Boolean function metrics can assist modelers to check and choose logical rules

John Zobolas[1],[*], Pedro T. Monteiro[2],[3], Martin Kuiper[1] and Åsmund Flobak[4],[5]

[1]Department of Biology, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

[2]Department of Computer Science and Engineering, Instituto Superior Técnico (IST) - Universidade de Lisboa, Lisbon, Portugal

[3]INESC-ID, Lisbon, Portugal

[4]Department of Clinical and Molecular Medicine, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

[5]The Cancer Clinic, St. Olav's Hospital, Trondheim, Norway

[*]To whom correspondence should be addressed.

April 6, 2021

Abstract

Computational models of biological processes provide one of the most powerful methods for a detailed analysis of the mechanisms that drive the behavior of complex systems. Logic-based modeling has enhanced our understanding and interpretation of those systems. Defining rules that determine how the output activity of biological entities is regulated by their respective inputs has proven to be challenging. Partly this is because of the inherent noise in data that allows multiple model parameterizations to fit the experimental observations, but some of it is also due to the fact that models become increasingly larger, making the use of automated tools to assemble the underlying rules indispensable.

We present several Boolean function metrics that provide modelers with the appropriate framework to analyze the impact of a particular model parameterization. We demonstrate the link between a semantic characterization of a Boolean function and its consistency with the model's underlying regulatory structure. We further define the properties that outline such consistency and show that several of the Boolean functions under study violate them, questioning their biological plausibility and subsequent use. We also illustrate that regulatory functions can have major differences with regard to their asymptotic output behavior, with some of them being biased towards specific Boolean outcomes when others are dependent on the ratio between activating and inhibitory regulators.

Application results show that in a specific signaling cancer network, the function bias can be used to guide the choice of logical operators for a model that matches data observations. Moreover, graph analysis indicates that the standardized Boolean function bias becomes more prominent with increasing numbers of regulators, confirming the fact that rule specification can effectively determine regulatory outcome despite the complex dynamics of biological networks.

Keywords Boolean regulatory networks · Boolean functions · Truth Density · Bias · Complexity

1 Introduction

The understanding of biological processes has been greatly stimulated by systems biology approaches [1, 2, 3]. The integration of mathematical models with the underlying biological knowledge and empirical observations can help us observe emergent systems properties, test new hypotheses, enhance the interpretability of the studied systems and guide innovations in areas such as medicine and drug discovery [4]. While multiple mathematical modeling frameworks exist, the scarcity of experimental data and the challenges posed by the development of quantitative large-scale biological networks, has favoured the simplicity and intuitiveness of more qualitative approaches, such as logic-based modeling [5].

At the heart of the mathematical representation of molecular biological networks lies the concept of regulation. Regulation of activity, typically by changing the modification state, location or concentration of a biological entity, is a process which can be expressed by a mathematical function that combines the various regulatory inputs that affect the target, with a logic that describes how these regulators are integrated. In Boolean logic-based modeling, the regulatory inputs are entities which can be expressed in two states: active (1) or inactive (0). These entities are combined with logical rules to derive the *Boolean regulatory function* (BRF) of the target entity. For every possible regulatory input (combination of 0 and 1's) the BRF will produce the end regulatory product, which is the activity of the target (0 or 1).

The construction of a Boolean computational model starts with the assembly of information from literature and experimental observations, in the form of a Prior Knowledge Network (PKN), i.e. a list of network entities and their causal interactions (positive or negative) [6, 7]. The use of a PKN for accurate representation of biological reality and subsequent analysis and simulation requires the definition of the model formalism. This is one of the most important steps in dynamical modeling since it directly translates to the choice of BRFs, i.e. the logical rules that together with the regulators define the activity state of each network target [8]. There have been several approaches related to the choice of BRFs, from using a standardized format [9], to automatically generating all possible BRFs compatible with the PKN and calibrating the rules in order to fit perturbation data [10, 11, 12]. State-of-the-art approaches involve the automated construction of large-scale logical networks by inferring the logical rules from the topology and semantics of molecular interaction maps [13].

Regardless of how a logical model is constructed, it has been shown in practice that expert curation, i.e. the manual fine-tuning of the logical rules to fit experimental data, can result in highly predictive models [14, 15], yet this is not trivially obtained with automatically constructed networks [16]. Because of the large function space complemented with a sparsity of observations and inherent noise in existing data, there is a wide range of plausible BRFs. Thus, it is crucial to properly define function characteristics that can guide the modeler to a more informed function choice. Our work is focused on explicating some of these metrics and using them to show for example which BRFs can be discarded due to biological inconsistencies with the underlying regulatory topology and which are biased towards specific Boolean outcomes.

The paper is structured as follows: Section 2 provides a list of notations and definitions to be used later in the text. In Section 3, we discuss the benefits of using the equivalent disjunctive normal form of a Boolean function to delineate its biological interpretability. In Section 4, we provide a set of properties that characterize the Boolean functions that are consistent with a given regulatory topology and show that several functions under study violate them. In Section 5, we present the truth density metric as a means to evaluate if a Boolean function is biased or balanced with increasing number of regulators. We also discuss the asymptotic properties of different functions relating to the ratio between activators and inhibitors. Lastly, in Section 6, we present evidence that the standardized Boolean functions are indeed biased and show how modelers can exploit such information for their own benefit. The results are demonstrated in Boolean models derived from a cancer signaling network as well as from scale-free topologies that are applicable to most biological networks. We close the paper with some discussion points in Section 7 and directions for future research in Section 8.

2 Background

2.1 Boolean regulatory functions

Boolean regulatory functions (BRFs) are Boolean functions used in the context of biological networks and modeling. A mathematical description of such a function associates the activity output of a target biological entity with the Boolean input values of n variables (the *regulators*), such that $f_{BRF} : \{0,1\}^n \rightarrow \{0,1\}$. Thus, the target's output state is binary, i.e. either 0 (*False*, denoting an inactive or inhibited state) or 1 (*True*, indicating an active state).

One intuitive representation of a Boolean function is its *truth table*, which is a list of all possible Boolean input configurations of the n regulators along with their associated function output. Since every regulator can be assigned two possible values (0 and 1), the total number of input configurations (i.e. rows) in a truth table is 2^n. For example, a Boolean function $f(x_1, x_2, x_3)$ with 3 regulators has a total of $2^3 = 8$ rows in its corresponding truth table, starting from the input configuration $(0,0,0)$ and ending with $(1,1,1)$ (Table 1).

The total number of BRFs with n regulators is 2^{2^n} since for each of the 2^n input configurations (i.e. rows of the truth table) there can be two possible function outcomes (0 or 1). For example, with 3 regulators and a total of 8 rows in the truth table, that would be a total of $2^8 = 256$ functions, three of which are shown in Table 1.

2.2 Disjunctive normal form

The most frequently used form of a Boolean function is its analytical expression, where variables are connected with logical operators such as AND ($\wedge$), OR ($\vee$), NOT ($\neg$), XOR ($\oplus$), etc. and the output of the function is calculated using basic Boolean algebra. In Table 1 for example, we provide the analytical forms for the functions f_1 and f_2. Note that there can be multiple analytical forms that essentially compute the same function, e.g. another form of the f_1 function is $f_1' = (\neg x_1 \wedge x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge x_2 \wedge \neg x_3)$.

Truth Table			Boolean functions		
x_1	x_2	x_3	$f_1 = (x_1 \wedge \neg x_3) \vee (x_2 \wedge \neg x_3)$	$f_2 = x_1 \vee (\neg x_2 \wedge \neg x_3)$	$f_3 = 1$
0	0	0	0	1	1
0	0	1	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1

Table 1: Truth table of three Boolean functions with three input variables x_1, x_2 and x_3. Functions f_1 and f_2 are expressed in disjunctive normal form (DNF) with the minimum possible number of terms. f_3 is a tautology.

This brings us to the notion of a general form which could be used to define useful metrics common to all Boolean functions (e.g. complexity), as well as the need to provide minimal forms based on specific criteria. For example, a more compact function form enhances readability, which can be seen by comparing f_1 with f_1'.

Every Boolean function can be represented in a *disjunctive normal form* (DNF), requiring only AND ($\land$), OR ($\lor$) and NOT ($\neg$) operators as building blocks. In such a representation, *literals*, which are variables (e.g. positive literal x) or their logical negations (e.g. negative literal NOT x), are connected by AND's, producing *terms*, which are then in turn connected by OR's [17]. For example, every function in Table 1 is expressed in DNF, while the Boolean expressions $\neg(x_1 \lor x_2)$ and $\neg(x_1 \land x_2) \lor x_3$ are not. Note that a Boolean function can have multiple DNF formulations.

2.3 Link operator functions

We consider the class of BRFs that partitions the input regulators to two sets: the set of positive regulators (*activators*) and the set of negative regulators (*inhibitors*). Let f be such a Boolean function $f_{BRF}(x, y)$: $\{0, 1\}^n \to \{0, 1\}$, with $m \geq 1$ activators $x = \{x_i\}_{i=1}^m$ and $k \geq 1$ inhibitors $y = \{y_j\}_{j=1}^k$, that is a total of $n = m + k$ regulators. The *link operator* BRFs have an analytical formula which places the two distinct types of regulators in two separate expressions, while connecting them with a special logical operator that we call a *link operator*. An example of such a function that has been used extensively in the logical modeling literature is the standardized BRF formula with the "AND-NOT" link operator [9]:

$$f_{AND-NOT}(x, y) = \left(\bigvee_{i=1}^m x_i \right) \land \neg \left(\bigvee_{j=1}^k y_j \right) \tag{1}$$

A variation of the above function is the "OR-NOT" link operator function:

$$f_{OR-NOT}(x, y) = \left(\bigvee_{i=1}^m x_i \right) \lor \neg \left(\bigvee_{j=1}^k y_j \right) \tag{2}$$

Note that the presence of the link operator is what forces the condition $m, k \geq 1$ (at least one regulator in each category). For the rest of this work, we will not consider BRFs with only one type of regulator, since these can be represented by simple logical functions without loss of biological consistency. Following the notation introduced in Mendoza et al. [9], in the case of only positive regulators, the presence of at least one activator makes the target active, i.e. $f(x) = \bigvee_{i=1}^m x_i$. In the case of only inhibitory regulators, the presence of at least one inhibitor is sufficient to make the target inactive, i.e. $f(y) = \neg \bigvee_{j=1}^k y_j = \bigwedge_{j=1}^k \neg y_j$.

Borrowing notation from circuit theory, we will also use other link operators like the "NAND", "NOR", "XNOR" gates, with or without the "NOT" symbol in front. Note that the logical operator used to connect the same type of regulators (e.g. the activators) is usually OR, but other operators could be used as well.

Another link operator function that we will consider in this work is the "Pairs" function:

$$f_{Pairs}(x, y) = \bigvee_{\forall(i,j)}^{m,k} (x_i \land \neg y_j) \tag{3}$$

The intuition behind the name is derived from the fact that the function will return *True* if there is at least one pair of regulators consisting of a present activator and an absent inhibitor. For a formulation of the "Pairs" function that is consistent with the link operator terminology as defined above, see (Eq. 9).

2.4 Threshold functions

Threshold functions are a special type of Boolean functions, the output of which depends on the condition that the sum of (possibly weighted) activities of the input regulators surpasses a given *threshold* value [18, 19].

In this work we will consider two simple threshold functions, which both output $True$ when the number of present activators is larger than the number of present inhibitors. As such, the activities of the positive and negative regulators are combined in an *additive* manner, with their respective assigned weights set to ± 1 and the threshold parameter to 0, formulating thus a *majority rule* which defines the value of the function [20, 21]. These functions differ with regards to their output when there is balance between the activities of the positive and negative regulators: the first outputs 1 (the activators "win") while the second outputs 0 (the inhibitors "win"):

$$f_{Act-win}(x, y) = \begin{cases} 1, & \sum_{i=1}^{m} x_i \geq \sum_{j=1}^{k} y_j \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$f_{Inh-win}(x, y) = \begin{cases} 1, & \sum_{i=1}^{m} x_i > \sum_{j=1}^{k} y_j \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

3 Disjunctive Normal Form unmasks biological interpretation

3.1 Interpretability issues in Boolean modeling

Two main features make Boolean modeling attractive to users. First, transforming conditions for the activation or inhibition of a target biological entity to Boolean equations is a relatively easy task using a qualitative, logic-based modeling formalism. Second, the reverse is also true, i.e. Boolean equations can be more interpretable and closer to a simplified description of biological reality that "makes sense" than the use of other kinds of formalisms (e.g. kinetic modeling). For example, consider the simple case of a target entity, which is regulated by one positive regulator x_1 and one negative regulator y_1. The use of the "AND-NOT" link operator function in this case (Eq. 1) is very easy to understand and interpret since the formula directly connects to the underlying biology. Thus, the mathematical formulation is simply written as $f_{AND-NOT} = x_1$ AND NOT y_1, while the modeler reads "the target becomes active when x_1 (the activator) is present and y_1 (the inhibitor) absent".

Issues start arising when considering the *interpretability* of such Boolean expressions in cases where a larger number of regulators act on a target, e.g. in a more complex scenario with three positive (x_1, x_2, x_3) and three negative (y_1, y_2, y_3) regulators, the mathematical formulation expressing the target's activity output can be easily written using the link operator function form, as $f_{AND-NOT} = (x_1$ OR x_2 OR $x_3)$ AND NOT $(y_1$ OR y_2 OR $y_3)$. A modeler could read this as "the target becomes active when at least one activator is present, and all of its inhibitory regulators are absent", but a precise semantic description that explicates the conditions under which the target gets activated, can in general be difficult to assess. A similar issue arises when reflecting on the use of a different link operator instead of the standard "AND-NOT" or even of an entirely different regulatory function, for which the biological interpretation might be difficult to derive from the expression itself.

BRF (standard form)	BRF (CDNF)	Biological Interpretation	Consistent	Complexity
$(x_1$ OR $x_2)$ **NOR** $(y_1$ OR $y_2)$	NOT x_1 AND NOT x_2 AND NOT y_1 AND NOT y_2	Absence of all regulators	NO	1 (always)
$(x_1$ OR $x_2)$ **NAND** $(y_1$ OR $y_2)$	(NOT x_1 AND NOT x_2) **OR** (NOT y_1 AND NOT y_2)	Absence of all activators **or** absence of all inhibitors	NO	2 (always)
$(x_1$ OR $x_2)$ **AND NOT** $(y_1$ OR $y_2)$ "AND-NOT" (Eq. 1)	$(x_1$ AND NOT y_1 AND NOT $y_2)$ **OR** $(x_2$ AND NOT y_1 AND NOT $y_2)$	Presence of at least one activator **and** absence of all inhibitors	YES	2 (m)
$(x_1$ OR $x_2)$ **NOR NOT** $(y_1$ OR $y_2)$	$(y_1$ AND NOT x_1 AND NOT $x_2)$ **OR** $(y_2$ AND NOT x_1 AND NOT $x_2)$	Presence of at least one inhibitor **and** absence of all activators	NO	2 (k)
$(x_1$ OR $x_2)$ **OR NOT** $(y_1$ OR $y_2)$ "OR-NOT" (Eq. 2)	x_1 **OR** x_2 **OR** (NOT y_1 AND NOT y_2)	Presence of any activator **or** absence of all inhibitors	YES	3 ($m+1$)
$(x_1$ OR $x_2)$ **NAND NOT** $(y_1$ OR $y_2)$	y_1 **OR** y_2 **OR** (NOT x_1 AND NOT x_2)	Presence of any inhibitor **or** absence of all activators	NO	3 ($k+1$)
$(x_1$ OR $x_2)$ **XOR** $(y_1$ OR $y_2)$	$(x_1$ AND NOT y_1 AND NOT $y_2)$ **OR** $(x_2$ AND NOT y_1 AND NOT $y_2)$ **OR** (NOT x_1 AND NOT x_2 AND y_1) **OR** (NOT x_1 AND NOT x_2 AND y_2)	Presence of at least one activator and absence of all inhibitors **or** presence of at least one inhibitor and absence of all activators	NO	4 ($m+k$)
$(x_1$ OR $x_2)$ AND (NOT y_1 OR NOT $y_2)$ "Pairs" (Eq. 3)	$(x_1$ AND NOT $y_1)$ **OR** $(x_1$ AND NOT $y_2)$ **OR** $(x_2$ AND NOT $y_1)$ **OR** $(x_2$ AND NOT $y_2)$	Presence of at least one activator **and** absence of at least one inhibitor	YES	4 ($m \times k$)
$(x_1$ OR $x_2)$ **XNOR** $(y_1$ OR $y_2)$	$(x_1$ AND $y_1)$ **OR** $(x_1$ AND $y_2)$ **OR** $(x_2$ AND $y_1)$ **OR** $(x_2$ AND $y_2)$ **OR** (NOT x_1 AND NOT x_2 AND NOT y_1 AND NOT y_2)	Presence of at least one activator and inhibitor pair **or** absence of all regulators	NO	5 ($m \times k+1$)
$True$ when $x_1 + x_2 > y_1 + y_2$ "Inh-win" (Eq. 5)	$(x_1$ AND x_2 AND NOT $y_1)$ **OR** $(x_1$ AND x_2 AND NOT $y_2)$ **OR** $(x_1$ AND NOT y_1 AND NOT $y_2)$ **OR** $(x_2$ AND NOT y_1 AND NOT $y_2)$	Number of present activators is larger than the number of present inhibitors	YES	4
$True$ when $x_1 + x_2 \geq y_1 + y_2$ "Act-win" (Eq. 4)	$(x_1$ AND $x_2)$ **OR** $(x_1$ AND NOT $y_1)$ **OR** $(x_1$ AND NOT $y_2)$ **OR** $(x_2$ AND NOT $y_1)$ **OR** $(x_2$ AND NOT $y_2)$	Number of present activators is larger than or equal to the number of present inhibitors	YES	5

Table 2: Several Boolean regulatory functions with four regulators ($m = 2$ positive $\{x_1, x_2\}$, $k = 2$ negative $\{y_1, y_2\}$) and some metrics are presented. The two first columns provide two different function forms: a standard one, i.e. either the link operator form distinguishing activating and inhibiting regulators or a simple description in the case of the threshold functions, and the CDNF which is a special case of DNF (Section 4.1). The "Biological Interpretation" states in words the conditions that make a BRF become $True$, and is explicitly translated from the terms in the corresponding CDNF. The "Consistent" column states if the functions satisfy the properties 1-3 from Section 4.1 (YES, green-colored) or there are inconsistencies with the underlying regulatory structure (NO, red-colored), i.e. if an activator (resp. inhibitor) appears as a negative (resp. positive) literal in the corresponding CDNF. The functions are sorted according to an increasing complexity metric ("Complexity" column), which is the number of terms in each respective, minimum-length CDNF expression. In parentheses we provide the generalized formula for the number of CDNF terms of the link operator functions with m activators and k inhibitors.

3.2 DNF links to biological semantics

We argue here that the DNF is the most adequate function form to help us address the aforementioned issues. Every Boolean regulatory function expressed in DNF, has a biological characterization that is directly derived from the formula itself: each term in the DNF is an activation condition, i.e. a list of regulators, some present (the positive literals) and some absent (the negative literals), which, when combined, make the target (output of the function) active. Further merging of all the conditions using OR-semantics into a description of how the regulators influence the target's output, facilitates the biological interpretation of any Boolean regulatory function.

In Table 2, we show a list of BRFs with two positive and two negative regulators. Most of the BRFs presented have a different link operator separating the activators from the inhibitors. Using the functions standard expressions (1st column) makes it very hard to derive a meaningful biological characterization as expressed in the 3rd column of Table 2. For example, defining a meaningful description of the "NOR" or "NAND-NOT" equations using only their standard expression, is a very difficult task. In contrast, by using the equivalent DNFs (2nd column) we can make an explicit, "1-1" correspondence between mathematical formulation and biological interpretation and use it to compare the different functions' meanings. Thus, by expressing the "NAND-NOT" equation in DNF, we can precisely identify the conditions that make the outcome of the function *True* and translate these into a meaningful description such as "Presence of any inhibitor or absence of all activators". Consequently, we are led to a generalized and independent of the number of regulators description of this link operator function. Such a description is intuitive to human interpretation and reasoning, in terms of the function's applicability, e.g. in comparing the "AND-NOT" and "NAND-NOT" biological interpretations, we see that the first is semantically plausible while the second completely contradicts the underlying biology.

4 Characterizing consistent regulatory functions

4.1 The 3 consistency properties

As observed in Table 2, not only can the DNF be used to uncover the biological interpretation of any BRF and subsequently help determine its plausibility, but it also provides a means to compare the different function meanings. Still, we need a more refined, technical description that is able to express the implausibility of the "NAND-NOT" or "NOR" cases directly from their mathematical formulas, and which would be applicable to every BRF. We define the *consistency* attribute of a BRF to describe its compliance with the underlying regulatory network structure.

The first step in making a Boolean model is to build a graph (PKN), assembling the regulatory entities of interest from various databases or the scientific literature, and use causality information to connect them through their regulatory action on other entities. As such, a network structure can be defined, in which entities can regulate (either positively or negatively) some of the other entities. Using such a simple network-driven formalization, we define a set of three properties that describe the set of all the *consistent Boolean regulatory functions*, i.e. the functions that comply with the underlying regulatory structure. So, for a consistent BRF, the following propositions are satisfied [22]:

1. Its regulators can be partitioned into two disjoint sets: the set of *activators* (positive regulators, enhance target's activity) and the set of *inhibitors* (negative regulators, suppress target's activity). This stems from the fact that every interaction in the PKN has a fixed sign (either positive or negative). As such, there are no dual regulations, i.e. a regulator cannot activate and inhibit a target at the same time. This property essentially makes the set of consistent BRFs a subset of the *monotone* Boolean functions [17].

2. All regulators are *essential*: for every regulatory input, inverting their values, will also, in at least one configuration of states of other regulators, change the output of the function. This means that all regulators are indispensable for deriving the target's activity output.

3. A consistent BRF can be represented in a unique *complete* DNF (CDNF) which is also known as Blake's Canonical Form [23]. This is a consequence of property (1), since monotone Boolean functions expressed in any DNF, can be further simplified by removing redundant literals, resulting in the equivalent unique CDNF expression [17]. This property is really important since it allows us to identify which regulatory entities are activators and which are inhibitors from the corresponding CDNF expression of a consistent BRF: an activator will always appear as a positive literal, whereas an inhibitor will always appear as a negative literal.

We provide an example to delineate the difference between the DNF and CDNF forms and show violations of the consistency properties. In Table 3, we present three Boolean functions, expressing the output of a target regulated by one activator (x_1) and one inhibitor (y_1). The functions f_2 and f_3 are in CDNF whereas f_1 is in normal DNF, since the positive regulator x_1 appears both as a positive and a negative literal (i.e. it acts as a dual regulator, making f_1 inconsistent). Notice that f_1 reduces to f_2 by removing the redundant negative literal ($\neg x_1$) in the term ($\neg x_1 \wedge y_1$): y_1 "absorbs" the larger term and thus a shorter expression manifests, one that covers more $True$ outcomes (i.e. 1's in the truth table). In addition, we observe that f_2 is inconsistent, since inhibitor y_1 appears as a positive literal. On the other hand, using a negative literal for inhibitor y_1 and a positive one for activator x_1, makes f_3 consistent.

Truth Table		Term	Boolean functions		
x_1	y_1	$(\neg x_1 \wedge y_1)$	$f_1 = x_1 \vee (\neg x_1 \wedge y_1)$	$f_2 = x_1 \vee y_1$	$f_3 = x_1 \vee \neg y_1$
0	0	0	0	0	1
0	1	1	1	1	0
1	0	0	1	1	1
1	1	0	1	1	1

Table 3: Truth table of three different Boolean regulatory functions with two input regulators, one positive (x_1) and one negative (y_1). All functions are expressed in DNF. f_1 and f_2 result in the same target Boolean output, with f_2 expressed in CDNF. Activator x_1 regulates the target both positively and negatively in f_1, making the function non-monotone and thus inconsistent. Inhibitor y_1 is a positive literal in f_2's CDNF, making it inconsistent as well. Function f_3 is consistent since it's written in CDNF with the activator x_1 and inhibitor y_1 appearing as positive and negative literals respectively.

4.2 Most link operator functions are inconsistent

Examining Table 2, we note that the 2nd column presents not just any DNF expression of the studied Boolean regulatory functions, but precisely the CDNF. Thus we can immediately identify which BRFs violate at least one of the three properties discussed in Section 4.1 and are therefore inconsistent with the regulatory topology (this information is presented in the 4th column, labeled "Consistent"). Two examples of such inconsistencies include the "NAND-NOT" and "XOR" link operator functions, which have terms in their corresponding CDNF in which an activator x_i appears as a negative literal (NOT x_i) and an inhibitor y_j as a positive literal (as itself). In total, from all the BRFs presented in Table 2, only the standardized "AND-NOT", the "OR-NOT", the "Pairs" and the two threshold functions respect the underlying regulatory topology, as

can be verified by examining their respective CDNFs. The rest of the link operator functions presented are inconsistent and will not be considered for further analysis in this paper.

5 Truth Density as a measure of expected function output

In this section we present another interesting Boolean function metric, whose properties can be used to add further knowledge about a Boolean function's behavior. This metric, which we call *truth density*, allows us to project what the regulatory target's output will most likely be when the number of input regulators changes and investigate how the ratio between activators and inhibitors may affect that output. From a modeler's perspective, this metric is useful to check if an assigned model parameterization (i.e. use of a specific BRF) can asymptotically predefine the activity state of some targets. Equipped with this knowledge, a modeler can verify the degree of fitness with the observations that such a parameterization allows, and thus discard a specific function in favor of another, if the latter has a truth density value that better matches the outcome observed in the data.

5.1 Truth Density

We define the truth density (TD) of a Boolean function as the fraction of all input configurations in its corresponding truth table that yield a $True$ (1) outcome. As such, $TD \in [0, 1]$. This quantity was first introduced in [24] and more recently in [25] under the name of *bias* and was similarly defined as the probability that a Boolean function takes on the value 1. Using the example with the three Boolean functions from Table 1, we have $TD_{f_1} = 3/8 = 0.375$, $TD_{f_2} = 5/8 = 0.625$ and $TD_{f_3} = 8/8 = 1$, where the last function is a tautology, with the maximum possible truth density. Colloquially, we can say that a Boolean function is *biased*, when it's truth density is close to 0 or 1. Since the size of a truth table grows exponentially with the number of inputs of the Boolean function (n inputs correspond to 2^n rows), the existence of bias conveys the information that most of the input regulatory configurations result in either an activated or inhibited target (bias towards 1 or 0 respectively). On the other hand, we shall say that a Boolean function is *balanced*, if it takes on the values 0 and 1 equally often, or equivalently, it's truth density is approximately centered around 1/2 [26].

5.2 Asymptotic truth density results of the consistent regulatory functions

In Appendix A, we present a list of propositions and proofs that provide the exact truth density formulas for the generic forms of the five consistent BRFs we studied in previous sections, namely the "AND-NOT", "OR-NOT" and "Pairs" link operator functions, and the two threshold functions, "Act-win" and "Inh-win". A very important element that enables the straightforward derivation of these formulas, is the use of the equivalent DNF expressions in the proofs, especially for the case of the link operator Boolean functions. We also noted that the truth densities of all the aforementioned BRFs depend on two variables: the number of activators and the number of inhibitors (the total number of regulators also appears as a separate variable but it depends on the first two, i.e. it is just their sum). Thus, we logically asked if a BRF's truth density asymptotically tends towards specific values in the $[0, 1]$ interval (e.g. the function could be biased or balanced), when the number of its input regulators increases or the ratio between activators and inhibitors changes. The results of the asymptotic behavior of the truth density formulas are analytically presented in Appendix B.

The asymptotic analysis of the truth density formulas confirmed the intuitive perception that the link operator "AND-NOT" and "OR-NOT" functions show a characteristically opposite behavior with increasing number of regulators: the standardized "AND-NOT" formula depends only on the number of inhibitors and its output tends towards 0, whereas the "OR-NOT" formula depends only on the number of activators and is biased towards 1. On the other hand, the "Pairs" and threshold functions truth densities don't have an asymptotic

limit since they depend on both the number of activators and inhibitors. Therefore, we proceeded in clarifying the role of the *activator-to-inhibitor* ratio by investigating three scenarios which explicitly reveal the functions truth density behavior for a significantly large number of regulators:

- A 1 : 1 activator-to-inhibitor ratio, where approximately half of the regulators are activators and half are inhibitors.
- A high activator-to-inhibitor ratio, where all regulators are activators except one inhibitor.
- A low activator-to-inhibitor ratio, where all regulators are inhibitors except one activator.

In the 1 : 1 ratio scenario, where there is an equal number of activators and inhibitors, the asymptotic behavior of the "AND-NOT" and "OR-NOT" functions corresponds to absolute inhibition (0) and activation (1) respectively, following the biased behavior shown previously. The "Pairs" function behaves similarly to the "OR-NOT" function and therefore is also biased towards 1. Only the threshold functions show balanced behavior with their truth density value reaching asymptotically $1/2$, since the majority rule does favor neither activators nor inhibitors in this scenario. On the other hand, in the two extremely unbalanced scenarios, where one set of regulators completely outweighs the other, the asymptotic truth density results of the "AND-NOT" and "OR-NOT" functions depend on each respective scenario. Specifically, when the inhibitors dominate over the activators, the "OR-NOT" is balanced and the "AND-NOT" is biased, since the former has been shown to depend exclusively on the number of activators (which is just one in this case) for increasingly more regulators, whereas the latter on the number of inhibitors. Their behavior is reversed when the activators outbalance the inhibitors. In contrast, the "Pairs" function behaves in a balanced manner, having a truth density asymptotically equal to $1/2$ in both these scenarios, since the single minority regulator is paired with every regulator from the dominant group in the respective DNF expression (Eq. 3) and as a result, it significantly influences the function's output. Lastly, the asymptotic results for the threshold functions follow the larger size regulatory group, being biased towards 0 with significantly more inhibitors and biased towards 1 with significantly more activators.

5.3 Validation of asymptotic behavior

One key issue of immense practical importance for the modeler, which arises when analyzing the asymptotic behavior of the truth density formulas, is the actual number of regulators that effectively make each of the studied functions exhibit the demonstrated behavior. We noticed that most of the truth density formulas (Eq. 11, 12 and 13) are the sum of two to three terms, with only one of them depending exclusively on the number of regulators n. Also, this term is usually $1/2^n$, and can be omitted when considering values larger than $n = 10$ regulators since it's insignificant ($1/2^{10} \approx 0.001$). This suggests that the limit value of the truth density formulas may be already derived from a much smaller number of regulators than what is implied by the study of asymptotes. Therefore, we need to have a more data-centric view of the results from our previous asymptotics analysis of the different BRFs, one that will enable us to verify the mathematically observed behaviors but also identify an approximate range for the number of regulators where the asymptotics decide the outcome of the studied functions.

We generated the complete truth tables for the five consistent BRFs of Table 2, from 2 up to 20 regulators, accounting for every possible activator-to-inhibitor ratio. For example, for $n = 10$ regulators, every combination of at least one activator and one inhibitor that adds up to 10 (1 activator + 9 inhibitors, 2 activators + 8 inhibitors, etc.) resulted in a different truth table for each considered Boolean function. Subsequently, using the generated truth tables, we could easily calculate the exact truth density value for each function at every considered ratio. The results are shown in Figures 1A and 1B for the link operator and threshold functions, respectively.
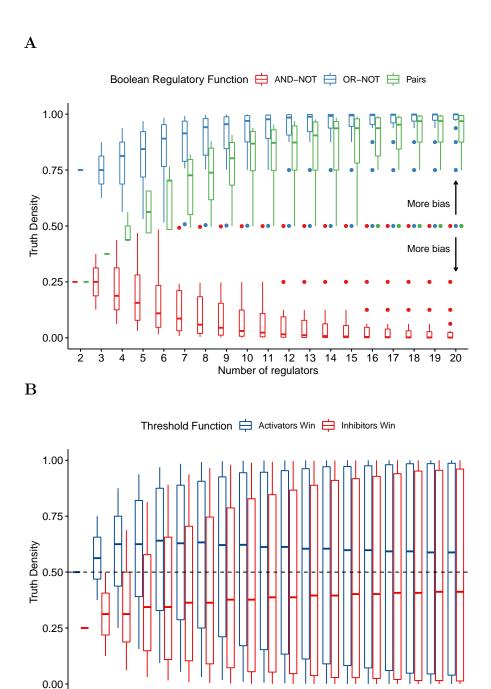
A

B

Figure 1: Comparing the truth densities of five different Boolean regulatory functions for different numbers of regulators and activator-to-inhibitor ratios. For each specific number of regulators, every possible combination of at least one activator and one inhibitor that add up to that number, results in a different truth table output with its corresponding truth density value. All such possible configurations up to 20 regulators are shown. (**A**) The standardized "AND-NOT" function, along with the "OR-NOT" and "Pairs" functions, show an increasingly biased behavior with more regulators. (**B**) The two threshold functions "Act-win" and "Inh-win" show a more balanced behavior, since they respect the activator-to-inhibitor ratio and thus demonstrate a larger spectrum of possible truth density values even for higher numbers of regulators.

The data in general shows that the different regulatory functions demonstrate quite dissimilar behaviors with regard to their asymptotic outcome. In particular, we recapitulate the findings from the asymptotics analysis, namely the bias of the link operator functions, which is evident even from 7 to 10 input regulators. Interestingly, the "Pairs" function follows asymptotically the behavior of the "OR-NOT" function but is in general less biased. We note that the outliers in Figure 1A with truth density values closer to 1/2, represent imbalanced activator-to-inhibitor ratio scenarios, i.e. either considerably more activators than inhibitors for the "AND-NOT" function and the reverse for the "OR-NOT" function, or any imbalanced ratio for the "Pairs" function. Lastly, Figure 1B shows that the threshold functions exhibit a more balanced behavior, expressed as a higher spectrum of truth density values for any single number of regulators and with the median truth density asymptotically reaching 1/2. This result is due to the fact that threshold functions faithfully follow the activator-to-inhibitor ratio, i.e. with more activators the outcome is biased towards 1 whereas with more inhibitors the function outcome tends towards 0.

6 Link operator parameterization determines activity state in biological networks

In this section we investigate if a model's parameterization can effectively decide the activity state of nodes in biological networks. In more detail, we will use the "AND-NOT" link operator function [9] and its symmetric function "OR-NOT" (Eq. 1 and 2), to build Boolean models from prior causal knowledge and check if their activity state profile as determined by dynamic attractor analysis, shows the biased behavior that we observed in Section 5.

A major motivation for this analysis is the fact that the "AND-NOT" function is extensively used by logical modelers and thus the knowledge of its bias, made possible through the lens of the truth density metric, should be clearly demonstrated in practical use cases, e.g. biological network targets should mostly be in an inhibited state when the "AND-NOT" parameterization is used in their respective Boolean equations and in an active state in the case of the "OR-NOT". As such, a modeler could make use of the link operator function bias to select the appropriate model parameterization which statistically guarantees an activity state profile that best matches the one supported by experimental evidence.

6.1 From topology to link operator Boolean models

In order to define Boolean models with the "AND-NOT" and "OR-NOT" link operator parameterization forms, we implemented the software *abmlog*, which stands for "**A**ll possible **B**oolean **M**odels **L**ink **O**perator **G**enerator" (Software and Data Availability). Given a simple interaction (.sif) format file [7], representing a PKN with clearly defined, positive and negative causal interactions, the abmlog software outputs all combinatorially possible Boolean models where each link operator equation (deciding the state of a *link operator node*, i.e. one whose Boolean activity state is determined by both positive and negative regulators) will have either the "AND-NOT" or the "OR-NOT" function form. The models are saved in both the widely-used BoolNet (.bnet) [27] format and the gitsbe format [28], with the latter additionally including the attractors of the Boolean model, calculated via the BioLQM Java library [29]. A simple overview of the software is presented in Figure 2.

By default, abmlog generates all possible Boolean models with the two link operator parameterizations, the number of which depends on the number of link operator nodes. For example, if a network has 12 nodes with both activating and inhibitory regulators, then a total of $2^{12} = 4096$ Boolean models will be generated. In case the number of all possible Boolean models is very large or space restrictions do not allow the storage of that many models, the software can also be used to generate a random sample of link operator Boolean models from the total parameterization space. In summary, abmlog is a useful tool that can generate a large pool of Boolean models for subsequent analyses, each with a unique link operator parameterization.
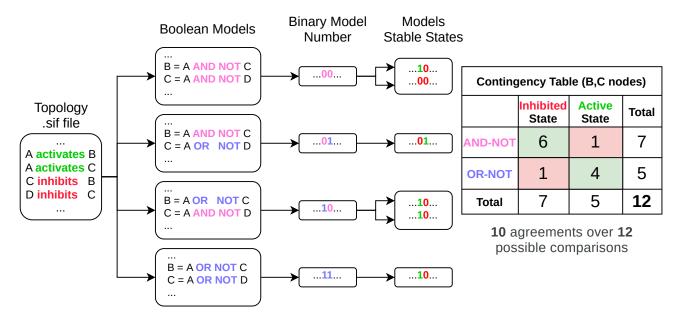
Figure 2: Data-flow overview diagram of the abmlog software and its related contingency table between output model parameterization and stable state activity. A simple interaction file is given as an input to produce a series of Boolean models where equations with both activating and inhibitory regulators have either the "AND-NOT" or the "OR-NOT" formulation. Two link operator equations give rise to a total of $2^2 = 4$ different Boolean models. Each unique parameterization can be represented by a single binary model number, where a "0" corresponds to an equation with the "AND-NOT" link operator and a "1" to an equation with the "OR-NOT". This representation of parameterization can be directly compared to each of the models' stable states, which enables the creation of a contingency table for the data pertaining to nodes B and C and the derivation of measures of agreement (see Section 6.2).

6.2 Measuring agreement between parameterization and stable state

In order to quantify the link operator function bias, we use measures of agreement between parameterization and stable state. The idea is that the more biased the link operator parameterization is, the higher the expected agreement will be between a target node's link operator assignment and its corresponding stable state. For the rest of this work, we shall use two measures of agreement, namely the *percent agreement* and Cohen's *kappa statistic* [30].

In more detail, using the Boolean model data generated by abmlog, we focus in two categorical variables related to a particular node of interest: its link operator parameterization ("AND-NOT"/"0" or "OR-NOT"/"1") and its corresponding stable state activity ("inhibition" or "activation"), obtained via attractor analysis. We shall say that these two variables "agree" when a node whose target Boolean equation has the "AND-NOT" link operator (resp. "OR-NOT") ends up with an inhibited (resp. active) state in the corresponding attractor. In the case of a Boolean model with multiple attractors, each of the stable states is used separately to measure the agreement between the two aforementioned variables, since the activity of a node might change between the different attractors, but its parameterization stays the same.

To define measures of agreement between the two proposed categorical variables, we visualize their interrelation using a contingency table. A total of four data comparison counts can be used to fill in the table's cells: two where the parameterization and stable state match (i.e. node had the "AND-NOT" link operator form and an inhibited stable state or the "OR-NOT" form and an active state) and two where they differ (i.e. node had the "OR-NOT" form and its state was inhibited, or the "AND-NOT" form and an active state). The

percent agreement is then simply defined as the total number of matches divided by the total number of comparisons and is directly interpreted as the percentage of data that the two variables agree upon. In the example of Figure 2, the corresponding contingency table counts all the matches and mismatches between the link operator assignments for nodes B and C and their corresponding activity state (12 comparisons in total). Since there are only two mismatches, the percent agreement is equal to $10/12 = 0.83$, meaning that in 83% of the presented data, the link operator parameterization dictated function outcome. Naturally, a value of 0 is the absolute minimum score and indicates complete disagreement between the two variables while a perfect agreement score is equal to 1 or 100%.

A more robust statistic that we also apply in the Boolean model data is Cohen's kappa (κ) coefficient [30]. This statistic is used to measure the extent to which data collectors (raters) assign the same score to the same variable (inter-rater reliability) and takes into account the possibility of agreement occurring by chance. In our case, this can be conceived as one rater that assigns link operator parameterization ("AND-NOT" or "OR-NOT") and another that assigns stable state activity ("inhibition" or "activation"). Both variables are converted to a binary outcome (0 or 1), allowing the creation of a contingency table and subsequently the calculation of Cohen's formula for κ. The kappa statistic ranges from -1 to $+1$, where a value of 0 represents the amount of agreement that can be expected from random chance, and a value of 1 (resp. -1) indicates perfect agreement (resp. disagreement) between the raters. In the example contingency table of Figure 2, $\kappa = 0.657$, which is a considerable reduction in the level of congruence compared to the 0.83 percent agreement.

6.3 Truth Density bias in biological networks

6.3.1 Bias guides model parameterization in a cancer signaling network

We used abmlog on a cancer signaling network, consisting of 77 nodes and a total of 149 curated causal interactions that cover a variety of pathways linked to prosurvival and antisurvival cell signaling (e.g. cyclin expression and caspase activation). This PKN, named CASCADE (**CA**ncer **S**ignaling **CA**usality **D**atabas**E**), was successfully used to build a Boolean model able to predict anti-cancer drug combination effects in gastric cell lines [14]. We used the CASCADE version from the Flobak paper (version CASCADE 1.0), with some node naming changes for compatibility with the newest versions [31]. The number of nodes with both activating and inhibiting regulators in the CASCADE 1.0 topology is 23, while the rest of the nodes have regulators that belong to only one of the two regulatory categories. Thus, using abmlog, we generated all 2^{23} possible Boolean models with the "AND-NOT" and "OR-NOT" link operator parameterizations. The resulting stable state distribution across all produced models is presented in Figure 3A. For our subsequent analysis we will use only the $2\,802\,224$ Boolean models that had exactly one stable state, as it makes the calculation of agreement between a node's assigned link operator and its corresponding activity state across all the selected models more straightforward.

The agreement results between link operator parameterization and stable state activity across all the selected CASCADE models are presented in Figures 3B (percent agreement, per node) and 3C (Cohen's κ, nodes with the same number of regulators are grouped together). The percent agreement results show a high variability across the link operator nodes and range from a minimum of 53% to a perfect agreement (100%). This suggests that for all nodes, across any selected CASCADE 1.0 Boolean model, there is a higher than random probability that the assignment of the "AND-NOT" (resp. "OR-NOT") link operator formula in the associated Boolean equations will result in the inhibition (resp. activation) of the target nodes. So, even though none of the nodes have more than 5 regulators, we already start seeing signs of the truth density bias in the link operator regulatory functions across a wide range of Boolean models.

When applying Cohen's κ to evaluate level of agreement, we chose a conservative threshold equal to 0.6, corresponding empirically to a substantial level of agreement [32, 33]. We found that 60% (14 out of 23) of

the nodes have a κ value below the specified threshold. Our conclusion is that biological networks with higher in-degree nodes (i.e. more than $7-10$ regulators) are needed to properly assess if there is a truly high level of agreement between Boolean parameterization and function state outcome in the case of the link operator regulatory functions, providing thus conclusive proof of their bias (Section 6.3.2).
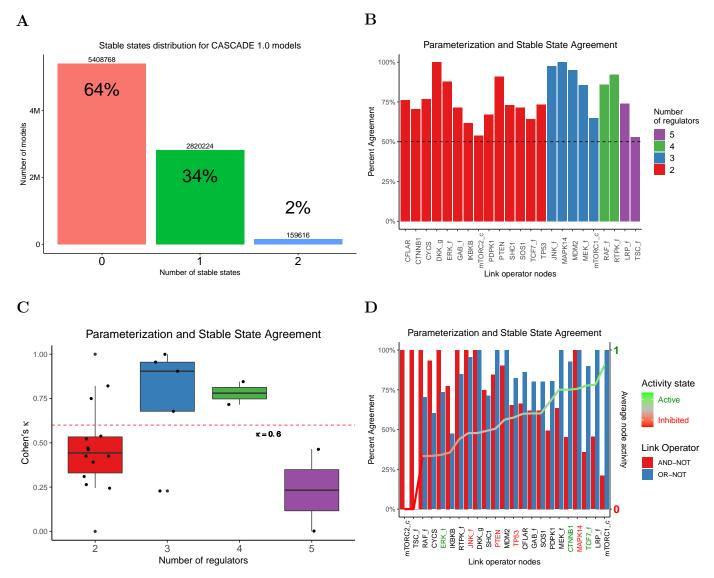

Figure 3: (**A**) Stable states distribution across all link operator parameterized Boolean models generated by the abmlog software using the CASCADE 1.0 signaling topology. (**B**) Percent agreement scores between parameterization and activity state across all single stable state CASCADE 1.0 models, for 23 nodes with both inhibiting and activating regulators. Nodes are sorted according to the total number of input regulators. (**C**) Same as (**B**), with the difference that the link operator nodes are now grouped into categories based on the total number of input regulators and Cohen's κ is used as an agreement statistic. (**D**) Same as (**B**), with the agreement now calculated as the proportion of matches between a node's link operator and its activity state, in the models that had the specific parameterization. The link operator nodes are sorted according to the average activity state across the considered CASCADE models and the colored node labels indicate literature curated activity profiles from Flobak et al. [14]

Regardless of the presence of bias or not, the agreement results can be used to show how experimental data and topological regulatory knowledge (e.g. the activator-to-inhibitor ratio) can be coupled with the truth density metric to guide the choice of regulatory functions. In one example scenario, a modeler asks what the most probable link operator parameterization is among the "AND-NOT" and "OR-NOT" forms that matches available experimental evidence. We used a literature curated activity profile derived for the AGS cell line from [14], to annotate 7 of the link operator nodes in Figure 3D according to their experimentally validated state (activation or inhibition). To clearly identify which of the two parameterizations best fits the observed data, for each node we split the CASCADE models in two model pools, representing the "AND-NOT" and "OR-NOT" node parameterizations, and calculated the proportion of models within each pool whose link operator matched the expected state outcome. For example, in the contingency table of Figure 2, the equivalent calculation would be to divide the number of matches in each row with the corresponding row total sum, resulting in $6/7 = 85.7\%$ of the "AND-NOT" Boolean equations with an inhibited stable state and $4/5 = 80\%$ of the "OR-NOT" equations with an active target node. Moreover, the link operator nodes of Figure 3D are sorted in increasing order by their average stable state activity in the considered CASCADE 1.0 Boolean models. It is evident that nodes with higher average activity in the stable state have a higher agreement with the "OR-NOT" parameterization whereas nodes with lower average activity, a higher agreement with the "AND-NOT" parameterization (0.85 and -0.74 Pearson correlation coefficients with $p_{corr}^{\text{OR-NOT}} = 2.6 \times 10^{-7}$ and $p_{corr}^{\text{AND-NOT}} = 5 \times 10^{-5}$ respectively, see Software and Data Availability).

More specifically, we observe that for all experimentally validated nodes, a modeler could a priori set the link operator to the appropriate form and get a stable state activation profile that matches the observations ("AND-NOT" to match an inhibition node profile or "OR-NOT" for an activation profile) with a higher probability than if he was randomly choosing one of the two. For example, the data shows that 90% of the models with an "OR-NOT" Boolean equation for the target family node `TCF7_f`, had the node as active in their respective stable state. The same is observed for the `CTNNB1` (92%) and `ERK_f` active nodes (74%), as well as for the `TP53` (65%) and `PTEN` (85%) inhibited nodes with the choice of the "AND-NOT" parameterization. Additionally, all the aforementioned nodes have two regulators (one activator and one inhibitor) and using the respective truth density formulas (Eq. 6 and 7) with $n = 2$ and $m = k = 1$, we have that $TD_{AND-NOT} = 0.25$ (closer to 0 or inhibition) and $TD_{OR-NOT} = 0.75$ (closer to 1 or activation), as was also shown in Figure 1A. As such, the nodes observed output matches the statistically expected binary outcomes, showing that even with a low number of regulators, the BRF bias can be used to guide function choice.

In another scenario, a modeler knows that a particular node has a skewed activator-to-inhibitor ratio and wants to exploit such knowledge to make the node conform to a particular activity state of his choice. A nice example from our data is the family node `LRP_f`, with four activators and one inhibitor. Using the truth density formulas for the two link operator parameterizations (Eq. 6 and 7) with $n = 5$, $m = 4$ and $k = 1$, we have that $TD_{AND-NOT} = 0.47$ and $TD_{OR-NOT} = 0.97$. So, if the modeler wants to have an active `LRP_f` in the stable state, the "OR-NOT" parameterization should be preferred since the "AND-NOT" has an approximate 50% probability for this to happen from a statistical point of view. These truth density values also match the results from Figure 3D, since only half of the models that use the "AND-NOT" parameterization end up with an inhibited `LRP_f` in the stable state while all of them have an active `LRP_f` (100% agreement) in the case where the "OR-NOT" form is used. Also, the average activity of `LRP_f` across all models is one of the highest in the data, suggesting that imbalanced activator-to-inhibitor ratios could be a direct proxy for predicting regulation outcome. In a similar situation, but at the other range of the activity spectrum, we have the `TSC_f` family node with one activator and four inhibitors. The truth density values (now using $n = 5$, $m = 1$ and $k = 4$) are $TD_{AND-NOT} = 0.03$ and $TD_{OR-NOT} = 0.53$ respectively. Therefore, the "AND-NOT" parameterization guarantees the inhibition of the `TSC_f` node (data shows 100% agreement) and it should be a modeler's first choice if that is the desired outcome. On the other hand, if the activation of `TSC_f` was a modeler's preference, then the choice of the "OR-NOT" form would be the most

statistically appropriate according to the truth density metric. We observe though that there was no model having `TSC_f` inhibited in the stable stable, indicating that the complex dynamics of the cancer network can also play a significant role in the function outcome. In general, we note that the particular configuration of activating and inhibiting regulators of a target in a specific model instance, can influence the dynamics attributable to the parameterization, causing several results from our analysis to differ from the expected behavior of the Boolean functions studied.

6.3.2 Hub node bias in random scale-free networks

In the previous section we showed that the truth density bias can be used to predict regulatory function outcome in a specific cancer signaling network, but the question still remains open for general biological networks. Also, we found evidence suggesting that Boolean dynamics also plays a significant role in deciding each node's state in the attractors and in some cases activity state results may contradict what is expected from the use and asymptotic interpretation of the truth density formulas. Therefore, we now proceed to investigate if networks with higher in-degree nodes (i.e. more input regulators) have stable states that can be unquestionably decided a priori by the truth density metric, using the respective TD formulas for the "AND-NOT" and "OR-NOT" link operator parameterizations.

We study the specific class of scale-free networks [34], based on the hypothesis that most biological networks exhibit that property, i.e. their node degree distribution follows asymptotically a power law $P(k) \sim k^{-\gamma}$, with k the number of regulators and γ the scale-free exponent. We note that the CASCADE 1.0 model also exhibits the scale-free property (see Software and Data Availability) and there has been evidence in the literature both in favor and against this hypothesis. In particular, earlier studies showed that many complex networks (including metabolic ones) are approximately scale-free [35, 36, 37, 38], whereas more recent efforts demonstrated that not all cellular biological networks may share that property [39], but those that do, exhibit the strongest level of evidence of scale-free structure [40]. Consequently, we shall use scale-free topologies as acceptable substitutes of real biological networks in our analysis.

The methodology is as follows: we start by generating scale-free topology files with a total of 50 nodes each and a maximum in-degree $k_{max} = 50$ [27]. For each network, the number of input regulators per node is drawn from a Riemann Zeta distribution with parameter γ [41]. The choice of regulators for each network node, as well as the type of regulation (positive or negative), is uniformly random. The Zeta distribution allows the creation of in-degree values that far exceed the average connectivity in a network, giving rise to the highest-degree nodes (often called "hubs"), which are the most defining characteristic of the scale-free networks. The value of the scale-free exponent influences the number of hubs and their in-degree distribution. More specifically, we created scale-free networks with $\gamma = 2$ and $\gamma = 2.5$, since most of the studied networks have an exponent between 2 and 3 [41, 42]. Comparing the networks built with the above methodology, we found that those with $\gamma = 2$ have more nodes with both activating and inhibiting regulators and higher degree hubs than networks with $\gamma = 2.5$ (Figures 4A and 4B). These two characteristics suggest that the scale-free networks with $\gamma = 2$ are better suited for use with the abmlog software, since the larger the number of link operator nodes, the more Boolean models can be generated and thus more data comparisons can be made between node parameterization and stable state activity. Additionally, the presence of higher degree hubs is the perfect testbed for the link operator function bias, which manifests especially for nodes with more than $7 - 10$ regulators, as we found from our earlier truth density asymptotics analysis (Figure 1A).

Our methodology proceeds with using each of the scale-free topologies with $\gamma = 2$ as input to the abmlog software, and generating ensembles of Boolean models parameterized with every possible mix of the "AND-NOT" and "OR-NOT" regulatory functions along with the calculation of their stable states (as demonstrated in Figure 2). The produced Boolean models had zero, one, or more stable states. Interestingly, we observed that around half of the tested scale-free topologies generated Boolean models with no stable states, no matter which combination of link operators was used to define the model parameterization. Therefore, the randomly
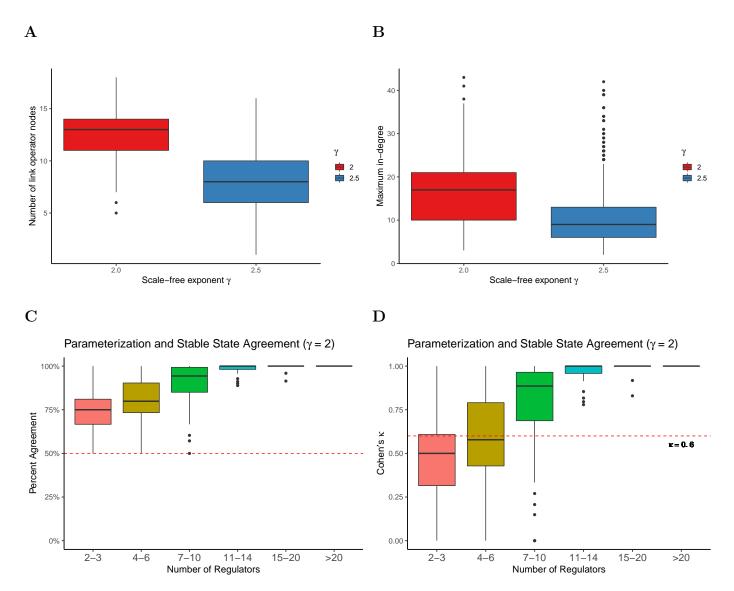
A

B

C

D

Figure 4: (**A**)-(**B**) Network statistics for scale-free topologies with different degree exponents. Every network tested has 50 nodes and a maximum in-degree $k_{max} = 50$. A total of 100 topologies for $\gamma = 2$ and 1000 topologies for $\gamma = 2.5$ are compared. Networks with $\gamma = 2$ have a higher median number of nodes with both activating and inhibiting regulators and higher degree hubs. (**C**)-(**D**) Agreement statistics between link operator parameterization and stable state activity. The data is taken from Boolean models generated with the abmlog software, using scale-free topologies with exponent $\gamma = 2$. A total of 757 link operator nodes were compared across multiple link operator parameterization configurations with their corresponding stable states. Nodes are grouped in buckets, where each bucket indicates a different range of input regulators. Both the percent agreement and Cohen's κ show considerable congruence between link operator assignment ("AND-NOT" or "OR-NOT") and resulting stable state (inhibition or activation respectively) for nodes with more than 10 input regulators.

assigned regulators, regulatory effects, and Zeta distribution in-degree values, may result in networks which do not have stable phenotypes, suggesting that alternative parameterizations might be more suitable in modeling scenarios which specifically examine stable dynamics. Nonetheless, we discarded the models with no stable states and used the rest that had single or multiple attractors in our analysis. Then, for each model

node with both activating and inhibiting regulators, we compared its assigned link operator with the activity state value in the corresponding stable state(s), across all the link operator parameterization spectrum that yielded models with stable phenotypes. The agreement results between parameterization and stable state activity are presented in Figure 4C for the percent agreement and in Figure 4D for Cohen's kappa statistic.

We observe that both presented statistics show a large variation of agreement for nodes with less than 10 regulators and an increasing agreement with more regulators. This agreement manifests in link operator nodes parameterized with the "AND-NOT" or "OR-NOT" Boolean functions, while at the same time exhibiting inhibited or active states respectively in the associated model attractors. Therefore, we conclude that the considered standardized Boolean regulatory functions are biased and their outcomes can be determined a priori from the choice of the corresponding link operator parameterization, especially for nodes with more than $7 - 10$ regulators.

7   Discussion

The specification of mathematical rules that describe the behavior of biological systems is one of the core aspects of computational modeling. It is therefore of considerable value to have a list of metrics that can be used to compare different model parameterizations and make an informed decision with regard to the selection of an appropriate regulatory function that better matches the expected behavior in a specific modeling application.

We specifically discussed two characterizations that can assist modelers in comparing various regulatory functions and select the most plausible ones with regard to the causal interaction-based knowledge at hand. Expressing Boolean functions in DNF makes biological interpretation concrete by explicitly specifying the conditions (presence or absence of the positive and negative regulators, respectively) that make a target active. Expressing the functions in CDNF allows to easily check for compliance with the underlying regulatory topology and subsequently, the rejection of functions that violate such consistency. The difference between these two characterizations lies in the fact that the consistency terminology stems from the mathematical world, while biological interpretability is tightly connected to the world of language semantics and thus closer to the modeler's point of view. Finally, truth density is an informative measure which can be used to verify if the function parameterization dictates biased Boolean outcomes. It can also be used as a test metric to understand how a function behaves when the number of regulators increases or the balance between the number of activators and inhibitors changes.

Using the truth density metric, we showed the presence of link operator function bias in the hubs of randomly constructed scale-free networks. A potential application of this finding could be to dramatically decrease the time needed to train Boolean models to fit observations via various optimization methods, by pre-assigning the parameterization of link operator nodes with sufficiently many regulators. The pruning of the searchable parameterization space, guided by the truth density metric, can result in more efficient automated methods and can enable the training of larger models against data from numerous resources (e.g. large cell line panels). The hub node bias has also interesting links to the presence of order in biological networks [43]. The dynamics of a Boolean network can exhibit ordered or chaotic behavior. Ordered dynamics is characterized by the presence of less stable states and limit cycle attractors with smaller mean length (number of states in a complex attractor) and transition times (number of steps needed to reach an attractor starting out from an arbitrary configuration) [41]. It is also known that the truth density (probability of target expression) as well as the degree exponent γ (related to network connectivity) can modulate the dynamic transition between the ordered and chaotic phases. Moreover, it has been shown that above the critical value of $\gamma_c \sim 2.47$, ordered behavior in the form of stable state dynamics manifests independently of the truth density, whereas for values closer to $\gamma = 2$, order coincides with the presence of high biased nodes (see Fig. 4 in [41]). Our work confirms this phenomenon, since the use of the link operator parameterization guarantees the presence of biased hubs,

which enable the scale-free networks to exhibit stability and homogeneity in terms of regulatory output, and thus stay in the ordered dynamic regime.

Searching for other function metrics that are applicable to logical modeling, the *sensitivity* of a Boolean function is one of the most relevant [44]. As its name suggests, it measures how sensitive the output of the function is to small changes of its inputs. Sensitivity is tightly linked to the truth density metric, since a highly homogeneous Boolean function (i.e. a biased one), is unlikely to change its value between similar regulatory input configurations and so, its sensitivity is relatively low. To compute the average sensitivity value for an arbitrary Boolean function we need to sum over all the *influences* of the input variables, which essentially represent a way to measure individual variable importance. In the context of regulatory functions, a regulator's influence is defined as the probability that a random toggle on its activity (from active to inactive and vice-versa) will change the value of the Boolean function [45]. Therefore, by calculating the influence of every regulator, the modeler can gain knowledge of which ones are more important and control the respective function's outcome. This transition of perspective from the function level to the regulator level might be advantageous in cases where the modeler's intention is to compare different parameterizations and choose the one for which a particular regulator is labeled as significantly more important than the others, based on the available biological knowledge.

Lastly, an important addition to a universal list of Boolean function metrics for modeling purposes, is the notion of function *complexity*. A recent definition is given by Gherardi et al. [25], where the authors defined it as the number of terms in the shortest possible DNF expression of a given Boolean function, divided by the total number of rows in the corresponding truth table. We presented this information in the last column of Table 2, where the BRFs are sorted from lower to higher complexity (note that the CDNF has the minimum number of terms for every BRF included in the table). One useful observation is that the standardized "AND-NOT" formula [9] is the function with the lowest complexity that is also consistent and thus biologically plausible - all properties that make it a good choice from the modeler's perspective. Assessing the complexity of the studied regulatory functions using the derived formulas for the minimum number of CDNF terms for any number of activators m and inhibitors k (see last column of Table 2), we comment on the fact that all BRFs have very low complexity since $\mathcal{O}(m \times k) \ll 2^{m+k}$, i.e. the number of function terms does not grow as fast as the number of rows in the corresponding truth table. Same observation has been shown to be true in manually-tuned, experimentally-validated Boolean functions [25], providing us with another confirmation that the consistent functions from Table 2 are good candidates for logic-based modeling approaches.

8 Future work

In this work we make an attempt to address the logical rule specification problem, which can be simply stated as: "Many functions may fit the available observations, which one is the most proper to use?" Of course what is "proper" can be fairly subjective, but the main point is that a careful consideration of the underlying application context (i.e. what output do I expect in a specific scenario of interest) along with a list of metrics that explicate a Boolean function's behavior and semantics, provides the user with the appropriate framework to decide on the function parameterization that sets the basis for further model analysis and simulation. In that regard, interesting directions for further research include the application of the metrics presented in this work in different published biological models, and the subsequent comparison of different regulatory functions within this framework. Such meta-analyses could potentially indicate regulatory functions that achieve a higher degree of fitness with the observed data or general properties that are common in all Boolean functions used to model biological systems.

An interesting study for example would be to analyze Boolean functions from published biological models that have extreme activator-to-inhibitor ratios. If such imbalanced ratios also result in proportionally skewed Boolean outcomes (i.e. with more activators, the truth density is closer to 1 and the reverse with more inhibitors), suggesting that target outcome follows the majority regulatory groups, then the use of threshold

functions could be a more proper parameterization alternative, as was shown in Figure 1B. Of course, we note that each individual case must be examined with care, since there might be high influence nodes, whose activity defines the target's output even in the presence of a much larger regulatory group with opposite effects. For example, `CASP3` is a biological entity that, when activated, will almost certainly result in the cell's death even in the presence of a majority of proliferation-positive regulators at any given time. Subsequently, a more appropriate choice based on the results of this study can be made, either by choosing between the biased functions, which demonstrate a more balanced behavior for such extreme activator-to-inhibitor ratios (e.g. using the "Pairs" or the "AND-NOT" functions which are balanced vs using the "OR-NOT" which would make the target activated most of the time, see Scenario 2) or by using refined threshold functions, in which each regulator's weight will differ in order to match the influence that it has on the target.

There have been only a handful examples of published logical models [46, 47] and research papers [21, 48, 49, 50, 51, 52, 53, 54] that use the threshold modeling framework in biological systems. This is partly due to the lack of tools that make threshold functions accessible to the average user, and the availability of such software in open-source environments such as the CoLoMoTo Interactive Notebook [55]. We believe that the existence of such novel software will enable the construction and configuration of generic Boolean threshold models and provide users of the logical-modeling community and beyond with the necessary toolbox to further study these models. This will enable applications that depend on the dynamical analysis of Boolean threshold models (identification of attractors, reachability properties, formal verification and control) and the use of optimization methods to calibrate the threshold function parameters to best fit the available experimental data, as is done currently with analytical logic-based functions [12].

Software and Data Availability

The *abmlog* software that was used to generate Boolean models with the "AND-NOT" and "OR-NOT" Boolean regulatory functions is available at https://github.com/druglogics/abmlog under the MIT License. We used the version 1.6.0 for this analysis, which is also offered as a standalone package at https://github.com/druglogics/abmlog/packages.

An extended analysis accompanying the results of this paper is available at https://druglogics.github.io/brf-bias. It includes links to the produced model datasets and scripts to reproduce the results and figures of this paper. In particular, the correlation analysis between average node state in the CASCADE 1.0 models and percent agreement per each link operator is available at https://druglogics.github.io/brf-bias/cascade-1-0-data-analysis.html#node-state-and-percent-agreement-correlation. The degree distribution of the CASCADE 1.0 topology and other network statistics are examined in https://druglogics.github.io/brf-bias/cascade-1-0-data-analysis.html#network-properties.

Funding

Acknowledgments

A   Truth Density formula proofs

For all the following propositions, we consider f to be a Boolean regulatory function $f_{BRF} : \{0,1\}^n \rightarrow \{0,1\}$, with a total of n input regulators separated to two distinct sets, the set of $m \geq 1$ activators $x = \{x_i\}_{i=1}^m$ and the set of $k \geq 1$ inhibitors $y = \{y_j\}_{j=1}^k$, such that $n = m + k$.

Proposition 1 ("AND-NOT" Truth Density). *The truth density of the "AND-NOT" link operator function* $f_{AND-NOT}(x, y) = (\bigvee_{i=1}^m x_i) \wedge \neg \left(\bigvee_{j=1}^k y_j \right)$, *with $m \geq 1$ activators and $k \geq 1$ inhibitors, is given by the formula:*

$$TD_{AND-NOT} = \frac{2^m - 1}{2^n} = \frac{1}{2^k} - \frac{1}{2^n} \tag{6}$$

Proof. Using the distributive property and De Morgan's law we can express $f_{AND-NOT}$ (Eq. 1) in the equivalent DNF:

$$
\begin{aligned}
f_{AND-NOT}(x, y) &= \left(\bigvee_{i=1}^m x_i \right) \wedge \neg \left(\bigvee_{j=1}^k y_j \right) \\
&= \bigvee_{i=1}^m \left(x_i \wedge \neg \left(\bigvee_{j=1}^k y_j \right) \right) \\
&= \bigvee_{i=1}^m \left(x_i \wedge \bigwedge_{j=1}^k \neg y_j \right) \\
&= \bigvee_{i=1}^m \left(x_i \wedge \neg y_1 \wedge ... \wedge \neg y_k \right)
\end{aligned}
$$

To calculate $TD_{AND-NOT}$, we need to find the number of rows in $f_{AND-NOT}$'s truth table that result in a *True* output result and divide that by the total number of rows, which is 2^n (n input regulators).

Note that $f_{AND-NOT}$, written in it's equivalent DNF, has exactly m terms. Each term has a unique *True/False* assignment of regulators that makes it *True*. This happens when the activator of the term is *True* and all of the inhibitors *False*. Since the condition for the inhibitors is the same regardless of the term we are examining and f is expressed in DNF, the *True* outcomes of the function f are defined by all logical assignment combinations of the m activators that have at least one of them being *True* and all inhibitors assigned as *False*. There are a total of 2^m possible *True/False* logical assignments of the m activators (from all *False* to all *True*) and $f_{AND-NOT}$ becomes *True* on all except one of them (i.e. when all activators are *False*), with the corresponding $2^m - 1$ truth table rows having all inhibitors assigned as *False*. Therefore, $TD_{AND-NOT} = (2^m - 1)/2^n$. ∎

Proposition 2 ("OR-NOT" Truth Density). *The truth density of the "OR-NOT" link operator function* $f_{OR-NOT}(x, y) = (\bigvee_{i=1}^m x_i) \vee \neg \left(\bigvee_{j=1}^k y_j \right)$, *with $m \geq 1$ activators and $k \geq 1$ inhibitors, is given by the formula:*

$$TD_{OR-NOT} = \frac{2^n - (2^k - 1)}{2^n} = 1 - \frac{1}{2^m} + \frac{1}{2^n} \tag{7}$$

Proof. Using De Morgan's law we can express f_{OR-NOT} (Eq. 2) in the equivalent DNF:

$$f_{OR-NOT}(x,y) = \left(\bigvee_{i=1}^{m} x_i \right) \vee \neg \left(\bigvee_{j=1}^{k} y_j \right)$$

$$= \left(\bigvee_{i=1}^{m} x_i \right) \vee \left(\bigwedge_{j=1}^{k} \neg y_j \right)$$

$$= x_1 \vee x_2 \vee ... \vee x_m \vee (\neg y_1 \wedge ... \wedge \neg y_k)$$

To calculate TD_{OR-NOT}, we find the number of rows of f_{OR-NOT}'s truth table that result in a *False* output (R_{false}), subtract that number from the total number of rows (2^n) to get the rows that result in f being *True*, and then divide by the total number of rows. As such, $TD_{OR-NOT} = (2^n - R_{false})/2^n$.

Note that f_{OR-NOT}, expressed in it's equivalent DNF, has exactly $m+1$ terms. To make f_{OR-NOT} *False*, we assign the m activators as *False* and then we investigate which logical assignments of the inhibitors $\{y_j\}_{j=1}^{k}$ make the last DNF term also *False*. Out of all the possible 2^k *True/False* logical assignments of the k inhibitors (ranging from all *False* to all *True*) there is only one that does not make the last term of f_{OR-NOT} *False*, which happens specifically when all k inhibitors are *False*. Therefore, $R_{false} = 2^k - 1$ and $TD_{OR-NOT} = (2^n - (2^k - 1))/2^n$. □

Proposition 3 ("Pairs" Truth Density). *The truth density of the "Pairs" link operator function* $f_{Pairs}(x,y) = \bigvee_{\forall(i,j)}^{m,k} (x_i \wedge \neg y_j)$, *with* $m \geq 1$ *activators and* $k \geq 1$ *inhibitors, is given by the formula:*

$$TD_{Pairs} = \frac{(2^m - 1)(2^k - 1)}{2^n} \tag{8}$$

Proof. Using the distributive property we can express f_{Pairs} (Eq. 3) in its equivalent conjunction normal form (CNF), where two separate clauses are connected with AND's ($\wedge$) and inside the clauses the literals are connected with OR's ($\vee$):

$$f_{Pairs}(x,y) = \bigvee_{\forall(i,j)}^{m,k} (x_i \wedge \neg y_j) = \left(\bigvee_{i=1}^{m} x_i \right) \wedge \left(\bigvee_{j=1}^{k} \neg y_j \right) \tag{9}$$

To calculate TD_{Pairs}, based on its given CNF, we find the number of rows in its truth table that have at least one *True* activator (R_{act}) and subtract from these the rows in which all inhibitors are *True* (R_{inh}). Therefore, only the rows that have at least one *True* activator and at least one *False* inhibitor will be left, corresponding to the biological interpretation of f_{Pairs}. As such, $TD_{Pairs} = (R_{act} - R_{inh})/2^n$.

R_{act} can be found by subtracting from the total number of rows (2^n), the rows that have all activators as *False*. The number of these rows depends on the number of inhibitors, since for each one of the total possible 2^k *True/False* logical assignments of the k inhibitors (ranging from all *False* to all *True*), there will be a row in the truth table with all activators as *False*. Therefore, $R_{act} = 2^n - 2^k = 2^{m+k} - 2^k = 2^k(2^m - 1)$.

R_{inh} depends on the number of activators, since for each one of the total possible 2^m *True/False* logical assignments of the m activators (ranging from all *False* to all *True*), there will be a row in the truth table with all inhibitors as *True*. Note that we have to exclude one row from this result, which is exactly the row that has all activators as *False* since it's not included in the R_{act} rows. Therefore, $R_{inh} = 2^m - 1$ and $TD_{Pairs} = (R_{act} - R_{inh})/2^n = (2^k(2^m - 1) - (2^m - 1))/2^n$. □

Proposition 4 (Threshold functions Truth Density). *The truth density of the Boolean threshold functions "Act-win" (Eq. 4) and "Inh-win" (Eq. 5), with $m \geq 1$ activators and $k \geq 1$ inhibitors, is given by the formula:*

$$TD_{thres} = \frac{\sum_{i=1}^{m} \left[\binom{m}{i} \times \sum_{j=0}^{min(u,k)} \binom{k}{j} \right]}{2^n} \tag{10}$$

where $u = i$ or $i - 1$, depending on the use of the "Act-win" or "Inh-win" function respectively.

Proof. The truth density formula can be easily derived from the observation that we need to count the number of rows in the respective truth table that have more $True$ activators than $True$ inhibitors. In the case of the "Act-win" function, we also need to add the rows that have an equal number of $True$ regulators in each respective category.

Firstly, we count all the subset input configurations that have up to m activators assigned to $True$. These include the partial $True/False$ logical assignments that have either a single $True$ activator, a pair of $True$ activators, a triplet, etc. This is exactly the term $\sum_{i=1}^{m} \binom{m}{i}$. Note that each of these activator input configurations is multiplied by a factor of 2^k in the truth table to make *complete* rows, i.e. rows where the activators logical assignments stay unchanged and the inhibitor values range from all $False$ to all $True$. Therefore, we need to specify exactly which inhibitor logical assignments are appropriate for each activator subset input configuration. To do that, we multiply the size of each activator subset $\binom{m}{i}$ with the number of configurations that have less $True$ inhibitors, i.e. $\sum_{j=0}^{i-1} \binom{k}{j}$.

Let's consider an example with $m, k > 2$ and set $i = 2$. We find that the number of subsets with 2 $True$ activators is $\binom{m}{2}$. Next, we multiply by the number of configurations that have one or no $True$ inhibitors, i.e. $\sum_{j=0}^{1} \binom{k}{j}$. This results in the number of rows of interest for the "Inh-win" function, i.e. the rows where there are exactly 2 activators assigned to $True$ and less than 2 $True$ inhibitors. For "Act-win", we have to multiply up to the $True$ inhibitor pairs, i.e. $\sum_{j=0}^{2} \binom{k}{j}$. In summation, we count the configurations that have exactly i out of m activators assigned to $True$, and for each one, we multiply by the number of cases that have 0 up to i inhibitors assigned to $True$ to find the respective rows, i.e. $\binom{m}{i} \times \sum_{j=0}^{i} \binom{k}{j}$. Repeating this calculation for every possible subset of i activators (from 1 up to all m of them), and summing the rows up, will result in the numerator of the TD_{thres} formula for the "Act-win" function.

Lastly, note that the *largest* inhibitor configuration subset size that we consider, is the minimum value between the current activator subset size ($u = i$ or $i - 1$, depending on which threshold function we use) and the total number of inhibitors k. Therefore, we take into account the case where the number of inhibitors is less than the activator subset size, i.e. $k < u$. This explains the term $min(u, k)$ in the truth density formula and concludes the proof. $\square$

B Truth Density asymptotic behavior

We study the asymptotic behavior of the four truth density formulas (Appendix A) for a large number of regulators ($n \to \infty$). Note that for the calculations involving the two threshold functions, we will only use the truth density formula corresponding to the "Act-win" function (Eq. 10, with $u = i$), since both functions have similar formulas and therefore, their limiting behavior is analogous. The asymptotics results for each regulatory function are as follows:

1. The "AND-NOT" function truth density (Eq. 6) depends only on the number of inhibitors k:

$$TD_{AND-NOT} = \frac{1}{2^k} - \frac{1}{2^n} \sim \frac{1}{2^k} \tag{11}$$

   For large k, it is biased towards 0:

$$TD_{AND-NOT} = \frac{1}{2^k} \xrightarrow{k \to \infty} 0$$

2. The "OR-NOT" function truth density (Eq. 7) depends only on the number of activators m:

$$TD_{OR-NOT} = 1 - \frac{1}{2^m} + \frac{1}{2^n} \sim 1 - \frac{1}{2^m} \tag{12}$$

For large m, it is biased towards 1:

$$TD_{OR-NOT} = 1 - \frac{1}{2^m} \xrightarrow{m \to \infty} 1$$

3. The "Pairs" function truth density (Eq. 8) depends on both activators and inhibitors:

$$TD_{Pairs} = \frac{(2^m - 1)(2^k - 1)}{2^n} = \frac{2^n - 2^m - 2^k + 1}{2^n} = 1 - \frac{2^m + 2^k}{2^n} + \frac{1}{2^n} \sim 1 - \frac{1}{2^k} - \frac{1}{2^m} \tag{13}$$

4. The threshold functions truth density (Eq. 10) depends on both m and k variables and does not have a single fixed limit for $n \to \infty$.

We now focus on the effect of the ratio $(m : k)$ between number of activators and inhibitors on the asymptotic truth density values for $n \to \infty$. We consider the following three scenarios for each of the Boolean functions:

Scenario 1 A $1 : 1$ activator-to-inhibitor ratio, where approximately half of the regulators are activators and half are inhibitors, i.e. $m \approx k \approx n/2$ (consider n is even without loss of generality).

1. The "AND-NOT" function truth density is biased towards 0:

$$(\text{Eq. } 11) \Rightarrow TD_{AND-NOT} \sim \frac{1}{2^{n/2}} \xrightarrow{n \to \infty} 0$$

2. The "OR-NOT" function truth density is biased towards 1:

$$(\text{Eq. } 12) \Rightarrow TD_{OR-NOT} \sim 1 - \frac{1}{2^{n/2}} \xrightarrow{n \to \infty} 1$$

3. The "Pairs" function truth density is biased towards 1:

$$(\text{Eq. } 13) \Rightarrow TD_{Pairs} \sim 1 - \frac{1}{2^{n/2}} - \frac{1}{2^{n/2}} \xrightarrow{n \to \infty} 1$$

4. The threshold functions truth density is balanced, meaning its limit asymptotically approaches $1/2$.

Proof. We first rewrite the truth density formula substituting $m = k = n/2$:

$$(\text{Eq. } 10) \Rightarrow TD_{thres} = \frac{\sum_{i=1}^{n/2}\left[\binom{n/2}{i} \times \sum_{j=0}^{min(i,n/2)}\binom{n/2}{j}\right]}{2^n} = \frac{\sum_{i=1}^{n/2}\left[\binom{n/2}{i} \times \sum_{j=0}^{i}\binom{n/2}{j}\right]}{2^n} = \frac{N}{2^n}$$

Next we simplify N, by using the notation $z = n/2$ and $\boldsymbol{x}$ as a meta-symbol for $\binom{z}{x}$. For example, $\binom{n/2}{1} = \binom{z}{1} = \mathbf{1}$. N is therefore expressed as:

$$N = \mathbf{1}(\mathbf{0} + \mathbf{1}) + \mathbf{2}(\mathbf{0} + \mathbf{1} + \mathbf{2}) + ... + \boldsymbol{z}(\mathbf{0} + \mathbf{1}... + \boldsymbol{z})$$

Using the symmetry of binomial coefficients: $\binom{z}{x} = \binom{z}{z-x} \sim \boldsymbol{x} = \boldsymbol{z} - \boldsymbol{x}$, we can re-write N as:

$$N = (\boldsymbol{z} - \mathbf{1})[\boldsymbol{z} + (\boldsymbol{z} - \mathbf{1})] + (\boldsymbol{z} - \mathbf{2})[\boldsymbol{z} + (\boldsymbol{z} - \mathbf{1}) + (\boldsymbol{z} - \mathbf{2})] + ... + \mathbf{0}[\boldsymbol{z} + ... + \mathbf{0}]$$

Adding the two expressions for N we have that:

$$2N = [\mathbf{0} + \mathbf{1}... + \mathbf{z}]^2 + \mathbf{1}^2 + \mathbf{2}^2 + ... + (\mathbf{z} - \mathbf{1})^2 = 2^{2z} + \sum_{\mathbf{x}=\mathbf{1}}^{z-1} \mathbf{x}^2$$

Substituting back $\binom{z}{x} = \boldsymbol{x}$ and $i = x$ (change of index) in expression N, we have that the threshold functions truth density is written as:

$$TD_{thres} = \frac{N}{2^{2z}} = \frac{(1/2)\left[2^{2z} + \sum_{i=1}^{z-1}\binom{z}{i}^2\right]}{2^{2z}}$$

As $n \to \infty$ (and hence $z \to \infty$), the term $\sum_{i=1}^{z-1}\binom{z}{i}^2$ does not grow as fast as 2^{2z} - it is smaller by a factor of $\sqrt{\pi z}$ (see answer to Problem 9.18 in [56]), and so it becomes negligible:

$$\lim_{z\to\infty} TD_{thres} = \lim_{z\to\infty} \frac{(1/2)2^{2z}}{2^{2z}} = \frac{1}{2}$$

$\square$

Scenario 2   A high activator-to-inhibitor ratio $(n - 1 : 1)$, where all regulators are activators except one inhibitor, i.e. $m = n - 1, k = 1$.

1. The "AND-NOT" function truth density is balanced:
$$(\text{Eq. 11}) \Rightarrow TD_{AND-NOT} \sim \frac{1}{2^1} = \frac{1}{2}$$

2. The "OR-NOT" function truth density is biased towards 1:
$$(\text{Eq. 12}) \Rightarrow TD_{OR-NOT} \sim 1 - \frac{1}{2^{n-1}} \xrightarrow{n\to\infty} 1$$

3. The "Pairs" function truth density is balanced:
$$(\text{Eq. 13}) \Rightarrow TD_{Pairs} \sim 1 - \frac{1}{2^1} - \frac{1}{2^{n-1}} \xrightarrow{n\to\infty} \frac{1}{2}$$

4. The threshold functions truth density is biased towards 1:
$$(\text{Eq. 10}) \Rightarrow TD_{thres} = \frac{\sum_{i=1}^{n-1}\left[\binom{n-1}{i} \times \sum_{j=0}^{min(i,1)}\binom{1}{j}\right]}{2^n} = \frac{\sum_{i=1}^{n-1}\left[\binom{n-1}{i} \times \sum_{j=0}^{1}\binom{1}{j}\right]}{2^n}$$
$$= \frac{\sum_{i=1}^{n-1}\binom{n-1}{i} \times 2}{2^n} = \frac{2^{n-1} - 1}{2^{n-1}} = 1 - \frac{1}{2^{n-1}} \xrightarrow{n\to\infty} 1$$

Scenario 3   A low activator-to-inhibitor ratio $(1 : n - 1)$, where all regulators are inhibitors except one activator, i.e. $m = 1, k = n - 1$.

1. The "AND-NOT" function truth density is biased towards 0:
$$(\text{Eq. 11}) \Rightarrow TD_{AND-NOT} \sim \frac{1}{2^{n-1}} \xrightarrow{n\to\infty} 0$$

2. The "OR-NOT" function truth density is balanced:
$$(\text{Eq. 12}) \Rightarrow TD_{OR-NOT} \sim 1 - \frac{1}{2^1} = \frac{1}{2}$$

3. The "Pairs" function truth density is balanced:
$$(\text{Eq. 13}) \Rightarrow TD_{Pairs} \sim 1 - \frac{1}{2^{n-1}} - \frac{1}{2^1} \xrightarrow{n\to\infty} \frac{1}{2}$$

4. The threshold functions truth density is biased towards 0:
$$(\text{Eq. 10}) \Rightarrow TD_{thres} = \frac{\sum_{i=1}^{1}\left[\binom{1}{i} \times \sum_{j=0}^{min(i,n-1)}\binom{n-1}{j}\right]}{2^n} = \frac{\sum_{j=0}^{min(1,n-1)}\binom{n-1}{j}}{2^n}$$
$$= \frac{\sum_{j=0}^{1}\binom{n-1}{j}}{2^n} = \frac{1 + (n-1)}{2^n} = \frac{n}{2^n} \xrightarrow[\text{L'Hôpital Rule}]{n\to\infty} 0$$

References

[1] Hiroaki Kitano. Computational systems biology. *Nature*, 420(6912):206–210, Nov 2002. ISSN 00280836. doi: 10.1038/nature01254.

[2] Han-Yu Chuang, Matan Hofree, and Trey Ideker. A Decade of Systems Biology. *Annual Review of Cell and Developmental Biology*, 26(1):721–744, Nov 2010. ISSN 1081-0706. doi: 10.1146/annurev-cellbio-100109-104122.

[3] Rolf Apweiler, Tim Beissbarth, Michael R Berthold, Nils Blüthgen, Yvonne Burmeister, Olaf Dammann, Andreas Deutsch, Friedrich Feuerhake, Andre Franke, Jan Hasenauer, Steve Hoffmann, Thomas Höfer, Peter LM Jansen, Lars Kaderali, Ursula Klingmüller, Ina Koch, Oliver Kohlbacher, Lars Kuepfer, Frank Lammert, Dieter Maier, Nico Pfeifer, Nicole Radde, Markus Rehm, Ingo Roeder, Julio Saez-Rodriguez, Ulrich Sax, Bernd Schmeck, Andreas Schuppert, Bernd Seilheimer, Fabian J Theis, Julio Vera, and Olaf Wolkenhauer. Whither systems medicine? *Experimental & Molecular Medicine*, 50(3):e453, Mar 2018. ISSN 2092-6413. doi: 10.1038/emm.2017.290.

[4] Bree B. Aldridge, John M. Burke, Douglas A. Lauffenburger, and Peter K. Sorger. Physicochemical modelling of cell signalling pathways. *Nature Cell Biology*, 8(11):1195–1203, Nov 2006. ISSN 1465-7392. doi: 10.1038/ncb1497.

[5] Melody K. Morris, Julio Saez-Rodriguez, Peter K. Sorger, and Douglas A. Lauffenburger. Logic-based models for the analysis of cell signaling networks. *Biochemistry*, 49(15):3216–3224, Apr 2010. ISSN 00062960. doi: 10.1021/bi902202q.

[6] Pauline Traynard, Luis Tobalina, Federica Eduati, Laurence Calzone, and Julio Saez-Rodriguez. Logic Modeling in Quantitative Systems Pharmacology. *CPT: Pharmacometrics & Systems Pharmacology*, 6 (8):499–511, Aug 2017. ISSN 21638306. doi: 10.1002/psp4.12225.

[7] Vasundra Touré, Åsmund Flobak, Steven Vercruysse, Anna Niarakis, and Martin Kuiper. The status of causality in biological databases: data resources and data retrieval possibilities to support logical modeling. *Briefings in Bioinformatics*, 2020. doi: 10.1093/bib/bbaa390.

[8] Rui-Sheng Wang, Assieh Saadatpour, and Réka Albert. Boolean modeling in systems biology: an overview of methodology and applications. *Physical Biology*, 9(5):55001, Sep 2012. doi: 10.1088/1478-3975/9/5/055001.

[9] Luis Mendoza and Ioannis Xenarios. A method for the generation of standardized qualitative dynamical systems of regulatory networks. *Theoretical Biology and Medical Modelling*, 3(1):13, Mar 2006. ISSN 17424682. doi: 10.1186/1742-4682-3-13.

[10] Julio Saez-Rodriguez, Leonidas G Alexopoulos, Jonathan Epperlein, Regina Samaga, Douglas A Lauffenburger, Steffen Klamt, and Peter K Sorger. Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction. *Molecular Systems Biology*, 5(1): 331, Jan 2009. ISSN 1744-4292. doi: 10.1038/msb.2009.87.

[11] Santiago Videla, Julio Saez-Rodriguez, Carito Guziolowski, and Anne Siegel. caspo: a toolbox for automated reasoning on the response of logical signaling networks families. *Bioinformatics*, 33(6): 947–950, 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw738.

[12] Enio Gjerga, Panuwat Trairatphisan, Attila Gabor, Hermann Koch, Celine Chevalier, Franceco Ceccarelli, Aurelien Dugourd, Alexander Mitsos, and Julio Saez-Rodriguez. Converting networks to predictive logic models from perturbation signalling data with CellNOpt. *Bioinformatics*, 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa561.

[13] Sara Sadat Aghamiri, Vidisha Singh, Aurélien Naldi, Tomáš Helikar, Sylvain Soliman, and Anna Niarakis. Automated inference of Boolean models from molecular interaction maps using CaSQ. *Bioinformatics*, 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa484.

[14] Åsmund Flobak, Anaïs Baudot, Elisabeth Remy, Liv Thommesen, Denis Thieffry, Martin Kuiper, and Astrid Lægreid. Discovery of Drug Synergies in Gastric Cancer Cells Predicted by Logical Modeling. *PLOS Computational Biology*, 11(8):e1004426, Aug 2015. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004426.

[15] Barbara Niederdorfer, Vasundra Touré, Miguel Vazquez, Liv Thommesen, Martin Kuiper, Astrid Lægreid, and Åsmund Flobak. Strategies to Enhance Logic Modeling-Based Cell Line-Specific Drug Synergy Prediction. *Frontiers in Physiology*, 11:862, Jul 2020. ISSN 1664-042X. doi: 10.3389/fphys.2020.00862.

[16] Amel Bekkar, Anne Estreicher, Anne Niknejad, Cristina Casals-Casas, Alan Bridge, Ioannis Xenarios, Julien Dorier, and Isaac Crespo. Expert curation for building network-based dynamical models: a case study on atherosclerotic plaque formation. *Database*, 2018(2018):31, Jan 2018. ISSN 1758-0463. doi: 10.1093/database/bay031.

[17] Yves Crama and Peter L Hammer. *Boolean functions: Theory, algorithms, and applications*. Cambridge University Press, 2011.

[18] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[19] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558, Apr 1982. ISSN 00278424. doi: 10.1073/pnas.79.8.2554.

[20] Stefan Bornholdt. Boolean network models of cellular regulation: prospects and limitations. *Journal of The Royal Society Interface*, 5, Aug 2008. ISSN 1742-5689. doi: 10.1098/rsif.2008.0132.focus.

[21] Claudine Chaouiya, Ouerdia Ourrad, and Ricardo Lima. Majority Rules with Random Tie-Breaking in Boolean Gene Regulatory Networks. *PLoS ONE*, 8(7):69626, Jul 2013. ISSN 19326203. doi: 10.1371/journal.pone.0069626.

[22] José E. R. Cury, Pedro T. Monteiro, and Claudine Chaouiya. Partial Order on the set of Boolean Regulatory Functions, Jan 2019. arXiv preprint arXiv:1901.07623.

[23] Archie Blake. Canonical expressions in boolean algebra. *PhD Thesis*, 1937. Department of Mathematics, University of Chicago.

[24] Stuart A Kauffman. *The origins of order: Self-organization and selection in evolution*. Oxford University Press, USA, 1993.

[25] Marco Gherardi and Pietro Rotondo. Measuring logic complexity can guide pattern discovery in empirical systems. *Complexity*, 21:397–408, Aug 2016.

[26] Itai Benjamini, Oded Schramm, and David B. Wilson. Balanced Boolean functions that can be evaluated so that every input bit is unlikely to be read. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, pages 244–250, New York, USA, 2005. ACM Press. doi: 10.1145/1060590.1060627.

[27] Christoph Müssel, Martin Hopfensitz, and Hans A. Kestler. BoolNet — an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics*, 26(10):1378–1380, May 2010. ISSN 1460-2059. doi: 10.1093/bioinformatics/btq124.

[28] John Zobolas. Gitsbe format documentation, 2020. Retrieved from https://druglogics.github.io/druglogics-doc/gitsbe-config.html#gitsbe-format.

[29] Aurélien Naldi. BioLQM: A Java Toolkit for the Manipulation and Conversion of Logical Qualitative Models of Biological Networks. *Frontiers in Physiology*, 9:1605, Nov 2018. ISSN 1664-042X. doi: 10.3389/fphys.2018.01605.

[30] Jacob Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, Apr 1960. ISSN 0013-1644. doi: 10.1177/001316446002000104.

[31] Eirini Tsirvouli, Barbara Niederdorfer, John Zobolas, Touré Vasundra, Åsmund Flobak, and Martin Kuiper. CASCADE - CAncer Signaling CAusality DatabasE, Oct 2020. Retrieved from https://github.com/druglogics/cascade.

[32] J. Richard Landis and Gary G. Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159, Mar 1977. ISSN 0006341X. doi: 10.2307/2529310.

[33] Mary L. McHugh. Interrater reliability: The kappa statistic. *Biochemia Medica*, 22(3):276–282, Oct 2012. ISSN 13300962. doi: 10.11613/bm.2012.031.

[34] Albert László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439): 509–512, Oct 1999. ISSN 00368075. doi: 10.1126/science.286.5439.509.

[35] H. Jeong, B. Tombor, R. Albert, Z. N. Oltval, and A. L. Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, Oct 2000. ISSN 00280836. doi: 10.1038/35036627.

[36] Stefan Wuchty. Scale-Free Behavior in Protein Domain Networks. *Molecular Biology and Evolution*, 18 (9):1694–1702, Sep 2001. ISSN 1537-1719. doi: 10.1093/oxfordjournals.molbev.a003957.

[37] Réka Albert. Scale-free networks in cell biology. *Journal of Cell Science*, 118(21):4947–4957, Nov 2005. ISSN 00219533. doi: 10.1242/jcs.02714.

[38] Maximino Aldana, Enrique Balleza, Stuart Kauffman, and Osbaldo Resendiz. Robustness and evolvability in genetic regulatory networks. *Journal of Theoretical Biology*, 245(3):433–448, Apr 2007. ISSN 00225193. doi: 10.1016/j.jtbi.2006.10.027.

[39] Raya Khanin and Ernst Wit. How scale-free are biological networks. *Journal of Computational Biology*, 13(3):810–818, Apr 2006. ISSN 10665277. doi: 10.1089/cmb.2006.13.810.

[40] Anna D. Broido and Aaron Clauset. Scale-free networks are rare. *Nature Communications*, 10(1):1–10, Dec 2019. ISSN 20411723. doi: 10.1038/s41467-019-08746-5.

[41] Maximino Aldana. Boolean dynamics of networks with scale-free topology. *Physica D: Nonlinear Phenomena*, 185(1):45–66, 2003. doi: 10.1016/S0167-2789(03)00174-X.

[42] Réka Albert and Albert László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, Jan 2002. ISSN 00346861. doi: 10.1103/RevModPhys.74.47.

[43] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467, Mar 1969. ISSN 10958541. doi: 10.1016/0022-5193(69)90015-0.

[44] Ilya Shmulevich and Stuart A. Kauffman. Activities and sensitivities in Boolean network models. *Physical Review Letters*, 93(4):048701, Jul 2004. ISSN 00319007. doi: 10.1103/PhysRevLett.93.048701.

[45] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, Feb 2002. ISSN 1367-4803. doi: 10.1093/bioinformatics/18.2.261.

[46] Fangting Li, Tao Long, Ying Lu, Qi Ouyang, and Chao Tang. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, 101(14): 4781–4786, Apr 2004. ISSN 00278424. doi: 10.1073/pnas.0305937101.

[47] Maria I. Davidich and Stefan Bornholdt. Boolean Network Model Predicts Cell Cycle Sequence of Fission Yeast. *PLoS ONE*, 3(2), Feb 2008. ISSN 1932-6203. doi: 10.1371/journal.pone.0001672.

[48] Andreas Wagner. Evolution of gene networks by gene duplications: A mathematical model and its implications on genome organization. *Proceedings of the National Academy of Sciences of the United States of America*, 91(10):4387–4391, May 1994. ISSN 00278424. doi: 10.1073/pnas.91.10.4387.

[49] Panos Oikonomou and Philippe Cluzel. Effects of topology on network evolution. *Nature Physics*, 2(8): 532–536, Aug 2006. ISSN 17452481. doi: 10.1038/nphys359.

[50] Lars Kaderali, Eva Dazert, Ulf Zeuge, Michael Frese, and Ralf Bartenschlager. Reconstructing signaling pathways from RNAi data using probabilistic Boolean threshold networks. *Bioinformatics*, 25(17): 2229–2235, Sep 2009. ISSN 1460-2059. doi: 10.1093/bioinformatics/btp375.

[51] Sam F Greenbury, Iain G Johnston, Matthew A Smith, Jonathan P K Doye, and Ard A Louis. The effect of scale-free topology on the robustness and evolvability of genetic regulatory networks. *Journal of Theoretical Biology*, 267(1):48–61, 2010. ISSN 0022-5193. doi: 10.1016/j.jtbi.2010.08.006.

[52] John Jack, John F. Wambaugh, and Imran Shah. Simulating Quantitative Cellular Responses Using Asynchronous Threshold Boolean Network Ensembles. *BMC Systems Biology*, 5(1):1–13, Jul 2011. ISSN 17520509. doi: 10.1186/1752-0509-5-109.

[53] Jorge G.T. Zañudo, Maximino Aldana, and Gustavo Martínez-Mekler. Boolean threshold networks: Virtues and limitations for biological modeling. *Information Processing and Biological Systems*, 11: 113–151, 2011. ISSN 18684394. doi: 10.1007/978-3-642-19621-8_6.

[54] Roded Sharan and Richard M. Karp. Reconstructing Boolean Models of Signaling. *Journal of Computational Biology*, 20(3):249–257, Mar 2013. ISSN 1066-5277. doi: 10.1089/cmb.2012.0241.

[55] Aurélien Naldi, Céline Hernandez, Nicolas Levy, Gautier Stoll, Pedro T. Monteiro, Claudine Chaouiya, Tomáš Helikar, Andrei Zinovyev, Laurence Calzone, Sarah Cohen-Boulakia, Denis Thieffry, and Loïc Paulevé. The CoLoMoTo Interactive Notebook: Accessible and Reproducible Computational Analyses for Qualitative Biological Networks. *Frontiers in Physiology*, 9:680, Jun 2018. ISSN 1664-042X. doi: 10.3389/fphys.2018.00680.

[56] Ronald L Graham, Donald E Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Longman Publishing Co., Inc., USA, 2nd edition, 1994. ISBN 0201558025.