma}_j\} : \sigma_j(t) = \tilde{\sigma}_j \text{ as } t \rightarrow \infty \text{ for any } (\sigma_1(t=0), \dots, \sigma_k(t=0)) \text{ when } \sigma_i(t) = \tilde{\sigma}_i \text{ for any } t > 0\}$, where $\{\sigma_i(t) = \tilde{\sigma}_i\}$ represents the intervention consisting of a specific set of node states. Note that the following two notations are equivalent: $\mathcal{D}(\{\sigma_i = \tilde{\sigma}_i\}) \equiv \mathcal{D}(\{n_i^{ex}\})$. Similarly, we define the LDOI of a node state set, $\mathcal{LD}(\{\sigma_i = \tilde{\sigma}_i\})$, as all the nodes that can be stabilized by the first update in any BFS order-preserving (on the expanded network) update order starting from this given set of node states under all initial conditions. As in the single node state case, the LDOI of a node state set will be a subset of the DOI of the same node state set.

The LDOI of a node state set can be determined by a modified BFS on the expanded network, now using multiple starting points. This does not add complexity to the iterative implementation of BFS: we just need to initialize the queue with the set of given node states. Similar to the case of finding the LDOI of a single node state, we need to deal with the conflicts that may occur during the search process. To be precise, conflict means that during the search we visit a node state that is the negation of a node state included in the intervention. Two types of conflict can arise. First, a node state in the given set may be impacted by negative feedback and have a LDOI that was truncated to avoid containing its own negation. Second, the LDOI of two node states n_i^{ex} and n_j^{ex} may have the property $\sim n_i^{ex} \in \mathcal{LD}(n_j^{ex})$ or $\sim n_j^{ex} \in \mathcal{LD}(n_i^{ex})$, or both. In other words, node i may regulate node j (or *vice versa*) in a way that is incompatible with the intervention (e.g., a node whose sustained activity is part of the intervention may negatively regulate another node whose sustained activity is part of the intervention). We call intervention sets that have either type of conflict incompatible sets; we refer to the rest of the intervention sets as compatible sets. Similarly to the truncation we did to find the LDOI of a single node state, we do not include any node state that is the negation of any node state given in the intervention set and we stop searching that branch. We note that this truncation strategy avoids any following conflict. For example, if $n_C \in \mathcal{LD}(n_A)$ and $\sim n_C \in \mathcal{LD}(n_B)$, then one may expect that the LDOI of the set $\{n_A, n_B\}$ will have a conflict between n_C and $\sim n_C$. However, $n_C \in \mathcal{LD}(n_A)$ implies that $\sim n_C$ requires $\sim n_A$, this means that meeting the conflict between n_C and $\sim n_C$, must be after meeting the conflict between n_A and $\sim n_A$, which is avoided by our truncation strategy.

For a compatible set $\{n_i^{ex}\} \equiv \cup_i n_i^{ex}$, it is guaranteed that $\cup_i \mathcal{LD}(n_i^{ex}) \subseteq \mathcal{LD}(\cup_i n_i^{ex})$. For example, as shown in **Figure 1E**, the node set $\{n_2, n_4\}$ is a compatible node set as $\mathcal{LD}(n_2) = \emptyset$, $\mathcal{LD}(n_4) = \emptyset$ and $\mathcal{LD}(\{n_2, n_4\}) = \{\sim n_3, n_0, n_4, \sim n_1, n_2\}$. Note $\mathcal{LD}(n_2) \cup \mathcal{LD}(n_4) \subseteq \mathcal{LD}(\{n_2, n_4\})$. However, for an incompatible set, we just know that the situation $\cup_i \mathcal{LD}(n_i^{ex}) \not\subseteq \mathcal{LD}(\cup_i n_i^{ex})$ cannot happen and all the remaining situations are possible. In the network of **Figure 1**, node set $\{n_2, \sim n_4\}$ is an incompatible node set as $\mathcal{LD}(n_2) = \emptyset$, $\mathcal{LD}(\sim n_4) = \{n_3, \sim n_0, n_1, \sim n_2\}$, and $\mathcal{LD}(\{n_2, \sim n_4\}) = \{n_3, \sim n_0, n_1\}$. Note that neither n_4 nor $\sim n_2$ are included in $\mathcal{LD}(\{n_2, \sim n_4\})$ due to the truncation rule and $\mathcal{LD}(\{n_2, \sim n_4\}) \subsetneq \mathcal{LD}(n_2) \cup \mathcal{LD}(\sim n_4)$. Node set $\{\sim n_1, n_3\}$ is another incompatible set as $\mathcal{LD}(\sim n_1) = \{n_2\}$, $\mathcal{LD}(n_3) =$

$\{\sim n_0, n_1, \sim n_2, n_4, n_3\}$ and $\mathcal{LD}(\{\sim n_1, n_3\}) = \{n_2, \sim n_0, \sim n_4, n_3\}$. Note that $\mathcal{LD}(\{\sim n_1, n_3\}) \not\subseteq \mathcal{LD}(\sim n_1) \cup \mathcal{LD}(n_3)$, and $\mathcal{LD}(\sim n_1) \cup \mathcal{LD}(n_3) \not\subseteq \mathcal{LD}(\{\sim n_1, n_3\})$.

The properties of the LDOI of a single node can also be generalized to the LDOI of a given node set. For the first key property, let $S_j = \{\sigma_j = \tilde{\sigma}_j\}$ and $S_i = \{\sigma_i = \tilde{\sigma}_i\}$ be two sets of node states, if S_j is a subset of any (partial) fixed point and $S_j \subseteq \mathcal{LD}(S_i)$, then $\mathcal{LD}(S_j) \subseteq \mathcal{LD}(S_i)$. The intuition is similar, the requirement restricting us to consider those nodes which can be stabilized in the long term, that is, we rule out the possibility of S_i being an incompatible node set. For example in **Figure 1** consider $S_i = \{\sim n_3\}$ and $S_j = \{n_2, n_4\}$. As $\sim n_3$ is part of the stable motif $SM_2 = \{n_0, \sim n_1, n_2, \sim n_3, n_4\}$, corresponding to the fixed point $(\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4) = (1, 0, 1, 0, 1)$, $S_j \subseteq \mathcal{LD}(S_i)$ implies $\mathcal{LD}(S_j) \subseteq \mathcal{LD}(S_i)$. In fact, $\mathcal{LD}(S_j) = \mathcal{LD}(S_i)$.

The second key property also generalizes: if the LDOI of a given node state set contains the set itself, then the LDOI of the set contains at least one stable motif. The intuition and proof is similar to the case of a single node state. Taking the same example, consider $S_i = \{\sim n_3\}$ and $S_j = \{n_2, n_4\}$, note that both $S_i \subseteq \mathcal{LD}(S_i)$ and $S_j \subseteq \mathcal{LD}(S_j)$, this implies that both $\mathcal{LD}(S_i)$ and $\mathcal{LD}(S_j)$ contain a stable motif, which is SM_2 in this case.

Following these examples, we define the core of a stable motif to be a minimal subset of the stable motif whose LDOI contains the stable motif. Here by minimal we mean that no true subset of the core of the stable motif will contain the entire stable motif. The core of a stable motif can be a single node or more than one node. For example, as shown in **Figure 1C** $\sim n_3$ is a single-node core of the stable motif $SM_2 = \{n_0, \sim n_1, n_2, \sim n_3, n_4\}$. $\{n_2, n_4\}$ is another core of the same stable motif as $SM_2 \not\subseteq \mathcal{LD}(n_2)$, $SM_2 \not\subseteq \mathcal{LD}(n_4)$, and $SM_2 \subseteq \mathcal{LD}(\{n_2, n_4\})$.

We also define a driver node (set) of the stable motif to be a node (set) whose DOI contains the entire stable motif. The driver node (set) can be inside the stable motif, in which case it is the core of the stable motif; it can also be an upstream node that is sufficient to activate (the core of) the stable motif. We note that stabilization of a stable motif does not require the sustained state of a driver node, that is, oscillations can also lead to the stabilization of a stable motif. An example of this behavior was shown in **Figure 2B**: node B, which constitutes a self-sustaining stable motif, can stabilize by a single instance of $A = 1$, regardless of the fact that the negative self-regulation of A makes it oscillate.

2.7. Target Control Algorithm

Now that we have equipped ourselves with the tool of LDOI to find the long term effect of a sustained intervention, we can formulate the target control problem as the identification of a node set S^* whose LDOI contains the target node state set, i.e., $\mathcal{LD}(S^*) \supseteq Target$. This problem can be framed as a planning search problem (Russell and Norvig, 2003). We start with a null set whose LDOI is also null. We repeatedly add a new node to the set until the LDOI of this set contains the target node state set. We use LDOI instead of DOI for this purpose because identification of the DOI is a computationally more difficult problem. Our current solution using LDOI sets a tight upper bound for the optimal solution for the target control problem as $\mathcal{D}(S^*) \supseteq \mathcal{LD}(S^*) \supseteq Target$.

Previous work in the target control of Boolean models has focused on full enumeration of the solutions for the target control problem (Klamt et al., 2006; Samaga et al., 2010), which can be used to identify the solutions that involve combinations of a small number of nodes but is not generally viable because of combinatorial explosion. In our work, we use a complementary approach to avoid a full state space search in this combinatorial search problem. We apply a random heuristic algorithm called the greedy randomized adaptive search procedure (GRASP) (Pardalos et al., 1998; Festa et al., 2001). The pseudocode is described in Algorithm (**Tables 1, 2**). The algorithm consists of two main phases. The first phase is the construction of a greedy randomized solution and the second phase is a local search to remove any redundancy of the solution.

Algorithm 1 GRASP algorithm for Target Control Problem

```

1: procedure GRASP( $G\_expanded, Target, max\_itr$ )
2:    $solutions \leftarrow List()$ 
3:   for  $index \leftarrow 1, max\_itr$  do
4:      $solution \leftarrow ConstructGreedyRandomizedSolution$ 
                                     ( $G\_expanded, Target$ )
5:      $solution \leftarrow LocalSearch(G\_expanded, Target, solution)$ 
6:     if  $solution$  then
7:        $Solutions.append(solution)$ 
8:     end if
9:   end for
10:  return  $solutions$ 
11: end procedure

```

Algorithm 2 Algorithm for constructing a greedy randomized solution

```

1: procedure CONSTRUCTGREEDYRANDOMIZEDSOLUTION
                                     ( $G\_expanded, Target$ )
2:    $solution \leftarrow Set()$ 
3:    $\alpha \leftarrow random(0, 1)$ 
4:    $candidates \leftarrow Construct\_Initial\_Candidates(G\_expanded,$ 
                                                      $Target)$ 
5:    $G(v) \leftarrow Construct\_Greedy\_Functions(G\_expanded,$ 
                                                $candidates)$ 
6:   while  $candidates$  do
7:      $RCL \leftarrow MakeRCL(candidates, G(v), \alpha)$ 
8:      $s \leftarrow Select\_Candidate(RCL)$ 
9:      $solution \leftarrow solution \cup \{s\}$ 
10:    if  $Target \subset LDOI(solution)$  then
11:      return  $solution$ 
12:    end if
13:     $Update\_Candidates(candidates)$ 
14:  end while
15:  return  $Set()$ 
16: end procedure

```

In the first phase, we first generate an initial candidate list (line 4 in Algorithm 2). In the simplest case, the initial candidate list is all the non-composite nodes of the expanded network except

the nodes in the target set and their negation, both of which are ineligible for control. One can also be more selective to adapt to the specific needs of controlling biological systems. For example, we can forbid the use of certain nodes or node states when constructing the initial candidate list, to incorporate the fact that certain chemical species are harder or even unrealistic to control. Thus these nodes/chemical species will never appear in the final solution since they are not in the initial candidate list.

Then, we begin the procedure of iteratively adding nodes to the trial solution set (which is initially empty) and evaluating whether the LDOI of the trial solution set covers the target set. We form a restricted candidate list (RCL, line 7 in Algorithm 2) based on a greedy measure $G(v)$ defined for each candidate node v in the candidate list (line 5 in Algorithm 2). A greedy function is a heuristic score to estimate whether this node should be included in the solution. We evaluated five choices of $G(v)$, as described at the end of this section and in section 3.1. We determine the minimum score $G_{min} = \min_{v \in V} G(v)$ and maximum score $G_{max} = \max_{v \in V} G(v)$ among the heuristic scores of all the nodes. Then we use a previously generated random number α from a uniform distribution between 0 and 1 to set a passing score for the RCL as $G_{pass} = G_{min} + \alpha \cdot (G_{max} - G_{min})$. Then the RCL consist of nodes whose greedy function is no less than the passing score, i.e., $RCL = \{v \in V | G(v) \geq G_{pass}\}$. This procedure of generating RCL is summarized in Supplementary Material section 1.3.

Next we randomly pick a node from the RCL and add it to the current trial solution (line 8 and 9 in Algorithm 2). The trial solution is used as the *source* node set of the LDOI algorithm (whose pseudocode is presented in Supplementary Material section 1.1). If the target set is contained in the set of nodes returned by the LDOI algorithm, we end the first phase and start the second phase (local search procedure) with this candidate solution (line 10 and 11 in Algorithm 2). Otherwise, we update the candidate node set and start the next iteration toward adding another node from the RCL to the trial solution set. We update the candidate node set by removing the previously added node, its negation and any node in the LDOI of the current trial solution (line 13 in Algorithm 2). We do this latter exclusion because these nodes will stabilize because of the current trial solution, and it is useless to add any stabilized state to the trial solution. We repeat the whole procedure including selecting a node randomly from the candidate set as long as there are still candidate nodes (line 6 in Algorithm 2). We return an empty set if we do not find a solution (line 15 in Algorithm 2).

In the second phase (see the pseudocode in Supplementary Material section 1.2), we start with a candidate solution that covers the target set. We randomize the order of nodes in the candidate solution and then iteratively attempt to remove each node. If after removing this node the LDOI of the modified solution still covers the target set, then we replace the candidate solution with the modified solution. Thus after one iteration of traversing all the nodes, we obtain a final solution. At worst, no node is removed from the set and the final solution is the same as the candidate solution. The randomness in the removal order provides a possibility for obtaining different minimal solutions from the same candidate solution.

In this random heuristic algorithm, we introduce two aspects of randomness in the construction phase, one is the randomness of the passing score by a different α for each iteration of solution generation process (line 3 in Algorithm 1) and another is the random selection of a node each time from the RCL inside each solution generation process (line 8 in Algorithm 2). These techniques help strike a balance between the bias of a greedy function and exploring the whole node state space (Pardalos et al., 1998; Festa et al., 2001). An efficient greedy function/heuristic score is important to guide the search procedure toward the subspace with the optimal solution. However, a universally efficient greedy function may not exist; rather, the efficiency of a greedy function may depend on the specific network structure and target set. We have implemented five choices of greedy functions $G(v)$ for a given node state (equivalently, non-composite node of the expanded network): score 1 is the size of the LDOI of that node state (denoted as $|LDOI|$); score 2 is the size of the set of composite nodes which are nearest neighbors of the LDOI of that node state (denoted as $|Comp_LDOI|$); score 3 is a linear combination of the previous two measures with equal weight (denoted as $Scores_1+2$), and score 4 and 5 as the size of the LDOI of that node state with penalty if the LDOI contains a node that is the negation of a node in the target set (denoted as $|LDOI|_Pen1$ and $|LDOI|_Pen2$). The penalty can be implemented by multiplying this score by -1 (score 4) or by decreasing this score by the size of the largest LDOI among all node states (score 5); both of these implementations ensure that this score becomes non-positive. A python implementation of the target control algorithm is available at <https://github.com/yanggangthu/BooleanDOI>.

2.8. Computational Complexity of the Target Control Algorithm

The time complexity of calculating the LDOI of any set is bounded by $O(N_{ex} + E_{ex})$, where N_{ex} is the number of nodes and E_{ex} is the number of edges of the expanded network. For each non-composite node in the network, we initially calculate its LDOI and the value of its greedy function, with time complexity $O(N(N_{ex} + E_{ex}))$, where N is the number of nodes in the original network. We then cache these results to improve the performance of the GRASP algorithm. In the first phase of the GRASP algorithm, we run at most N iterations and we need to calculate the LDOI of the trial solution in each iteration, thus the time complexity is bounded by $O(N(N_{ex} + E_{ex}))$. In the second phase, the time complexity is also bounded by $O(N(N_{ex} + E_{ex}))$ as we need to go through each node, bounded by $O(N)$ as a crude estimate, delete the node from the solution and check the modified solution's LDOI, which is $O(N_{ex} + E_{ex})$. The Boolean regulatory functions of biological network models are often nested canalizing rules (Kauffman et al., 2003; Li et al., 2013), thus for each node with k regulators there are at most k newly generated composite nodes in the expanded network, as well as two corresponding non-composite nodes; each of these nodes have at most k regulators. Thus N_{ex} is bounded by $O(kN)$, and E_{ex} is bounded by $O(k^2N)$. Biological networks are sparse, with an average node in-degree $1 < k < 3$ (Newman, 2010).

Thus the complexity of the target control algorithm applied to biological network models is $O(k^2N^2) \sim O(N^2)$ for a well-behaved degree distribution in the sparse limit and bounded by $O(N^3)$ for an extremely skewed degree distribution in the sparse limit. Different iterations of the solution generation process (line 3 in Algorithm 1) can be easily parallelized as each iteration is independent. The space complexity of BFS search on the expanded graph is bounded by $O(N_{ex})$, and the space complexity of the entire procedure is bounded by time complexity times the storage space of LDOI of a node set, which is bounded by $O(N_{ex})$.

2.9. Damage Mitigation as Target Control

We can generalize the target control algorithm to solve a damage mitigation problem. Consider a Boolean network that has two steady states, one corresponding to the normal state of the system and the other corresponding to a disease state. The system is currently in the normal steady state, but damage to a node, which causes it to stabilize in the opposite state, will lead the system to the disease steady state without any intervention. Under such conditions, previous research has proposed modifying the network topology (as soon as possible, or preventatively) to block the propagation of damage (Yang et al., 2016). Here we are interested in designing a damage mitigation strategy to bring the system back to an attractor similar to the normal steady state in the sense that a subset of nodes are in the same state as their states in the normal steady state. This problem is almost the same as the target control problem except that we need to take the permanent damage into consideration. There are two ways of incorporating this. First, we treat this permanent damage as an initial condition and apply network reduction to the system. However, this risks reducing a significant fraction of the nodes in the network, including the target nodes we are interested in. Second, we can apply our GRASP algorithm as above while initializing the solution with the damaged node state(s) and forbidding the damaged node state to be removed in the local search phase in GRASP algorithm. This means that we include the damage as part of the intervention. When the LDOI of the node state set containing the damage effect covers the target set, the target nodes will stabilize in their desired states after a finite number of time steps under all initial conditions of the subspace of the damaged network. We note that we only need to do this when the damage is a permanent one; when the damage is temporary (i.e., when the node is allowed to go back to its original state), this can be treated as a different initial condition for the target control problem and we can still apply our GRASP algorithm to solve it as DOI/LDOI is robust to any initial condition by definition.

3. RESULTS

3.1. Application to Ensembles of Random Boolean Networks

We tested the two proposed properties of the LDOI and the target control algorithm on different random Boolean network ensembles. Specifically, we generated an ensemble of 1000 Erdős Rényi random graphs (Newman, 2010) (using the `gnm_random_graph()` function of NetworkX; Hagberg et al.,

2008), with size ranging from 15 to 50 nodes and average in-degree ranging from 1 to 2. The Boolean regulatory functions of the random ensemble are required to be effective (irreducible) Boolean functions (Zertuche, 2009) to be consistent with the generated topology, or nested canalizing functions to simulate biological systems. (A nested canalizing Boolean function with k inputs can be generated by determining two sequences, the input sequence (I_1, I_2, \dots, I_k) and the output sequence (O_1, O_2, \dots, O_k) , where I_i or O_i is either 0 or 1. The output o as a function of input configuration (i_1, \dots, i_k) is thus determined through the hierarchy $o = O_1$ if $i_1 = I_1$; $o = O_2$ if $i_1 \neq I_1$ and $i_2 = I_2$; \dots ; $o = O_k$ if $i_1 \neq I_1, \dots, i_{k-1} \neq I_{k-1}, i_k = I_k$; $o = NOT O_k$ if $i_1 \neq I_1, \dots, i_{k-1} \neq I_{k-1}, i_k \neq I_k$.) We have successfully tested and validated the two properties for the LDOI of each node in the generated networks. We also tested and validated the properties of the LDOI of node sets of size up to 3~7 depending on the specific network (as the complexity of testing the property grows faster than N^k for $k \ll N$, where N is the network size and k is the node set size).

With respect to testing the target control algorithm, we generate 50 random target sets with size 2 or 3 for each random network. We calculate the average number of generated solutions for each pair formed by a target set and a network. As shown in Table 1, the average number of solutions is significantly high, between 10 and 40 for ensembles with nested canalizing functions and between 25 and 70 for ensembles with effective Boolean functions. The dominance of the canalizing variables in determining certain outcomes tends to yield sparser expanded networks than non-canalizing functions, and fewer effective interventions. This is reflected in the smaller number of solutions in the ensembles with nested canalizing functions compared to the ensembles with effective Boolean functions. As shown in Figure S1, the average time for finding solutions for a target set (through 500 iterations) is 100 s or less for networks with 15–35 nodes and 20–60 edges. As expected, the runtime increases with the number of nodes and edges, reaching 600 s for 50 nodes and 100 edges. The relatively slow increase and practical runtime suggest that our algorithm is effective for logical models of biological systems.

It is not always possible to find a solution for a specific target set for a network, especially when the Boolean network model

TABLE 1 | Mean number of solutions found for each target set and random network pair for 50 target sets and 1,000 networks.

Custom score index and notation	1 LDOI	2 Comp_LDOI	3 Scores_1+2	4 LDOI_Pen1	5 LDOI_Pen2
Nested canalizing rules	12.29	31.66	12.30	31.21	40.64
Effective boolean rules	26.21	61.94	26.22	57.08	66.91

Half of 50 target sets have size two and the other half is of size three; none of them contain source nodes. The 2nd to 6th columns correspond to different custom score (greedy function) indexes and notations, which are described in the last paragraph in section 2.7. The second and third row corresponds to the random network ensemble with nested canalizing rules and effective Boolean rules, respectively.

does not have a (partial) fixed point type of attractor (i.e., if all nodes oscillate in the attractor) or when the desired target state set consists of node states that are part of different attractors, which conflict with each other. Another case where target control is impossible is when the target set is not reachable from the rest of the network. In the simulations of the two ensembles mentioned above, we verified that we are able to find a solution for more than 99.5% of the target sets when the target set satisfies two criteria: (i) it is a subset of a (partial) fixed point and (ii) the targets in this set are accessible from nodes outside of this set in the original network (that is, the targets do not consist of source nodes only and do not form a motif without any incoming edges). Note that there can be counter-examples where satisfying these criteria is not sufficient to find a solution. For example, in **Figures 2A,B**, there are no solutions for the target set $\{\sim B, \sim C\}$ as the remaining nodes are not enough to activate the composite node in **Figure 2B**. However, the probability of such situations is small in both random ensembles with moderate size and real biological network. Moreover, the fact that one cannot find a solution through our GRASP algorithm for the target control problem often indicates that the target set is not a reasonable target. It is likely that one would not be able to find a solution in such situation even with a whole state space search.

We also test the performance of different heuristic functions for the target control problem. We calculate the average number of generated solutions for each pair formed by a target set and a network. As shown in **Table 1**, greedy functions with a penalty for containing the negation of a node state included in the target set (score index 4 and 5) consistently perform better than the greedy functions directly using the size of the LDOI (score index 1 and 3). The intuition behind this is that it is more efficient to choose from those nodes whose DOI does not contain a conflict with the target. The second greedy function ($|Comp_LDOI|$) also performs quite well.

3.2. Biological Examples

We applied our methodology on four Boolean models of signal transduction networks. The four Boolean models are freely available on GitHub (<https://github.com/yanggangthu/BooleanDOI>) in SBML Qual format and in our custom format. In the following we demonstrate our algorithm on two of these, the epithelial-to-mesenchymal transition (EMT) network and the PI3K mutant ER+ breast cancer network. The results on the ABA induced stomatal closure network and the T-LGL leukemia network are shown in Supplementary Materials sections 3.3, 3.4. **Table 2** summarizes representative interventions found with our algorithm and compares them to the results of the most relevant previous analysis of these four biological network models. In Supplementary Data Sheet 1 we include the LDOI of each single node in the four models analyzed.

3.2.1. EMT Network

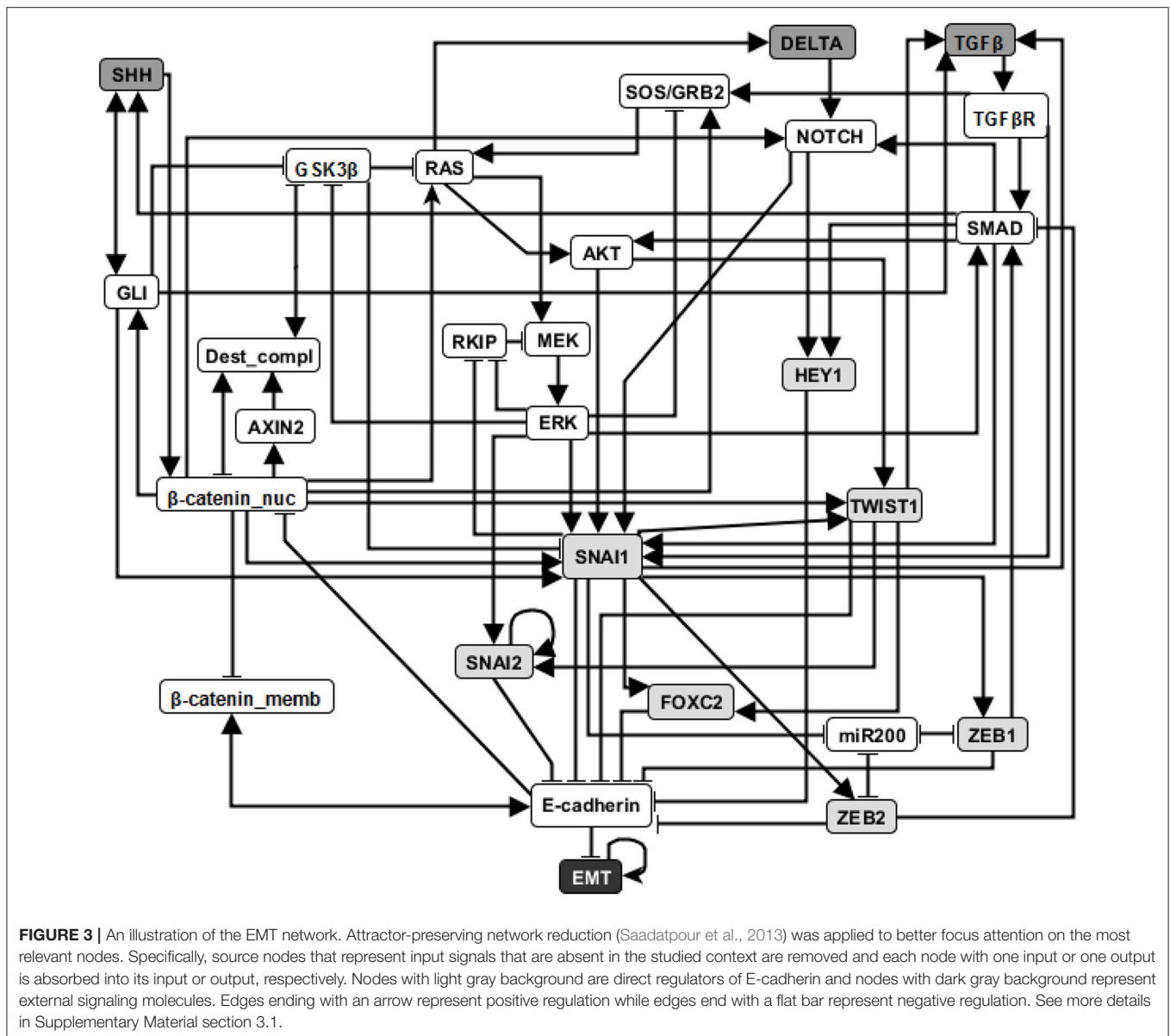
EMT is a cell fate change involved in embryonic development, which can be reactivated during cancer metastasis (Steinway et al., 2014). During EMT, epithelial cells lose their original adhesive property, and become mesenchymal cells which leave their primary site, invade neighboring tissue, and migrate to distant sites. A Boolean network model of EMT in the context of hepatocellular carcinoma invasion has been established by Steinway et al. (2014). Several predictions of this model were validated experimentally (Steinway et al., 2014, 2015). The EMT network has 70 nodes and 135 edges. The adhesion factor E-cadherin is the sink node; its OFF state indicates the transition to a mesenchymal state. The network has a normal (epithelial) steady state and an abnormal (mesenchymal) steady state. (See details in Supplementary Materials section 3.1). In **Figure 3** we show a simplified version of the EMT network; our analyses were done on the full network.

Previous research on this network has indicated that sustained activation of TGFβ signal can trigger EMT through the

TABLE 2 | Summary of representative target control solutions found by our algorithm for four biological network models.

Model	Target state(s)	Representative interventions found	Previous results from dynamic analysis	
			EMT-blocking	E cell inducing
EMT network	~EMT	β -catenin_memb = 1, SNAI1 = 0 β -catenin_memb = 1, SMAD = RAS = 0	SNAI = 0 SMAD = 0, RAS = 0	β -catenin_memb = 1, SMAD = RAS = 0, SNAI1 = GLI = 0
	~EMT, ~MEK	β -catenin_memb = 1, SNAI1 = RAS = 0 β -catenin_memb = 1, miR200 = RKIP = 1, RAS = 0	Not studied previously	
Breast cancer network	Apoptosis = 2, Proliferation = 0	PI3K = 0, ESR1 = 0	PI3K = 0, ESR1 = 0	
	Apoptosis <2, Proliferation >2 when PI3K = 0	PI3K = 0, ESR1 = 1	PI3K=0, ESR1 = 1	
ABA induced closure network	Closure = 1 when ABA = 0	Ca_c^{2+} = ROS = 1	ROS = 1	
	H ₂ O efflux when ABA = 0	K ⁺ efflux =1, SLAC1 = ROS = 1	Not studied previously	
T-LGL leukemia network	Apoptosis = 1	S1P = 0, RAS = 1	S1P = 0	

The first column indicates the relevant network model, the second lists one or two target states we considered, the third column presents the intervention set obtained by our algorithm, and the fourth column indicates the most relevant results of previous analysis of these network models. The previous analysis considered a unique initial state or a restricted family of initial states (in the case of the ABA network). The interventions found by our algorithm will be successful regardless of the initial state.



activation of eight stable motifs (Steinway et al., 2015). In addition, stabilization of any of these stable motifs can drive EMT. Our analysis of the LDOI of each node state indicates that any of 60 node states (out of 138 node states for the 69 nodes) can lead to EMT, including the previously established EMT drivers. Moreover, 43 node states (nodes of the expanded network) have the same LDOI, which contains 48 node states, including $EMT = 1$ (see Supplementary Data Sheet 1). Each of these 43 node states is either the core of one or more of the eight stable motifs, or an external driver of one or more of the eight stable motifs. Thus the EMT outcome and the mesenchymal steady state has a large basin of attraction. As we are more interested in designing therapeutic strategies to block the epithelial to mesenchymal transition, we set the negation of EMT as a

target. Previous analysis indicated that when considering an initial epithelial state and turning on the $TGF\beta$ signal, the knockout of any of the transcription factors that downregulate E-cadherin (i.e., knockout of SNAI1, SNAI2, FOXC2, TWIST1, ZEB1, ZEB2, HEY1) or multiple double node knockout combinations (knockout of SMAD and one of RAS, CSL, DELTA, NOTCH, NOTCH_{ic}, SOS/GRB2) are effective in blocking EMT (i.e., leading to $E-cadherin=ON$). The effectiveness of transcription factor knockout had been established in the literature; unfortunately these transcription factors cannot be targeted with existing drugs. Several double knockout combinations were validated experimentally in (Steinway et al., 2015) and are more amenable to drug targeting.

For EMT as target, our target control algorithm gives 7 two-node solutions (activation of β -catenin_{memb} and knockout of

any of *SNAI1*, *SNAI2*, *FOXC2*, *TWIST1*, *ZEB1*, *ZEB2*, *HEY1*) and 5 three-node solutions (activation of β -catenin_memb, knockout of *SMAD* and knockout of any of *RAS*, *CSL*, *DELTA*, *NOTCH*, and *NOTCH_ic*). The main difference between the target control solution and the previously found EMT-blocking single and double knockout interventions is that our target control solution includes the additional control of β -catenin_memb. To understand this difference, we note that EMT is in the LDOI of *TGF β* , however, EMT is not in the LDOI of the set consisting of *TGF β* together with any of the previously found EMT-blocking knockout interventions. This indicates that the knockout intervention is effective in the sense that it can block the process of reaching EMT. However, \sim EMT is also not in the LDOI of the set of *TGF β* together with any knockout intervention. The knockout intervention is effective when the initial condition is the epithelial steady state, however the knockout intervention does not block EMT for all initial conditions. The target control algorithm, which can block EMT for all initial conditions, requires one more node (β -catenin_memb) in the target control solution. In fact, treating this problem as a damage mitigation problem, where the damage is sustained activation of *TGF β* , we verify that EMT is in the LDOI of *TGF β* together with any of the target control solutions.

As established in previous results, the single node EMT-blocking knockouts do not lead back to an epithelial state but rather to hybrid epithelial or mesenchymal steady states (Steinway et al., 2015). The hybrid epithelial steady state has certain epithelial features, e.g., E-cadherin and β -catenin_memb are activated, and also some mesenchymal features, e.g., *MEK*, *ERK*, and *SNAI1* are activated. The hybrid mesenchymal steady state demonstrates the opposite features compared to the epithelial steady state. A good target set to avoid reaching such a hybrid state (which is likely pathological and may even be a worse outcome as the mesenchymal state) would be $\{\sim$ EMT, \sim MEK $\}$ (Steinway et al., 2015). The minimum solution found involves controlling three nodes: activation of β -catenin_memb, inhibition of *SNAI1*, inhibition of *RAS* or *RAF*. We also find a four-node intervention that does not involve *ERK* and *SNAI1*: activation of β -catenin_memb, *miR200* and *RKIP*, and also inhibition of *RAS*. If the target set is $\{\sim$ EMT, \sim MEK, \sim *SNAI1* $\}$, the minimum solution size is found to be six.

Stable motif control indicates that control of at least five nodes is needed to drive any initial state (including the mesenchymal state) to the epithelial state (see Supplementary Table 3 of Steinway et al., 2015) Although the control goal is different, one can still see the connection between our target control solution for the target \sim EMT and the stable motif control solution (to drive the system to the epithelial state). Specifically, they both require activation of β -catenin_memb. Knockout of *SNAI1*, knockout of *TWIST1*, or knockout of *SMAD* and *RAS*, as one of the target control solutions, also appear as a part of stable motif control solution that does not require control of *TGF β* or *TGF β R*.

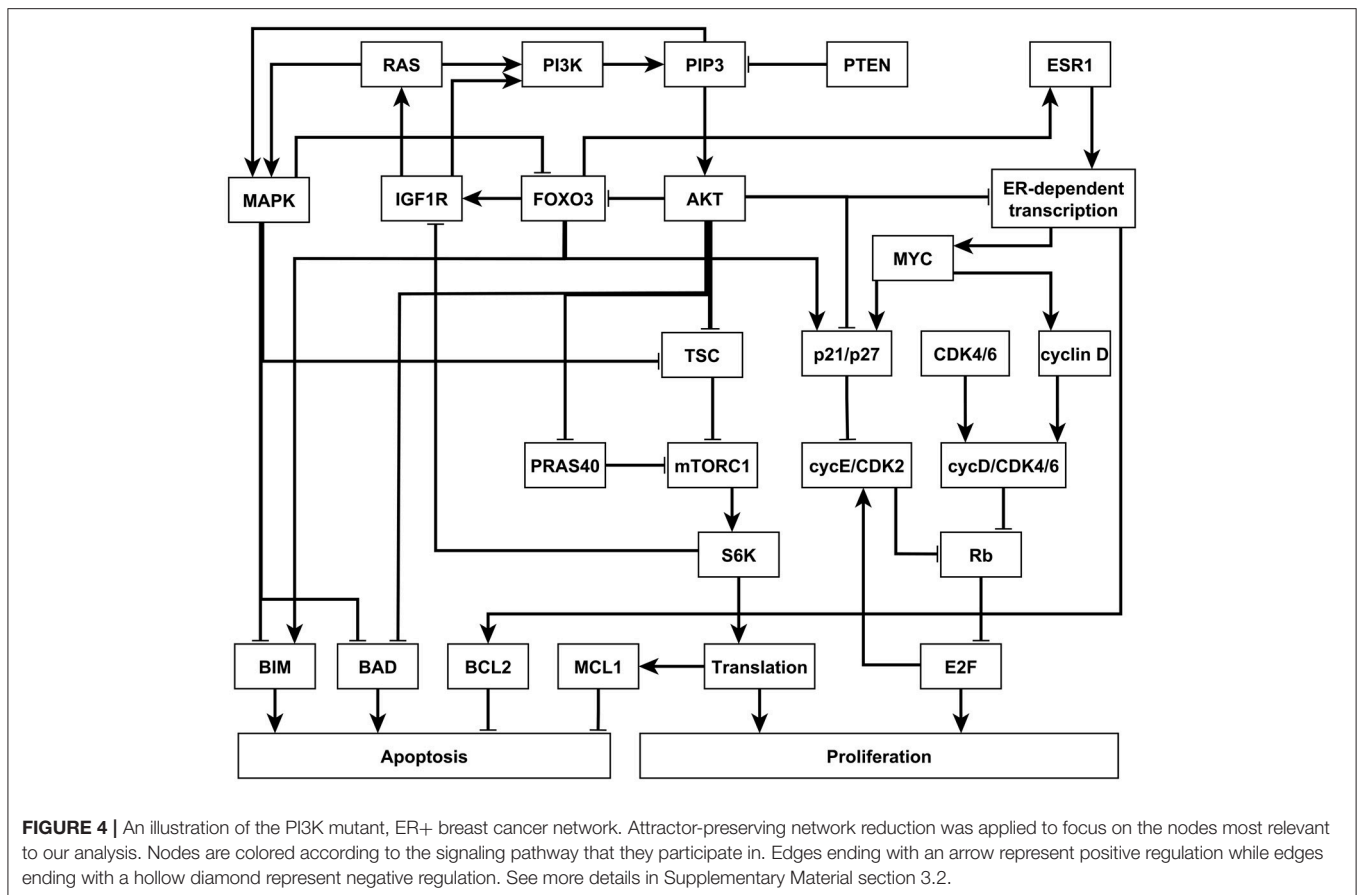
These results demonstrate both the accuracy and effectiveness of our target control algorithm. The solutions found through 1,000 iterations are comprehensive (comparable to the solution found through a systematic search of knockout pairs). Our

algorithm indicates intervention sites that are close to the target but also sites that are further away (e.g., *SMAD*). This diversity enables the selection of the most practical interventions.

3.2.2. Breast Cancer Network

In 2017, Zañudo et al. established a discrete dynamical model of the signal transduction processes involved in the PI3K mutant, estrogen receptor positive (ER+) breast cancer, as shown in **Figure 4** (Gómez Tejeda Zañudo et al., 2017). The model includes 58 nodes, which correspond to proteins, transcripts, drugs, and two cellular outcomes, apoptosis (programmed cell death) and proliferation (cell cycle progression). A fraction of the nodes (16), including the outcome nodes, are characterized by multiple levels, which is implemented by additional virtual nodes, e.g., apoptosis2 corresponds to level 2 of apoptosis, which has a more stringent regulatory function than apoptosis1 (level 1 of apoptosis). This network as implemented is essentially a Boolean network because all the regulatory functions are Boolean (Gómez Tejeda Zañudo et al., 2017). The network model successfully captures the key role of the PI3K/AKT/mTOR signaling pathway in determining the pathological proliferation and survival of cancer cells. In untreated simulated cancers cells, PI3K, MAPK, AKT, mTORC1, and ER signaling are active, leading to high level of proliferation and lack of apoptosis. The network model successfully captures the effectiveness of PI3K inhibiting drugs in leading to low level of proliferation and high level of apoptosis (Gómez Tejeda Zañudo et al., 2017). Through extensive simulations, the network model confirms known drug resistance mechanisms, i.e., additional mutations or other dysregulations that lead to the loss of effectiveness of PI3K-inhibiting drugs. It also predicts new possible resistance mechanisms and the degree of survivability under different resistance mechanisms (Gómez Tejeda Zañudo et al., 2017).

Similar insights can be drawn by LDOI analysis and applying the target control algorithm to the discrete dynamical network model without doing dynamical simulations, which demonstrate the rich information contained in the network topology and logic and the effectiveness of our control methodology. We obtained a (relatively large) reduced network by considering the system under the relevant initial condition of PI3K mutant, ER+ cancerous state, while keeping the seven drugs as source nodes (see details in Supplementary Material section 3.2). The five node states with the highest LDOI are Fulvestrant, \sim ESR1 (which both mean the inhibition of the estrogen receptor) and *Alpelisib*, \sim PI3K, \sim PIP3 (which all mean the inhibition of PI3K). The LDOI of these four nodes is very similar and includes 18 node states, including a high level of apoptosis (Apoptosis = 2), and a reduction in proliferation. Other drugs or node inhibitions yield subsets of the largest LDOI (see Supplementary Data Sheet 1). These results are consistent with, and yield further insight into the current knowledge on the effect of drugs in this network. If we now set the target to be high level of apoptosis and no proliferation, i.e., Target = {Apoptosis2, \sim Proliferation}, the algorithm gives multiple two-node interventions as minimal interventions, these consists of either of $\{\sim$ PI3K, \sim PIP3 $\}$ and inhibition of any node in the MYC-CDK4/6 axis of cell-cycle regulation, i.e., $\{\sim$ ESR1, \sim ER_transcription, \sim MYC, \sim CDK46,



\sim cyclinD, \sim cycD_CDK46, \sim Rb, \sim E2F}. There are several drugs that can target these nodes. For example, Alpelisib is a PI3K inhibitor, Fulvestrant is a ESR1 inhibitor and Palbociclib is a CDK4/6 inhibitor. This result is consistent with the results found in the (Gómez Tejada Zañudo et al., 2017): inhibition of PI3K leads to an increase in ER transcriptional regulatory activity, leading to a decrease in proliferation, and simultaneous PI3K and ER inhibition has a synergistic effect in completely blocking proliferation and maintaining a high level of apoptotic activity. If PI3K inhibitor or PIP3 inhibitor is not allowed to be used, the algorithm finds three node solutions involving an AKT inhibitor (e.g., Ipatasertib), MAPK inhibitor (e.g., Trametinib) and inhibition of any node from the MYC-CDK4/6 axis of cell-cycle regulation. In other words, inhibition of AKT together with MAPK provides a similar functionality with inhibition of PI3K. One can also use the LDOI to identify possible drug resistance mechanisms, i.e., perturbations that make PI3K inhibition less effective. As $\{\text{Apoptosis2}, \sim\text{Proliferation4}\} \subset \mathcal{LD}(\sim\text{PI3K})$, we simply go through all possible two-node interventions containing PI3K inhibitor and screen out those interventions whose LDOI either does not contain Apoptosis2 or contain Proliferation3 or higher level (Proliferation4). We reproduce most of the potential drug resistance mechanism to PI3K inhibitors indicated in Table 3 of Gómez Tejada Zañudo et al. (2017).

4. DISCUSSION

In summary, we have developed the new measures DOI and LDOI to describe the long-term effect of a sustained intervention. We have applied these measures to find solutions to the target control problem in logical network models. This work takes a step forward toward practical control of real biological systems, as illustrated by the applications presented here. The target control solutions we find recover previous predicted interventions obtained by other methods (dynamic simulations and stable motif analysis). As several of these previous predictions are validated experimentally, this agreement also serves as validation of our target control solutions. Notably, by generating a large number of valid target control solutions, we are going significantly beyond previous results (see Table 2). The multitude of predicted target control interventions allows their filtering according to biological or technological considerations.

Here we assumed the existence of a discrete dynamical model. As there are significant uncertainties in the existing models due to the scarcity of experimental information, we estimate the sensitivity of the LDOI measure to the incompleteness of the dynamical model. As the primary way of obtaining causal information that can be used in a logical model is to perform knockout experiments, the predominant causal information indicates a node as being necessary for the activation of another

node. For example, if the knockout of either of two regulators A or B leads to a decrease in the activity of target C , we would infer that the logical rule for C is $C = A \text{ AND } B$. Suppose that there is a so far undetected regulator of C , which we denote by X . This X will likely also be necessary, which would maintain agreement with the previous observations, i.e., $C = A \text{ AND } B \text{ AND } X$ is the true rule. Consider the rule for the complementary node $\sim C = \sim A \text{ OR } \sim B$ in the case of the incomplete system versus the true rule $\sim C = \sim A \text{ OR } \sim B \text{ OR } \sim X$. We can see that the LDOI of any of $\sim A, \sim B, A, B$ will be robust to the addition of X . The LDOI of node X and $\sim X$ need to be established in the true system. The LDOI of node state set $\{A, B\}$ will be affected by this change. (However, LDOI of $\sim A$ and $\sim B$ will not change). Thus the size of the solution of the target control problem may increase due to this incomplete information. Due to the binary essence of the Boolean rule, missing a sufficient regulator (an extra OR rule) will give similar results.

The LDOI is closely related to previously introduced concepts and methods used to analyze Boolean models. In particular, the LDOI of a node state (or a set of node states) is mathematically equivalent to the three-valued logical steady state that results when these node states of interest are fixed (Klamt et al., 2006; Samaga et al., 2010) and is also equivalent to the set of node states that become stationary if using network reduction techniques after fixing the node(s) of interest in the appropriate state(s) (Bilke and Sjunnesson, 2001; Naldi et al., 2012; Saadatpour et al., 2013). The work presented here goes beyond previous work and identifies general properties of the LDOI of node states and their union (the first key property of the LDOI), and of the relation of the LDOI and stable motifs (the second key property of the LDOI).

The algorithm to identify the LDOI using the expanded network bears similarities with the algorithm in Samaga et al. (2010) and Klamt et al. (2006), which uses signed interaction hypergraphs to calculate logical steady states resulting from fixing node states. An important difference is that the expanded network assigns a complementary node to each node to denote the inactive state of a node, while the hypergraph representation instead assign signs to nodes and to composite nodes to keep track of their states. Although the LDOI obtained using either method is the same, we argue that the expanded network representation has several desirable properties that differentiate it, in particular, (i) it makes the interpretation of the LDOI more intuitive and the algorithm for calculating it purely graph-theoretical, i.e., a modified breadth-first search on the expanded network, (ii) it treats the active/inactive states equally (a reflection of the fact that a change of variables can redefine what an active/inactive state means), and (iii) it provides a natural way to generalize the LDOI from Boolean to discrete models by defining a virtual node for each allowed node state (e.g., if a node has 3 states we would have 3 virtual nodes: one denoting state 0, one denoting state 1, and one denoting state 2).

The DOI and LDOI is also related the concept of elementary signaling mode (ESM), originally defined as a minimal subgraph that can propagate a signal from a source node to an output node (Wang and Albert, 2011; Sun and Albert, 2016). An ESM on the expanded network is the generalization of a path on a usual

directed network. Similarly, the LDOI of a node on the expanded network is analogous to a connected component reachable from a node on a usual directed network. In the same way a connected component reachable from node i consists of nodes that have a path starting from node i , the LDOI of a node consists of all the nodes included in any ESM that starts from that node. Recent work by Maheshwari and Albert (2017) developed a logic framework to identify causal relationships that are sufficient or necessary. This framework allows an alternative definition of the LDOI. The LDOI of the ON state of a node ($\tilde{\sigma}_i = 1$) includes all the nodes for which the node is a sufficient activator (these nodes will have $\tilde{\sigma}_j = 1$) or sufficient inhibitor (these nodes will have $\tilde{\sigma}_k = 0$). Similarly, the LDOI of the OFF state of a node includes all the nodes for which the node is a necessary activator (these nodes will have $\tilde{\sigma}_j = 0$) or necessary inhibitor (these nodes will have $\tilde{\sigma}_k = 1$).

An algorithm to construct ESMs through a backward search from an output node was presented in Wang et al. (2013); this algorithm can be adapted to find solutions of the target control problem of a single output. If we treat the output node as the root of a backward search, the set of nodes found in the ESM in each search depth (distance from the output node) can serve as a control solution. A truncation technique similar to ours needs to be applied to deal with inconsistent feed-forward or feedback loops. This algorithm can be generalized to solve the target control problem of a target set by simultaneous search from each target node. We chose to transform the target control problem into a planning search problem; and it has been established that such a planning search problem can be solved in both a forward propagation and a backward propagation approach, or even a mixed approach (Russell and Norvig, 2003). It will be an interesting future work if such techniques can improve the efficiency of the algorithm.

This work points out interesting questions as future research directions. First, though evaluating the DOI of a node (set) is computationally hard, a better estimation of the DOI rather than the LDOI is desirable and can be used to reduce the size of the solution given by our current target control algorithm. Second, the requirement that the solution works for all initial conditions in the setup of the target control problem gives robust solutions, however it may be overly conservative for biological systems in certain applications, especially if one is certain about the relevant initial condition subspace. A semi-structural approach (without doing dynamical simulations) to solve the target control problem starting from a subspace of initial conditions are also desirable.

AUTHOR CONTRIBUTIONS

GY, JG, and RA designed research and methodology. GY and RA performed the analyses. GY, JG, and RA wrote the paper.

FUNDING

This work was supported by NSF grants PHY-1205840, IIS-1161007, PHY-1545832, the Stand Up to Cancer

Foundation, and a Stand Up to Cancer Foundation/The V Foundation Convergence Scholar Award (D2015-039) to JG.

ACKNOWLEDGMENTS

This research or portions of this research were conducted with Advanced CyberInfrastructure computational resources

provided by The Institute for CyberScience at The Pennsylvania State University (<http://ics.psu.edu>).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fphys.2018.00454/full#supplementary-material>

REFERENCES

- Abou-Jaoudé, W., Monteiro, P. T., Naldi, A., Grandclaudon, M., Soumelis, V., Chaouiya, C., et al. (2015). Model checking to assess t-helper cell plasticity. *Front. Bioeng. Biotechnol.* 2:86. doi: 10.3389/fbioe.2014.00086
- Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., et al. (2016). Logical modeling and dynamical analysis of cellular networks. *Front. Genet.* 7:94. doi: 10.3389/fgene.2016.00094
- Albert, R., Acharya, B. R., Jeon, B. W., Zañudo, J. G. T., Zhu, M., Osman, K., et al. (2017). A new discrete dynamic model of ABA-induced stomatal closure predicts key feedback loops. *PLoS Biol.* 15:e2003451. doi: 10.1371/journal.pbio.2003451
- Albert, R., and Othmer, H. G. (2003). The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*. *J. Theor. Biol.* 223, 1–18. doi: 10.1016/S0022-5193(03)00035-3
- Alon, U. (2006). *An Introduction to Systems Biology: Design Principles of Biological Circuits*, 1 Edn. Chapman and Hall/CRC.
- Azuma, S., Yoshida, T., and Sugie, T. (2014). “Structural monostability of activation-inhibition boolean networks,” in *53rd IEEE Conference on Decision and Control*, 1521–1526.
- Barabási, A.-L., and Pósfai, M. (2016). *Network Science*. Cambridge: Cambridge University Press.
- Bilke, S., and Sjunnesson, F. (2001). Stability of the Kauffman model. *Phys. Rev. E* 65:016129. doi: 10.1103/PhysRevE.65.016129
- Bloomingdale, P., Nguyen, V. A., Niu, J., and Mager, D. E. (2018). Boolean network modeling in systems pharmacology. *J. Pharmacokinet. Pharmacodyn.* 45, 159–180. doi: 10.1007/s10928-017-9567-4
- Bornholdt, S. (2008). Boolean network models of cellular regulation: prospects and limitations. *J. R. Soc. Interface* 5(Suppl. 1), S85–S94. doi: 10.1098/rsif.2008.0132.focus
- Cohen, D. P. A., Martignetti, L., Robine, S., Barillot, E., Zinovyev, A., and Calzone, L. (2015). Mathematical modelling of molecular pathways enabling tumour cell invasion and migration. *PLOS Comput. Biol.* 11:e1004571. doi: 10.1371/journal.pcbi.1004571
- Cornelius, S. P., Kath, W. L., and Motter, A. E. (2013). Realistic control of network dynamics. *Nat. Commun.* 4:1942. doi: 10.1038/ncomms2939
- Espinosa-Soto, C., Padilla-Longoria, P., and Alvarez-Buylla, E. R. (2004). A gene regulatory network model for cell-fate determination during *Arabidopsis thaliana* flower development that is robust and recovers experimental gene expression profiles. *Plant Cell* 16, 2923–2939. doi: 10.1105/tpc.104.021725
- Festa, P., Pardalos, P. M., and Resende, M. G. C. (2001). Algorithm 815: Fortran subroutines for computing approximate solutions of feedback set problems using grasp. *ACM Trans. Math. Softw.* 27, 456–464. doi: 10.1145/504210.504214
- Gao, J., Liu, Y.-Y., D’Souza, R. M., and Barabási, A.-L. (2014). Target control of complex networks. *Nat. Commun.* 5:5415. doi: 10.1038/ncomms6415
- Glass, L. (1975). Classification of biological networks by their qualitative dynamics. *J. Theor. Biol.* 54, 85–107. doi: 10.1016/S0022-5193(75)80056-7
- Glass, L., and Kauffman, S. A. (1973). The logical analysis of continuous, non-linear biochemical control networks. *J. Theor. Biol.* 39, 103–129.
- Gómez Tejeda Zañudo, J., Scaltriti, M., and Albert, R. (2017). A network modeling approach to elucidate drug resistance mechanisms and predict combinatorial drug treatments in breast cancer. *Cancer Convergence* 1:5. doi: 10.1186/s41236-017-0007-6
- Grieco, L., Calzone, L., Bernard-Pierrot, I., Radvanyi, F., Kahn-Perlès, B., and Thieffry, D. (2013). Integrative modelling of the influence of mapk network on cancer cell fate decision. *PLoS Comput. Biol.* 9:e1003286. doi: 10.1371/journal.pcbi.1003286
- Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). “Exploring network structure, dynamics, and function using networkx,” in *Proceedings of the 7th Python in Science Conference*, eds G. Varoquaux, T. Vaught, and J. Millman (Pasadena, CA), 11–15.
- Hopkins, A. L., and Groom, C. R. (2002). The druggable genome. *Nat. Rev. Drug Discov.* 1, 727–730. doi: 10.1038/nrd892
- Iyengar, R., Zhao, S., Chung, S.-W., Mager, D. E., and Gallo, J. M. (2012). Merging systems biology with pharmacodynamics. *Sci. Trans. Med.* 4:126ps7. doi: 10.1126/scitranslmed.3003563
- Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* 22, 437–467.
- Kauffman, S., Peterson, C., Samuelsson, B., and Troein, C. (2003). Random boolean network models and the yeast transcriptional network. *Proc. Natl. Acad. Sci. U.S.A.* 100, 14796–14799. doi: 10.1073/pnas.2036429100
- Khan, F. M., Marquardt, S., Gupta, S. K., Knoll, S., Schmitz, U., Spitschak, A., et al. (2017). Unraveling a tumor type-specific regulatory core underlying E2F1-mediated epithelial-mesenchymal transition to predict receptor protein signatures. *Nat. Commun.* 8:198. doi: 10.1038/s41467-017-00268-2
- Kim, Y., Choi, S., Shin, D., and Cho, K.-H. (2017). Quantitative evaluation and reversion analysis of the attractor landscapes of an intracellular regulatory network for colorectal cancer. *BMC Syst. Biol.* 11:45. doi: 10.1186/s12918-017-0424-2
- Klamt, S., Saez-Rodriguez, J., Lindquist, J. A., Simeoni, L., and Gilles, E. D. (2006). A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC Bioinformatics* 7:56. doi: 10.1186/1471-2105-7-56
- Klarner, H., and Siebert, H. (2015). Approximating attractors of boolean networks by iterative CTL model checking. *Front. Bioeng. Biotechnol.* 3:130. doi: 10.3389/fbioe.2015.00130
- Klemm, K., and Bornholdt, S. (2005). Stable and unstable attractors in Boolean networks. *Phys. Rev. E* 72:055101. doi: 10.1103/PhysRevE.72.055101
- Kraeutler, M. J., Soltis, A. R., and Saucerman, J. J. (2010). Modeling cardiac β -adrenergic signaling with normalized-hill differential equations: comparison with a biochemical model. *BMC Syst. Biol.* 4:157. doi: 10.1186/1752-0509-4-157
- Laubenbacher, R., Hinkelmann, F., Murrugarra, D., and Veliz-Cuba, A. (2014). “Algebraic models and their use in systems biology,” in *Discrete and Topological Models in Molecular Biology*, Natural Computing Series (Berlin; Heidelberg: Springer), 443–474.
- Li, F., Long, T., Lu, Y., Ouyang, Q., and Tang, C. (2004). The yeast cell-cycle network is robustly designed. *Proc. Natl. Acad. Sci. U.S.A.* 101, 4781–4786. doi: 10.1073/pnas.0305937101
- Li, Y., Adeyeye, J. O., Murrugarra, D., Aguilar, B., and Laubenbacher, R. (2013). Boolean nested canalizing functions: a comprehensive analysis. *Theor. Comp. Sci.* 481, 24–36. doi: 10.1016/j.tcs.2013.02.020
- Lin, C.-T. (1974). Structural controllability. *IEEE Trans. Automatic Control* 19, 201–208.
- Liu, Y.-Y., and Barabási, A.-L. (2016). Control principles of complex systems. *Rev. Mod. Phys.* 88:035006. doi: 10.1103/RevModPhys.88.035006
- Liu, Y.-Y., Slotine, J.-J., and Barabási, A.-L. (2011). Controllability of complex networks. *Nature* 473, 167–173. doi: 10.1038/nature10011
- Maheshwari, P., and Albert, R. (2017). A framework to find the logic backbone of a biological network. *BMC Syst. Biol.* 11:122. doi: 10.1186/s12918-017-0482-5
- McCluskey, E. J. (1956). Minimization of Boolean functions. *Bell Syst. Tech. J.* 35, 1417–1444.
- Méndez-López, L. F., Davila-Velderrain, J., Domínguez-Hüttinger, E., Enríquez-Olguín, C., Martínez-García, J. C., and Alvarez-Buylla, E. R. (2017). Gene regulatory network underlying the immortalization of epithelial cells. *BMC Syst. Biol.* 11:24. doi: 10.1186/s12918-017-0393-5

- Mendoza, L. (2006). A network model for the control of the differentiation process in th cells. *Biosystems* 84, 101–114. doi: 10.1016/j.biosystems.2005.10.004
- Miskov-Zivanov, N., Turner, M. S., Kane, L. P., Morel, P. A., and Faeder, J. R. (2013). The duration of T cell stimulation is a critical determinant of cell fate and plasticity. *Sci. Signal.* 6:ra97. doi: 10.1126/scisignal.2004217
- Mochizuki, A., Fiedler, B., Kurosawa, G., and Saito, D. (2013). Dynamics and control at feedback vertex sets. II: a faithful monitor to determine the diversity of molecular activities in regulatory networks. *J. Theor. Biol.* 335, 130–146. doi: 10.1016/j.jtbi.2013.06.009
- Morris, M. K., Saez-Rodriguez, J., Sorger, P. K., and Lauffenburger, D. A. (2010). Logic-based models for the analysis of cell signaling networks. *Biochemistry* 49, 3216–3224. doi: 10.1021/bi902202q
- Motter, A. E. (2015). Network control theory. *Chaos* 25:097621. doi: 10.1063/1.4931570
- Murrugarra, D., and Dimitrova, E. S. (2015). Molecular network control through boolean canalization. *EURASIP J. Bioinformatics Syst. Biol.* 2015:9. doi: 10.1186/s13637-015-0029-2
- Murrugarra, D., Veliz-Cuba, A., Aguilar, B., and Laubenbacher, R. (2016). Identification of control targets in boolean molecular network models via computational algebra. *BMC Syst. Biol.* 10:94. doi: 10.1186/s12918-016-0332-x
- Naldi, A., Carneiro, J., Chaouiya, C., and Thieffry, D. (2010). Diversity and plasticity of th cell types predicted from regulatory network modelling. *PLoS Comput. Biol.* 6:e1000912. doi: 10.1371/journal.pcbi.1000912
- Naldi, A., Monteiro, P., and Chaouiya, C. (2012). “Efficient handling of large signalling-regulatory networks by focusing on their core control,” in *Computational Methods in Systems Biology*, Lecture Notes in Computer Science, Vol. 7605, eds D. Gilbert and M. Heiner (Berlin; Heidelberg: Springer), 288–306.
- Newman, M. E. J. (2010). *Networks: An Introduction*. Oxford, NY: Oxford University Press.
- Nicholl, D. S. T. (2008). *An Introduction to Genetic Engineering*, 3 Edn. Cambridge, UK: Cambridge University Press.
- Palsson, B. (2006). *Systems Biology: Properties of Reconstructed Networks*. Cambridge, NY: Cambridge University Press.
- Papin, J. A., Hunter, T., Palsson, B. O., and Subramaniam, S. (2005). Reconstruction of cellular signalling networks and analysis of their properties. *Nat. Rev. Mol. Cell Biol.* 6, 99–111. doi: 10.1038/nrm1570
- Pardalos, P. M., Qian, T., and Resende, M. G. (1998). A greedy randomized adaptive search procedure for the feedback vertex set problem. *J. Combinatorial Optim.* 2, 399–412.
- Paulevé, L., Magnin, M., and Roux, O. (2012). Static analysis of biological regulatory networks dynamics using abstract interpretation. *Math. Struct. Comp. Sci.* 22, 651–685. doi: 10.1017/S0960129511000739
- Paulevé, L., and Richard, A. (2012). Static analysis of boolean networks based on interaction graphs: a survey. *Electron. Notes Theor. Comput. Sci.* 284, 93–104. doi: 10.1016/j.entcs.2012.05.017
- Russell, S. J., and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall/Pearson Education.
- Saadatpour, A., Albert, I., and Albert, R. (2010). Attractor analysis of asynchronous boolean models of signal transduction networks. *J. Theor. Biol.* 266, 641–656. doi: 10.1016/j.jtbi.2010.07.022
- Saadatpour, A., and Albert, R. (2013). Boolean modeling of biological regulatory networks: A methodology tutorial. *Methods* 62, 3–12. doi: 10.1016/j.ymeth.2012.10.012
- Saadatpour, A., Albert, R., and Reluga, T. C. (2013). A reduction method for boolean network models proven to conserve attractors. *SIAM J. Appl. Dyn. Syst.* 12, 1997–2011. doi: 10.1137/13090537X
- Saez-Rodriguez, J., Simeoni, L., Lindquist, J. A., Hemenway, R., Bommhardt, U., Arndt, B., et al. (2007). A logical model provides insights into T cell receptor signaling. *PLoS Comput. Biol.* 3:e163. doi: 10.1371/journal.pcbi.0030163
- Samaga, R., Kamp, A. V., and Klamt, S. (2010). Computing combinatorial intervention strategies and failure modes in signaling networks. *J. Comput. Biol.* 17, 39–53. doi: 10.1089/cmb.2009.0121
- Shalem, O., Sanjana, N. E., Hartenian, E., Shi, X., Scott, D. A., Mikkelsen, T., et al. (2014). Genome-scale CRISPR-cas9 knockout screening in human cells. *Science* 343, 84–87. doi: 10.1126/science.1247005
- Steinway, S. N., Wang, R.-S., and Albert, R. (2016). “Discrete dynamic modeling: a network approach for systems pharmacology,” in *Systems Pharmacology and Pharmacodynamics*, eds D. Mager and H. Kimko (Cham: Springer International Publishing), 81–103.
- Steinway, S. N., Zañudo, J. G., Ding, W., Rountree, C. B., Feith, D. J., Loughran, T. P., et al. (2014). Network modeling of TGFβ signaling in Hepatocellular Carcinoma Epithelial-to-Mesenchymal Transition reveals joint sonic Hedgehog and Wnt pathway activation. *Cancer Res.* 74, 5963–5977. doi: 10.1158/0008-5472.CAN-14-0225
- Steinway, S. N., Zañudo, J. G. T., Michel, P. J., Feith, D. J., Loughran, T. P., and Albert, R. (2015). Combinatorial interventions inhibit tgfb-driven epithelial-to-mesenchymal transition and support hybrid cellular phenotypes. *Npj Syst. Biol. Appl.* 1:15014. doi: 10.1038/npijsba.2015.14
- Sun, Z., and Albert, R. (2016). Node-independent elementary signaling modes: a measure of redundancy in Boolean signaling transduction networks. *Netw. Sci.* 4, 273–292. doi: 10.1017/nws.2016.4
- Thomas, R., and D’Ari, R. (1990). *Biological Feedback*. Boca Raton, FL: CRC Press.
- Tyson, J. J., Baumann, W. T., Chen, C., Verdugo, A., Tavassoly, I., Wang, Y., Weiner, L. M., and Clarke, R. (2011). Dynamic modelling of oestrogen signalling and cell fate in breast cancer cells. *Nat. Rev. Cancer* 11:523–532. doi: 10.1038/nrc3081
- Tyson, J. J., Chen, K. C., and Novak, B. (2003). Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. *Curr. Opin. Cell Biol.* 15, 221–231. doi: 10.1016/S0955-0674(03)00017-6
- Wang, L.-Z., Su, R.-Q., Huang, Z.-G., Wang, X., Wang, W.-X., Grebogi, C., et al. (2016). A geometrical approach to control and controllability of nonlinear dynamical networks. *Nat. Commun.* 7:11323. doi: 10.1038/ncomms11323
- Wang, R.-S., and Albert, R. (2011). Elementary signaling modes predict the essentiality of signal transduction network components. *BMC Syst. Biol.* 5:44. doi: 10.1186/1752-0509-5-44
- Wang, R.-S., Saadatpour, A., and Albert, R. (2012). Boolean modeling in systems biology: an overview of methodology and applications. *Phys. Biol.* 9:055001. doi: 10.1088/1478-3975/9/5/055001
- Wang, R.-S., Sun, Z., and Albert, R. (2013). Minimal functional routes in directed graphs with dependent edges. *Int. Trans. Oper. Res.* 20, 391–409. doi: 10.1111/itor.12007
- Wells, D. K., Kath, W. L., and Motter, A. E. (2015). Control of stochastic and induced switching in biophysical networks. *Phys. Rev. X* 5:031036. doi: 10.1103/PhysRevX.5.031036
- Wynn, M. L., Consul, N., Merajver, S. D., and Schnell, S. (2012). Logic-based models in systems biology: a predictive and parameter-free network analysis method. *Integr. Biol.* 4, 1323–1337. doi: 10.1039/c2ib20193c
- Yang, G., Campbell, C., and Albert, R. (2016). Compensatory interactions to stabilize multiple steady states or mitigate the effects of multiple deregulations in biological networks. *Phys. Rev. E* 94:062316. doi: 10.1103/PhysRevE.94.062316
- Zañudo, J. G. T., and Albert, R. (2013). An effective network reduction approach to find the dynamical repertoire of discrete dynamic networks. *Chaos* 23:025111. doi: 10.1063/1.4809777
- Zañudo, J. G. T., and Albert, R. (2015). Cell fate reprogramming by control of intracellular network dynamics. *PLoS Comput. Biol.* 11:e1004193. doi: 10.1371/journal.pcbi.1004193
- Zañudo, J. G., Steinway, S. N., and Albert, R. (2018). Discrete dynamic network modeling of oncogenic signaling: mechanistic insights for personalized treatment of cancer. *Current Opinion in Systems Biology*, 9, 1–10.
- Zañudo, J. G. T., Yang, G., and Albert, R. (2017). Structure-based control of complex networks with nonlinear dynamics. *Proc. Natl. Acad. Sci. U.S.A.* 114, 7234–7239. doi: 10.1073/pnas.1617387114
- Zertuche, F. (2009). On the robustness of NK-kauffman networks against changes in their connections and boolean functions. *J. Math. Phys.* 50:043513. doi: 10.1063/1.3116166

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2018 Yang, Gómez Tejada Zañudo and Albert. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.