CANA: A Python Package for Quantifying Control and Canalization in Boolean Networks

Rion B. Correia [1,2,3], Alexander J. Gates [4], Xuan Wang [1] and Luis M. Rocha [1,3]*

[1] School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, United States, [2] CAPES Foundation, Ministry of Education of Brazil, Brasília, Brazil, [3] Instituto Gulbenkian de Ciência, Oeiras, Portugal, [4] Center for Complex Networks Research, Northeastern University, Boston, MA, United States

Logical models offer a simple but powerful means to understand the complex dynamics of biochemical regulation, without the need to estimate kinetic parameters. However, even simple automata components can lead to collective dynamics that are computationally intractable when aggregated into networks. In previous work we demonstrated that automata network models of biochemical regulation are highly canalizing, whereby many variable states and their groupings are redundant (Marques-Pita and Rocha, 2013). The precise charting and measurement of such canalization simplifies these models, making even very large networks amenable to analysis. Moreover, canalization plays an important role in the control, robustness, modularity and criticality of Boolean network dynamics, especially those used to model biochemical regulation (Gates and Rocha, 2016; Gates et al., 2016; Manicka, 2017). Here we describe a new publicly-available Python package that provides the necessary tools to extract, measure, and visualize canalizing redundancy present in Boolean network models. It extracts the pathways most effective in controlling dynamics in these models, including their *effective graph* and *dynamics canalizing map*, as well as other tools to uncover minimum sets of control variables.

Keywords: Boolean networks, automata, canalization, python package, biochemical regulation, logical modeling, network dynamics, complex systems

1. A TOOL TO STUDY REDUNDANCY AND CONTROL IN BOOLEAN NETWORKS

Mathematical and computational modeling of biological networks promises to uncover the fundamental principles of living systems in an integrative manner (Iyengar, 2009; Ideker and Nussinov, 2017). In particular, Boolean Networks (BN), a class of logical dynamical systems, provide an effective framework to capture the dynamics of interconnected biological systems without the need for detailed kinetic parameters (Bornholdt, 2005; Assmann and Albert, 2009). BN have been used to model and predict biochemical regulation in genetic networks (Li et al., 2004), cell signaling (Helikar et al., 2008), chemical reactions in metabolic networks (Chechik et al., 2008), anticancer drug response (Choi et al., 2017), action potentials in neural networks (Kurten, 1988), and many other dynamical systems involved in biomedical complexity (Albert and Othmer, 2003).

Two reasons contribute to the success of BN models: (i) the reduction of complex multivariate dynamics to a graph revealing the organization and constraints of the topology of interactions in biological systems, and (ii) a coarse-grained treatment of dynamics

that facilitates predictions of limiting behavior and robustness (Bornholdt, 2008). However, more than understanding the organization of complex biological systems, we need to derive control strategies that allow us, for example, to intervene on a diseased cell (Zhang et al., 2008), or revert a mature cell to a pluripotent state (Wang and Albert, 2011). Recently, several mathematical tools were developed to enhance our understanding of BN control by removing redundant pathways, identifying key dynamic modules (Marques-Pita and Rocha, 2013), and characterizing critical driver variables (Gates and Rocha, 2016).

Here we present CANA[1], a python package to study redundancy and control in BN models of biochemical dynamics (Correia et al., 2018). It provides a simple interface to access computational tools for three important aspects of BN analysis and prediction:

1. **Dynamics.** Python classes are included to enumerate all *attractors* and calculate the full *state transition graph* (STG) of BN, as described in section 2.
2. **Canalization.** The redundancy properties of automata functions have been characterized as a form of canalization (Kauffman, 1984), particularly when used to model dynamical interactions in models of genetic regulation and biochemical signaling (Kauffman et al., 2004; Reichhardt and Bassler, 2007; Marques-Pita and Rocha, 2013). At the level of individual Boolean transition functions (network nodes), canalization is observed when not all inputs are necessary to determine a state transition (see section 3 for formal definition). CANA can be used to calculate all measures of canalization that derive from removing dynamical redundancy via two-symbol schemata re-description (Marques-Pita and Rocha, 2013): *effective connectivity*, *input redundancy*, and *input symmetry*. At the network level, CANA also calculates the *effective graph*, a weighted and directed graph whose edge weights denote their effective contribution to node transitions, as well as the *dynamics canalizing map*, a parsimonious representation of the necessary and sufficient state transitions that define the entire dynamics of BN. All canalization measures and network representations are applicable to synchronous and asynchronous BN models, as described in section 3.
3. **Control.** From a subset of driver variables—nodes that act as the loci of control interventions—CANA computes the *controlled state transition graph* (CSTG), as well as the *controlled attractor graph* (CAG) capturing all controlled transitions between attractors possible via driver variable interventions (Gates and Rocha, 2016). CANA also computes measures of controllability that depend on the CSTG and CAG: *mean fraction of reachable configurations*, *mean fraction of controlled configurations*, and *mean fraction of reachable attractors*, as described in section 4. Currently, control analysis in CANA is applicable only to synchronous BN models.

Here we demonstrate the full functionality of the CANA package using the BN model of floral organ development in the flowering

---

[1]CANAlization: Redundancy & Control in Boolean Networks. For documentation and tutorials (see available online at: github.com/rionbr/CANA)

plant *Arabidopsis thaliana* (Chaos et al., 2006). Additionally, we provide an interface between CANA and the *Cell Collective* (Helikar et al., 2012), allowing for an extensive analysis of control and canalization in complex biological systems.

The CANA package fills a key void in the available library of computational software to analyze Boolean Network models. Existing software falls into two categories: either they are designed to reverse engineer BN models from biological experimental data, or they focus on simulating BN dynamics. Examples of the first category include the *CellNetOptimizer* which creates BN from high-throughput biochemical data (Terfve et al., 2012), and the *Dynamic Deterministic Effects Propagation Networks* (DDEPN) package which reconstructs signaling networks based from time-course experimental data (Bender et al., 2010). The second category of BN simulation packages is best exemplified by *BooleanNet*, a python package that simulates both synchronous and asynchronous dynamics (Albert et al., 2008), and PANET, a Cytoscape plugin that quantifies the robustness of BN models (Trinh et al., 2014). *The Cell Collective*, a collaborative platform and intuitive visual interface to share and build BN models, can also be used to simulate BN dynamics (Helikar et al., 2012). The CANA package expands the set of available tools of the second category, by providing Python classes to calculate measures and visualizations of canalization (dynamical redundancy) and control of BN models, as detailed below. CANA is designed as a toolbox for both computational and experimental system biologists. It enables the simplification of BN models and testing of network control algorithms, thus prioritizing biochemical variables more likely to be relevant for specific biological questions (e.g., genes controlling cell fate), and ideal candidates for knockout experiments.

2. BOOLEAN NETWORK REPRESENTATION AND DYNAMICS

A *Boolean automaton* is a binary variable, $x \in \{0, 1\}$, whose state is updated in discrete time-steps, t, according to a deterministic *Boolean state-transition function* of k inputs: $x^{t+1} = f(x_1^t, ..., x_k^t)$. The state-transition function, $f : \{0, 1\}^k \rightarrow \{0, 1\}$, is defined by a *look-up (truth) table* (LUT), $F \equiv \{f_\alpha : \alpha = 1, ..., 2^k\}$, with one entry for each of the 2^k combinations of input states and a mapping to the automaton's next state (transition or output), x^{t+1} (**Figure 1A**). In CANA, a Boolean automaton—a python class denoted *BooleanNode*—is instantiated from the list of transitions that define its LUT.

A *Boolean Network* is a graph $\mathcal{B} \equiv (X, C)$, where X is a set of N Boolean automata *nodes* $x_i \in X, i = 1, ..., N$ and C is a set of directed edges $c_{ji} \in C : x_i, x_j \in X$ that represent the interaction network, denoting that automaton x_j is an input to automaton x_i, as computed by F_i. The set of inputs for automaton x_i is denoted by $X_i = \{x_j \in X : c_{ji} \in C\}$, and its cardinality, $k_i = |X_i|$, is the *in-degree* of node x_i. At any given time t, $\mathcal{B}$ is in a specific configuration of automata states, $\boldsymbol{x}^t = \langle x_1^t, x_2^t, ..., x_N^t \rangle$, where we use the terms *state* for individual automata (x_i^t) and *configuration* ($\boldsymbol{x}^t$) for the collection of states of all automata of
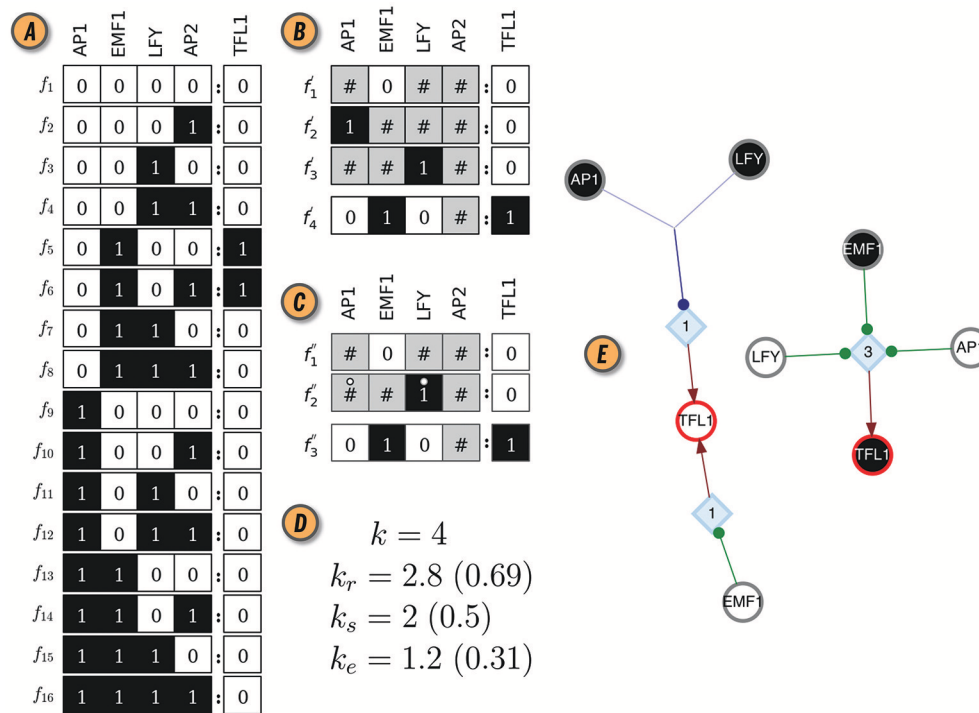
FIGURE 1 | CANA analysis of the Boolean automaton defining the dynamics of the TFL1 gene in the BN model of the floral organ arrangement in the flowering plant *Arabidopsis Thaliana*. **(A)** Look-up-table (LUT). **(B)** Wildcard schema redescription, F'(TFL1). Wildcards are denoted by gray states. As an example, schema f'_4 redescribes the subset of LUT entries $\Upsilon_4 \equiv \{f_5, f_6\}$, where the input variable AP2 can be either *on* or *off*. **(C)** Two-symbol schema redescription, F''(TFL1). Permutation of the inputs marked with the position-free symbol (○) in any schema of F''(TFL1) result in a wildcard schema in F'(TFL1). For example, f''_2 redescribes $\Theta'_2 \equiv \{f'_2, f'_3\}$. **(D)** In-degree ($k$), input redundancy ($k_r$), input symmetry ($k_s$), and effective connectivity (k_e) of TFL1 automaton. Values in parenthesis are the respective (relative) measures normalized by k, used for comparisons between automata with different number of inputs. **(E)** Canalizing Map (CM) of automaton TFL1, with its two possible states, TFL1 $\in \{0, 1\}$, shown as circles with red contour; white (black) fill color denotes state 0 (1). Input variables and their respective state are also shown as circles (*s-units*) with the same color criterion, and link to *t-units* shown as blue diamonds with corresponding threshold value inside; thus, TFL1 requires 3 input conditions (LFY = 0 ∧ EMF1 = 1 ∧ AP1 = 0) to turn on (TLF1 = 1), but only one (EMF1 = 0 ∨ AP1 = 1 ∨ LFY = 1) to turn off (TLF1 = 0); ∧ and ∨ denote the logical conjunction (and) and disjunction (or), respectively. Network rendering generated with Graphviz (Ellson et al., 2002).

the BN at time t, i.e. the collective network state. The set of all possible network configurations is denoted by $\mathcal{X} \equiv \{0, 1\}^N$, where $|\mathcal{X}| = 2^N$. The dynamics of $\mathcal{B}$ unfolds from an initial configuration, $\boldsymbol{x}^0$, by a *synchronous*, update policy in which all automata transition to the next state at the same time step, or an *asynchronous* update policy, in which automata update their next step in distinct time steps according to some update schedule (e.g. stochastically). The complete dynamical behavior of the system for all initial conditions is captured by the *state-transition graph* (STG), $\mathcal{G} \equiv STG(\mathcal{B}) = (\mathcal{X}, \mathcal{T})$, where each node is a configuration $\boldsymbol{x}_\alpha \in \mathcal{X}$, and an edge $T_{\alpha,\beta} \in \mathcal{T}$ denotes that a BN in configuration $\boldsymbol{x}_\alpha$ at time t will be in configuration $\boldsymbol{x}_\beta$ at time $t+1$. Under deterministic dynamics, only a single transition edge $T_{\alpha,\beta}$ is allowed out of every configuration node $\boldsymbol{x}_\alpha$. Configurations that repeat, such that $\boldsymbol{x}_\alpha^{t+\mu} = \boldsymbol{x}_\beta^t$, are known as *attractors* and differentiated as *fixed-point* attractors when $\mu = 1$, and *limit cycles* when $\mu > 1$, respectively. Because $\mathcal{G}$ is finite, it contains at least one attractor, as some configuration or limit cycle must repeat in time (Wuensche, 1998).

In CANA, a python class named *BooleanNetwork* represents a BN, and is instantiated from a dictionary containing the transition functions (LUT) of all its constituent automata nodes,

or loaded from a file. We also provide several predefined example BN models that can be directly loaded: the *Arabidopsis Thaliana* gene regulatory network (GRN) of flowering patterns (Chaos et al., 2006), a simplified version of the segment polarity GRN of *Drosophila melanogaster* (Albert and Othmer, 2003), the *Budding Yeast* cell-cycle regulatory network (Li et al., 2004), and the BN motifs analyzed in Gates and Rocha (2016). Beyond the aforementioned networks, our current release also incorporates all publicly available networks in the Cell Collective repository (Helikar et al., 2012). These were loaded from the Cell Collective API and converted into truth tables that can be read by CANA[2]. Our package has two built-in methods available to compute network dynamics: for relatively small BN ($N < 30$) the full state-space can be computed, whereas for larger BN, CANA uses a Boolean satisfiability (SAT-based) algorithm, capable of enumerating all attractors in a BN with thousands of variables (Dubrova and Teslenko, 2011).

---

[2]Future releases will provide a direct link to the Cell Collective API for conversion of Cell Collective models. Currently, models are converted to .CNET (truth table) format, and subsequently imported to CANA.

3. CANALIZATION

Important insights about BN dynamics are gained by observing that not all inputs to an automaton are equally important for determining its state transitions, a concept known as *canalization* (Reichhardt and Bassler, 2007). Originally, the term was proposed by Waddington (1942) and subsequently refined to characterize the buffering of genetic and epigenetic perturbations leading to the stability of phenotypic traits (Siegal and Bergman, 2002; Masel and Maughan, 2007; ten Tusscher and Hogeweg, 2009). Understanding how canalization occurs in a given BN model allows us to uncover and remove redundancy present in the pathways that control its dynamics. In CANA, we follow Marques-Pita and Rocha (2013) by quantifying canalization through the logical *redundancy* present in automata. Specifically, we use the Quine-McCluskey Boolean minimization algorithm (Quine, 1955) to identify those inputs of an automaton which are redundant given the state of its other inputs, thus reducing its LUT to a set of *prime implicants*. The prime implicants are in turn combined to create wildcard schemata, $F' \equiv \{f'_\upsilon\}$, in which the *wildcard* or "Don't care" symbol, # (also represented graphically in gray) denotes an input whose state is redundant given the state of other necessary input states. In this process, the original LUT F (**Figure 1A**) is redescribed by a more compressed set of schemata F' (**Figure 1B**). Every wildcard schema $f'_\upsilon \in F'$ redescribes a subset of entries in the original LUT, denoted by $\Upsilon_\upsilon \equiv \{f_\alpha : f_\alpha \longmapsto f'_\upsilon\} \subseteq F$; $\longmapsto$ means 'is redescribed by'. Finally, CANA also calculates the *two-symbol schemata* redescription, $F'' \equiv \{f''_\theta\}$, whereby in addition to the wildcard symbol, a *position-free* symbol, ∘, further captures *permutation redundancy* (i.e., group-symmetry): subsets of inputs whose states can permute without affecting the automaton's state (**Figure 1C**). Every two-symbol schema $f''_\theta \in F''$ redescribes a set $\Theta_\theta \equiv \{f_\alpha : f_\alpha \longmapsto f''_\theta\} \subseteq F$ of LUT entries of automaton x.

Several measures of canalization present in the LUT of an automaton are also defined in CANA, and can be accessed by function calls to both the *BooleanNode* and *BooleanNetwork* classes. *Input redundancy*, $k_r(x)$, measures the number of inputs that on average are not needed to compute the state of automaton x. This is measured by tallying the mean number of wildcard symbols present in the set of schemata $F'(x)$ or $F''(x)$ that redescribe the LUT $F(x)$ (Equation 1). *Effective connectivity*, k_e, is a complementary measure of $k_r(x)$ yielding the number of inputs that are on average necessary to compute the automaton's state (Equation 1). Whereas $k(x)$ is the number of inputs to automaton x present in the BN, $k_e(x)$ is the minimum number of such inputs that are on average necessary to determine the state of x—its effective connectivity or degree. Similarly, *input symmetry*, $k_s(x)$, is the mean number of inputs that can permute without effect on the state of x. It is measured by tallying the mean number of position-free symbols present in $F''(x)$ (Equation 1):

$$k_r(x) = \frac{\sum_{f_\alpha \in F} \max_{\upsilon : f_\alpha \in \Upsilon_\upsilon} (n^\#_\upsilon)}{|F|} \quad , \quad k_e(x) = k(x) - k_r(x) \quad ,$$

$$k_s(x) = \frac{\sum_{f_\alpha \in F} \max_{\theta : f_\alpha \in \Theta_\theta} (n^\circ_\theta)}{|F|} \tag{1}$$

where $n^\#_\upsilon$ and n°_θ are the number of inputs with a # or ∘ in schema f'_υ or f''_θ, respectively[3]. **Figure 1D** shows the values of these measures for the LUT of the TFL1 gene in the *thaliana* GRN model. Additional algorithmic details of the two forms of canalization, as well as their importance to study control, robustness, and modularity of BN models of biochemical regulation, are presented in Marques-Pita and Rocha (2013). Next we introduce new per-input measures of canalization as well as the effective graph, which CANA also computes.

Most automata contain redundancy of one or both of the two forms of canalization; only the two parity functions for any k have $k_r = 0$ (e.g., the *XOR* function and its negation for $k = 2$), and even those can have $k_s > 0$. Therefore, the original interaction graph of a BN tends to have much redundancy and does not capture how automata truly influence one another in a BN. To formalize this idea, the CANA package computes an *effective graph*, $\mathcal{E} \equiv (X, E)$, where X is as in section 2 and E is a set of weighted directed edges $e_{ji} \in [0, 1] \forall x_i, x_j \in X$ denoting the *effectiveness* of automaton x_j in determining the truth value of automaton x_i, and computed via Equation 2. Specifically, we define per-input measures of canalization for *redundancy*, *effectiveness*, and *symmetry*, respectively:

$$r_{ji} = \frac{\sum_{f_\alpha \in F_i} \operatorname*{avg}_{\upsilon : f_\alpha \in \Upsilon^i_\upsilon} (j \longmapsto \#)_\upsilon}{|F_i|} \quad , \quad e_{ji} = 1 - r_{ji} \quad ,$$

$$s_{ji} = \frac{\sum_{f_\alpha \in F_i} \operatorname*{avg}_{\theta : f_\alpha \in \Theta^i_\theta} (j \longmapsto \circ)_\theta}{|F_i|} \tag{2}$$

where $(j \longmapsto \#)_\upsilon$ is a logical condition that assumes the truth value 1(0) if input j is (not) a wildcard in schema f'_υ, and similarly for $(j \longmapsto \circ)_\theta$ for a position-free symbol in schema f''_θ; avg is the average operator. Naturally, $k_r(x_i) = \sum_j r_{ji}$, $k_e(x_i) = \sum_j e_{ji}$, and $k_s(x_i) = \sum_j s_{ji}$.

The effective graph was shown to be important in predicting the controllability of BN (Gates and Rocha, 2016). Furthermore, the mean k_e of BN (the mean in-degree of the effective graph) is a better predictor of criticality than the in-degree of the original interaction graph (Manicka, 2017), improving the existing theory for predicting criticality in BN (Aldana, 2003). Those results suggest that Natural Selection can select for canalization, thereby enhancing the stability and controllability of networks with high connectivity, that would otherwise exist in the chaotic regime (Gates et al., 2016; Manicka, 2017). As an example, the interaction and effective graphs of the *Thaliana* GRN BN model, as computed by CANA, are shown in **Figures 2A,B**, demonstrating that much redundancy exists in the original model. The most extreme case of redundancy occurs when an input from x_j to automaton x_i exists in the original interaction graph C, $c_{ji} = 1$, but not in the effective graph $\mathcal{E}$, $e_{ji} = 0$, because

---

[3]k_r and k_e can be computed on either set of schemata F' (as in Equation 1) or F'' (as in Marques-Pita and Rocha 2013), yielding the same result; k_s must be computed on F''.
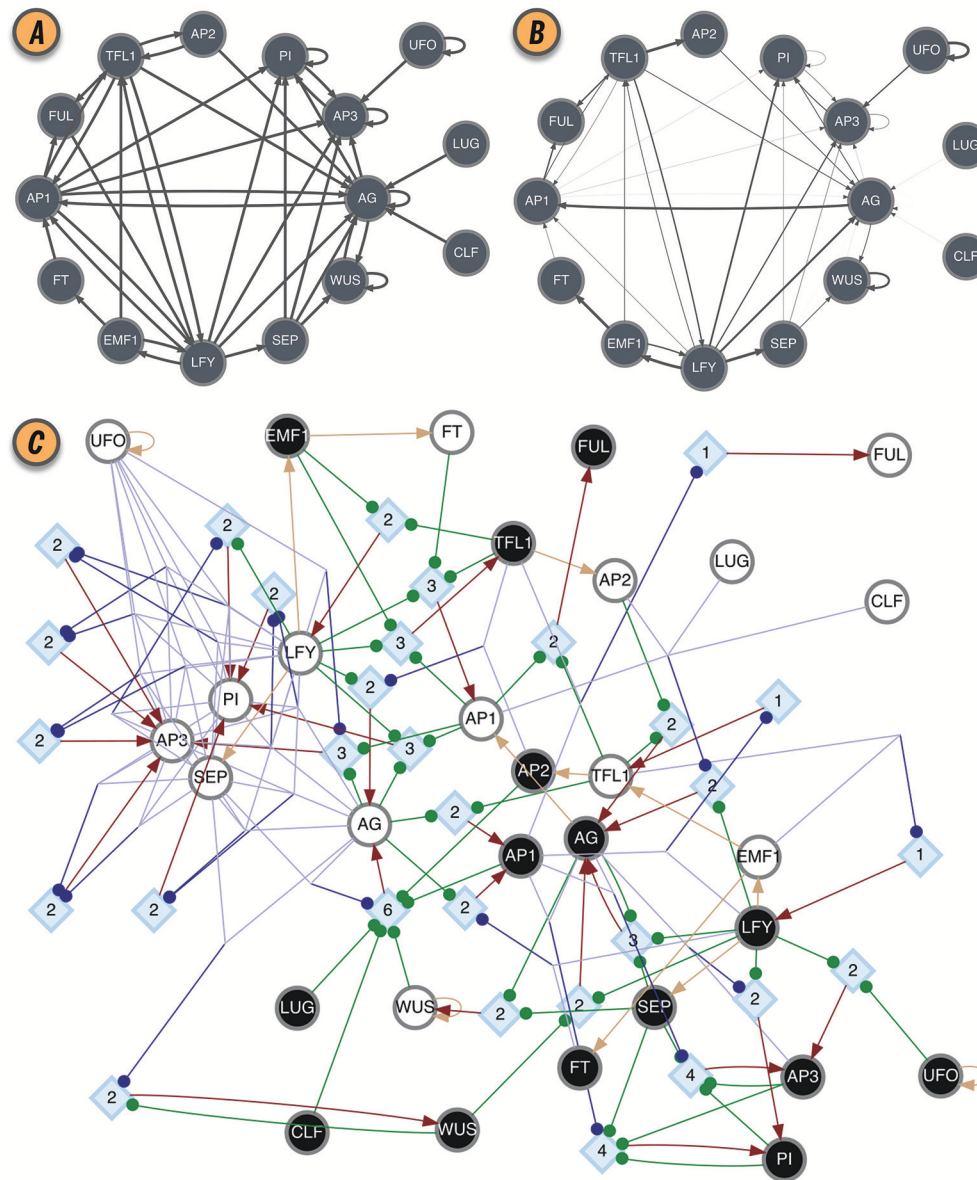
FIGURE 2 | BN model of the floral organ arrangement in the flowering plant *Arabidopsis Thaliana*. **(A)** Interaction graph *C*. **(B)** Effective graph *E*, where edge weights denote e_{ji} (Equation 2). Some edges, originally in *C*, are completely removed in *E* (e.g., AG→AG, AP1→AG, and AP2→TFL1). Others, have very small effectiveness (e.g., AP1→PI and CLF→AG). **(C)** Dynamics Canalization Map (DCM) representing the entire logic of interactions after removal of redundancy. Original BN automata nodes appear twice in the DCM, once for each Boolean truth value and denoted as *s-unit*, white (0) or black (1) circles. When *s-units* are determined by another single *s-unit*, for simplicity and without loss of generality, they are connected with a beige directed edge—a simplification to avoid the rendering of a *t-unit* with a threshold of one. All other variable state determinations occur via *t-units* with larger threshold values. Red edges represent outputs from *t-units* to *s-units*: a state determination of the receiving *s-unit*, after the logical condition of the *t-unit* is met. All other (blue or green) edges denote inputs from *s-units* to *t-units*, that is, the sufficient conditions for a state determination. Blue edges denote group disjunction constraints, whereby conditions captured by *s-units* can merge because any one of the merging conditions is sufficient [e.g., (TFL = 0 ∨ EMF1 = 0) → LFY = 1]. Green edges denote independent and necessary conditions. Directed edges into *s-units* are denoted by arrows, while directed edges into *t-units* are denoted by small circles. Network rendering by Graphviz (Ellson et al., 2002).

it is fully redundant and does not affect the automaton's transition (see several such cases in **Figures 2A,B**).

The canalizing logic of an automaton provided by the schemata set F'', can also be represented as a McCulloch and Pitts (1943) threshold network, named a *Canalizing Map* (CM) in Marques-Pita and Rocha (2013). **Figure 1E** depicts the CM

for the TFL1 gene. It consists of two types of nodes: *state units* (*s-unit*, denoted by circles), which represent automata in one of the Boolean truth values ($x_i = 0$, white, or $x_i = 1$, black), and *threshold units* (*t-unit*, denoted by diamonds), which implement a numerical threshold condition on its inputs. When the CM of all automata of a BN are linked, we obtain the
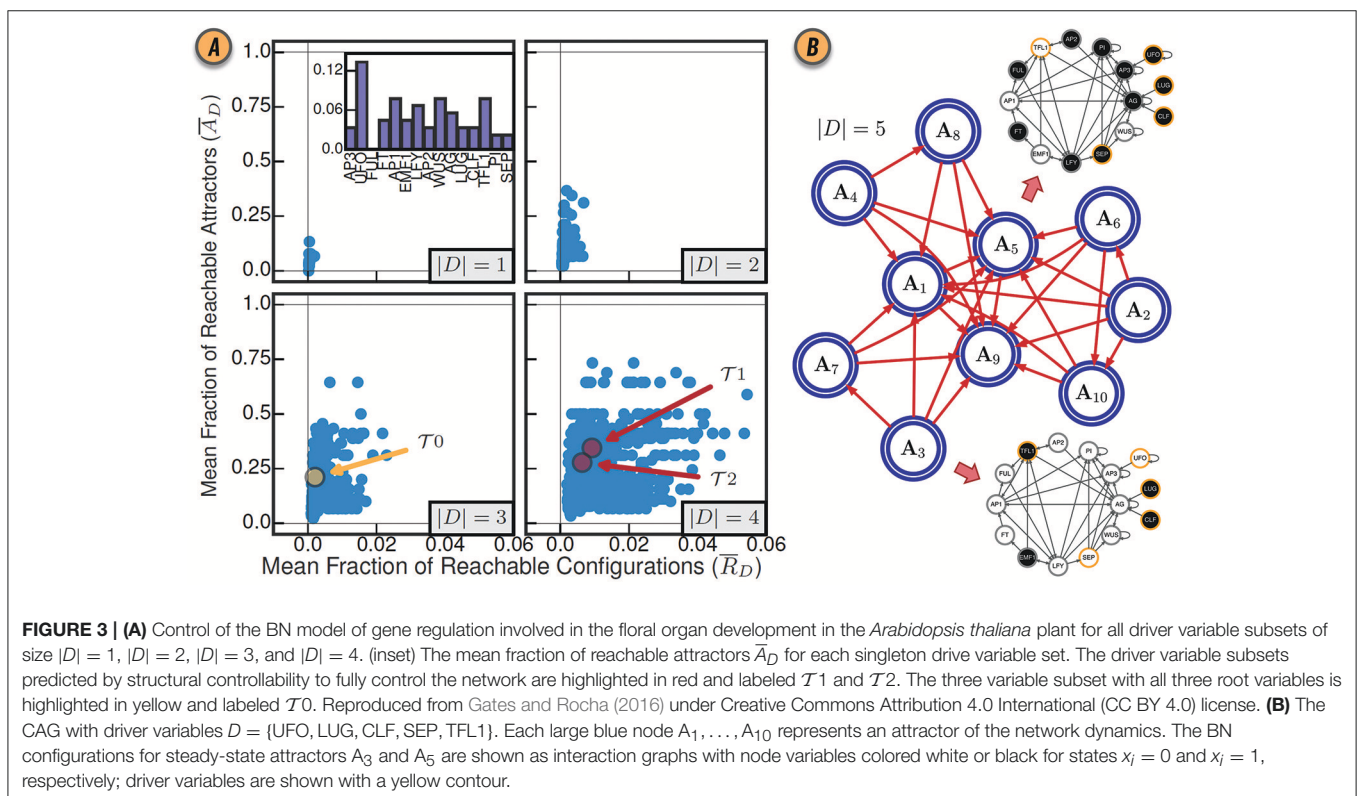
Dynamics Canalization Map (DCM), as shown in **Figure 2C** for the *Thaliana* GRN. Directed fibers connect nodes and propagate an activation pulse; fibers can merge and split, but each end-point always contributes one pulse to an s-unit. The DCM is a highly parsimonious representation of the dynamics of a BN. It contains only necessary information about how (canalizing) control signals determine network dynamics. It enables inferences about control, modularity and robustness to be made about the collective (macro-level) dynamics of BN (Marques-Pita and Rocha, 2013). Because it is assembled using solely the micro-level canalizing logic of individual automata, its computation scales linearly with the number of nodes of the network, and thus it can be computed for very large networks. The computational bottleneck can only be the number of inputs (k) to a particular automaton, since the Quine–McCluskey algorithm grows exponentially with the number of variables. Functions with a large number of variables have to be minimized with heuristic methods such as Espresso (Brayton et al., 1984). Because all measures of canalization, as well as the effective graph and the DCM, derive from removing dynamical redundancy at the level of individual automata, they are independent from the updating regime chosen for the network. In other words, the canalization analysis is applicable to synchronous and asynchronous BN models.

4. CONTROL

The discovery of control strategies in BN models is a central problem in systems biology; theoretical insights about controllability can enhance experimental turnover by focusing experimental interventions on genes and proteins more likely to result in the desired phenotype output. It is well-known that when the set of automata nodes X of a BN is large, enumeration of all configurations $\boldsymbol{x} \in \mathcal{X}$ of its STG becomes difficult, making the controllability of deterministic BN an NP-hard problem (Akutsu et al., 2007). Thus control methodologies which leverage the interaction graph or remove the redundancy in canalizing automata are highly desirable, since they can greatly simplify BN complexity.

CANA contains Python functions designed to provide a testbed for the development of BN control strategies, and to investigate the interplay between canalization, control, and other dynamics properties. Specifically, we study the control exerted on the dynamics of a BN, $\mathcal{B} = (X, C)$, by a subset of *driver variables* $D \subseteq X$—a subset of automata nodes of $\mathcal{B}$. Control *interventions* are realized by instantaneous bit-flip perturbations to the state of the variables in D (Willadsen and Wiles, 2007). This results in a *controlled state transition graph*, $CSTG(\mathcal{B}) \equiv \mathcal{G}_D \equiv (\mathcal{X}, \mathcal{T} \cup \mathcal{T}_D)$, which is an extension of the STG that captures all possible trajectories due to controlled interventions on D (Gates and Rocha, 2016). The additional edges $\mathcal{T}_D$ denote transitions from every configuration to a set of $2^{|D|} - 1$ configurations in the STG, which are reachable given the bit-flip perturbations of the driver variables. A BN is controllable when every configuration is reachable from every other configuration in $\mathcal{G}_D$ (Sontag, 1998), a condition equivalent to requiring that the CSTG $\mathcal{G}_D$ be strongly connected.

FIGURE 3 | (A) Control of the BN model of gene regulation involved in the floral organ development in the *Arabidopsis thaliana* plant for all driver variable subsets of size $|D| = 1$, $|D| = 2$, $|D| = 3$, and $|D| = 4$. (inset) The mean fraction of reachable attractors $\overline{A}_D$ for each singleton drive variable set. The driver variable subsets predicted by structural controllability to fully control the network are highlighted in red and labeled $\mathcal{T}1$ and $\mathcal{T}2$. The three variable subset with all three root variables is highlighted in yellow and labeled $\mathcal{T}0$. Reproduced from Gates and Rocha (2016) under Creative Commons Attribution 4.0 International (CC BY 4.0) license. **(B)** The CAG with driver variables $D = \{UFO, LUG, CLF, SEP, TFL1\}$. Each large blue node $A_1, \ldots, A_{10}$ represents an attractor of the network dynamics. The BN configurations for steady-state attractors A_3 and A_5 are shown as interaction graphs with node variables colored white or black for states $x_i = 0$ and $x_i = 1$, respectively; driver variables are shown with a yellow contour.

CANA computes the CSTG of $\mathcal{B}$ given a driver set D, which in turn is used to calculate the *mean fraction of reachable configurations*, $\overline{R}_D$, and the *mean fraction of controlled configurations*, $\overline{C}_D$, (Gates and Rocha, 2016):

$$\overline{R}_D = \frac{1}{2^N} \sum_{x_\alpha \in \mathcal{X}} r(\mathcal{G}_D, x_\alpha) \quad , \quad \overline{C}_D = \overline{R}_D - \overline{R}_\emptyset \quad . \quad (3)$$

where, for each configuration x_α, $r(\mathcal{G}_D, x_\alpha)$ is the *fraction of reachable configurations*, defined as the number of other configurations X_β lying on all directed paths from x_α, normalized by the total number of other configurations 2^{N-1}. Similarly, the *fraction of controlled configurations* counts the number of new configurations that are reachable due to interventions to D, but were not originally reachable in the STG: $c(\mathcal{G}_D, x_\alpha) = r(\mathcal{G}_D, x_\alpha) - r(\mathcal{G}, x_\alpha)$. When a BN is fully controlled by D, $\overline{R}_D = 1.0$, but for partially controlled BNs $\overline{R}_D \in [0.0, 1.0]$; note that $\overline{C}_D \leq \overline{R}_D$.

In Systems Biology applications, typically only the attractors of BN are meaningful configurations, used to represent different cell types (Kauffman, 1969, 1993; Müller and Schuppert, 2011), diseased or normal conditions (Zhang et al., 2008), and wild-type or mutant phenotypes (Albert and Othmer, 2003). In this context, a more relevant control measure is the extent to which driver variables can steer dynamics from attractor to attractor. To quantify such control, CANA computes the *controlled attractor graph* (CAG) of a BN $\mathcal{B} : \mathcal{C}_D = (\mathcal{A}, \mathcal{Z}_D)$. The nodes of this graph, $\mathbf{A}_\kappa \in \mathcal{A}$, represent an attractor of $\mathcal{B}$, and each edge $z_{\kappa\gamma} \in \mathcal{Z}_D$, denotes the existence of at least one path from attractor $\mathbf{A}_\kappa$ to attractor $\mathbf{A}_\gamma$ in the CSTG $\mathcal{G}_D$ (**Figure 3B**). The *mean fraction of reachable attractors* is then given by

$$\overline{A}_D = \frac{1}{|\mathcal{A}|} \sum_{\mathbf{A}_\kappa \in \mathcal{A}} r(\mathcal{C}_D, \mathbf{A}_\kappa) \quad (4)$$

where $\kappa = 1 \ldots |\mathcal{A}|$ (Gates and Rocha, 2016). Since this notion of control depends only on the enumeration of attractors, CANA can leverage a SAT-based bounded model algorithm to quantify the mean fraction of reachable attractors in a BN with thousands of variables (Dubrova and Teslenko, 2011). **Figure 3A** shows the values of $\overline{R}_D$ and $\overline{A}_D$ for various sizes of driver sets D in the *Thaliana* GRN.

Finally, CANA also provides the functionality to approximate the minimal driver variable subset using two prominent network control methodologies: *Structural Controlability* (SC) (Lin, 1974; Liu et al., 2011) and *Minimum Dominating Set* (MDS) (Nacher, 2012; Nacher and Akutsu, 2013).

5. SUMMARY AND CONCLUSION

We presented a novel, open-source and publicly-available software platform that integrates the analytic methodology used to study canalization in automata network dynamics. This methodology can now be used by others to simplify large automata networks, especially those in models of biochemical regulation dynamics. In addition to the extraction and visualization of specific effective pathways that regulate key phenotypic outcomes in a sea of redundant interaction, CANA includes functionality to measure canalization, uncover control variables, and study dynamical modularity, robustness, and criticality. We hope that the consolidation of redundancy and control algorithms into one package encourages other researchers to build upon our work on canalization, thus adding additional algorithms to CANA.

DATA AVAILABILITY STATEMENT

The CANA python package and all datasets analyzed for this study can be found on Github at github.com/rionbr/CANA.

AUTHOR CONTRIBUTIONS

RC, AG, and XW contributed to the CANA package. LR developed the per-input measures of canalization and the effective graph formulation. RC, AG, and LR wrote the manuscript.

FUNDING

ACKNOWLEDGMENTS

REFERENCES

Akutsu, T., Hayashida, M., Ching, W.-K., and Ng, M. K. (2007). Control of Boolean networks: hardness results and algorithms for tree structured networks. *J. Theor. Biol.* 244, 670–679. doi: 10.1016/j.jtbi.2006.09.023

Albert, I., Thakar, J., Li, S., Zhang, R., and Albert, R. (2008). Boolean network simulations for life scientists. *Source Code Biol. Med.* 3:16. doi: 10.1186/1751-0473-3-16

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

Aldana, M. (2003). Boolean dynamics of networks with scale-free topology. *Phys. D Nonlinear Phen.* 185, 45–66. doi: 10.1016/S0167-2789(03)00174-X

Assmann, S. M. and Albert, R. (2009). Discrete dynamic modeling with asynchronous update, or how to model complex systems in the

absence of quantitative information. *Methods Mol. Biol* 553, 207–225. doi: 10.1007/978-1-60327-563-7_10

Bender, C., Henjes, F., Fröhlich, H., Wiemann, S., Korf, U., and Beißbarth, T. (2010). Dynamic deterministic effects propagation networks: learning signalling pathways from longitudinal protein array data. *Bioinformatics* 26, i596–i602. doi: 10.1093/bioinformatics/btq385

Bornholdt, S. (2005). Systems biology. Less is more in modeling large genetic networks. *Science* 310, 449–451. doi: 10.1126/science.1119959

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

Brayton, R. K., Sangiovanni-Vincentelli, A. L., McMullen, C. T., and Hachtel, G. D. (1984). *Logic Minimization Algorithms for VLSI Synthesis*. Norwell, MA: Kluwer Academic Publishers.

Chaos, A., Aldana, M., Espinosa-Soto, C., de Leon, B. G. P., Arroyo, A. G., and Alvarez-Buylla, E. R. (2006). From genes to flower patterns and evolution: dynamic models of gene regulatory networks. *J. Plant Growth Regul.* 25, 278–289. doi: 10.1007/s00344-006-0068-8

Chechik, G., Oh, E., Rando, O., Weissman, J., Regev, A., and Koller, D. (2008). Activity motifs reveal principles of timing in transcriptional control of the yeast metabolic network. *Nat. Biotechnol.* 26, 1251–1259. doi: 10.1038/nbt.1499

Choi, M., Shi, J., Zhu, Y., Yang, R., and Cho, K.-H. (2017). Network dynamics-based cancer panel stratification for systemic prediction of anticancer drug response. *Nat. Commun.* 8:1940. doi: 10.1038/s41467-017-02160-5

Correia, R. B., Gates, A. J., Wang, X., and Rocha, L. M. (2018). *Canalization: Control & Redundancy in Boolean Networks*. Available online at: https://rionbr.github.io/CANA

Dubrova, E., and Teslenko, M. (2011). A SAT-based algorithm for finding attractors in synchronous boolean networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 8, 1393–1399. doi: 10.1109/TCBB.2010.20

Ellson, J., Gansner, E., Koutsofios, L., North, S. C., and Woodhull, G. (2002). "Graphviz—open source graph drawing tools," in *Graph Drawing*, eds P. Mutzel, M. Jünger, and S. Leipert (Berlin; Heidelberg: Springer), 483–484.

Gates, A., Manicka, S., Marques-Pita, M., and Rocha, L. M. (2016). "The effective structure of complex networks drives dynamics, criticality and control," in *Complex Networks 2016: The 5th International Workshop on Complex Networks & Their Applications* (Milan), 107–109.

Gates, A., and Rocha, L. M. (2016). Control of complex networks requires both structure and dynamics. *Sci. Rep.* 6:24456. doi: 10.1038/srep24456

Helikar, T., Konvalina, J., Heidel, J., and Rogers, J. A. (2008). Emergent decision-making in biological signal transduction networks. *Proc. Natl. Acad. Sci. U.S.A.* 105, 1913–1918. doi: 10.1073/pnas.0705088105

Helikar, T., Kowal, B., McClenathan, S., Bruckner, M., Rowley, T., Madrahimov, A., et al. (2012). The cell collective: toward an open and collaborative approach to systems biology. *BMC Syst. Biol.* 6:96. doi: 10.1186/1752-0509-6-96

Ideker, T., and Nussinov, R. (2017). Network approaches and applications in biology. *PLoS Comput. Biol.* 13:e1005771. doi: 10.1371/journal.pcbi.1005771

Iyengar, R. (2009). Why we need quantitative dynamic models. *Sci. Signal.* 2:eg3. doi: 10.1126/scisignal.264eg3

Kauffman, S. (1969). Homeostatis and differentiation in random genetic control networks. *Nature* 224, 177–178.

Kauffman, S. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. New York, NY: Oxford University Press.

Kauffman, S., Peterson, C., Samuelsson, B., and Troein, C. (2004). Genetic networks with canalyzing boolean rules are always stable. *Proc. Natl. Acad. Sci. U.S.A.* 101, 17102–17107. doi: 10.1073/pnas.0407783101

Kauffman, S. A. (1984). Emergent properties in random complex automata. *Phys. D Nonlinear Phen.* 10, 145–156. doi: 10.1016/0167-2789(84)90257-4

Kurten, K. E. (1988). Correspondence between neural threshold networks and kauffman boolean cellular automata. *J. Phys. A Math. Gen.* 21:L615. doi: 10.1088/0305-4470/21/11/009

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

Lin, C.-T. (1974). Structural controllability. *IEEE Trans. Automatic Control* 19, 201–208. doi: 10.1109/TAC.1974.1100557

Liu, Y.-Y., Slotine, J.-J., and Barabási, A.-L. (2011). Controllability of complex networks. *Nature* 473, 167–173. doi: 10.1038/nature10011

Manicka, S. (2017). *The Role of Canalization in the Spreading of Perturbations in Boolean Networks*. Doctoral dissertation, Indiana University, Informatics and Computing.

Marques-Pita, M., and Rocha, L. M. (2013). Canalization and control in automata networks: body segmentation in drosophila melanogaster. *PLoS ONE* 8:e55946. doi: 10.1371/journal.pone.0055946

Masel, J., and Maughan, H. (2007). Mutations leading to loss of sporulation ability in bacillus subtilis are sufficiently frequent to favor genetic canalization. *Genetics* 175, 453–457. doi: 10.1534/genetics.106.065201

McCulloch, W. S., and Pitts, W. H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophysics* 5, 115–133. doi: 10.1007/BF02478259

Müller, F.-J., and Schuppert, A. (2011). Few inputs can reprogram biological networks. *Nature* 478, E4–E5. doi: 10.1038/nature10543

Nacher, J. C., and Akutsu, T. (2013). Structural controllability of unidirectional bipartite networks. *Sci. Rep.* 3:1647. doi: 10.1038/srep01647

Nacher, J. C., and Akutsu, T. (2012). Dominating scale-free networks with variable scaling exponent: heterogeneous networks are not difficult to control. *N. J. Phys.* 14:073005. doi: 10.1088/1367-2630/14/7/073005

Quine, W. V. (1955). A Way to Simplify Truth Functions. *Am. Math. Monthly* 62, 627–631. doi: 10.1080/00029890.1955.11988710

Reichhardt, C. J. O. and Bassler, K. E. (2007). Canalization and symmetry in boolean models for genetic regulatory networks. *J. Phys. A Math. Theor.* 40:4339. doi: 10.1088/1751-8113/40/16/006

Siegal, M. L., and Bergman, A. (2002). Waddington's canalization revisited: developmental stability and evolution. *Proc. Natl. Acad. Sci. U.S.A.* 99, 10528–10532. doi: 10.1073/pnas.102303999

Sontag, E. D. (1998). *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. New York, NY: Springer.

ten Tusscher, K. H., and Hogeweg, P. (2009). The role of genome and gene regulatory network canalization in the evolution of multi-trait polymorphisms and sympatric speciation. *BMC Evol. Biol.* 9:159. doi: 10.1186/1471-2148-9-159

Terfve, C., Cokelaer, T., Henriques, D., MacNamara, A., Goncalves, E., Morris, M. K., et al. (2012). Cellnoptr: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC Syst. Biol.* 6:133. doi: 10.1186/1752-0509-6-133

Trinh, H.-C., Le, D.-H., and Kwon, Y.-K. (2014). Panet: a GPU-based tool for fast parallel analysis of robustness dynamics and feed-forward/feedback loop structures in large-scale biological networks. *PLoS ONE* 9:e103010. doi: 10.1371/journal.pone.0103010

Waddington, C. H. (1942). Canalization of development and the inheritance of acquired characters. *Nature* 150, 563–565.

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

Willadsen, K., and Wiles, J. (2007). Robustness and state-space structure of Boolean gene regulatory models. *J. Theor. Biol.* 249, 749–765. doi: 10.1016/j.jtbi.2007.09.004

Wuensche, A. (1998). "Discrete dynamical networks and their attractor basins," in *Complex Systems'98*, eds R. Standish, B. Henry, S. Watt, R. Marks, R. Stocker, D. Green, S. Keen, and T. Bossomaier (Sydney, NSW: University of New South Wales), 1–24.

Zhang, R., Shah, M. V., Yang, J., Nyland, S. B., Liu, X., Yun, J. K., et al. (2008). Network model of survival signaling in large granular lymphocyte leukemia. *Proc. Natl. Acad. Sci. U.S.A.* 105, 16308–16313. doi: 10.1073/pnas.0806447105