4-2015

Stratification and enumeration of Boolean functions by canalizing depth

Qijun He
Clemson University

Matthew Macauley
Clemson University, macaule@clemson.edu

STRATIFICATION AND ENUMERATION OF BOOLEAN FUNCTIONS BY CANALIZING DEPTH

QIJUN HE AND MATTHEW MACAULEY

ABSTRACT. Boolean network models have gained popularity in computational systems biology over the last dozen years. Many of these networks use canalizing Boolean functions, which has led to increased interest in the study of these functions. The canalizing depth of a function describes how many canalizing variables can be recursively "picked off", until a non-canalizing function remains. In this paper, we show how every Boolean function has a unique algebraic form involving extended monomial layers and a well-defined core polynomial. This generalizes recent work on the algebraic structure of nested canalizing functions, and it yields a stratification of *all* Boolean functions by their canalizing depth. As a result, we obtain closed formulas for the number of n-variable Boolean functions with depth k, which simultaneously generalizes enumeration formulas for canalizing, and nested canalizing functions.

1. INTRODUCTION

Boolean networks were invented in 1969 by S. Kauffman, who proposed them as models of gene regulatory networks [Kau69]. They were slow to catch on, but since a seminal paper [AO03] from 2003, where Albert and Othmer modeled the segment polarity gene in the fruit fly *Drosophila melanogaster*, they have emerged as popular models for a variety of biological networks. Random Boolean networks (RBNs) have been studied throughout the years, with various restrictions on the functions or wiring diagrams to better reflect salient properties of actual biological networks. For example, without such restrictions, RBNs display chaotic behavior in the sense that they are very sensitive to small perturbations. In contrast, biological systems must be robustly designed [LLL$^+$04] in order to withstand a variety of internal (e.g., mutation or gene knockout) and external (e.g., environmental) changes. In 1942, the geneticist H. Waddington defined the concept of *canalization* to study this robustness. Over 30 years later in [Kau74], Kauffman introduced the notion of canalizing Boolean functions in order to accurately reflect the behavior of biological systems in the setting of Boolean network models. Another thirty years after that, Kauffman and collaborators further expanded the canalization concept and introduced the class of nested canalizing functions [KPST03], which can be thought of as functions that are fully "recursively canalizing."

In the last decade, canalizing functions have been extensively studied by researchers in the fields of mathematics, biology, physics, computer science, and electrical engineering. For example, Shmulevich and Kauffman showed that canalizing functions have lower activities and sensitivities than random Boolean functions, and this causes Boolean network models using these functions to be more stable; see [SK04] and [KPST04]. More work on the dynamical stability of canalizing Boolean networks was done in [MA05] and in [KH07], where the authors explored the relationship between the proportion of canalizing functions in a network, and whether it lies in the ordered or chaotic dynamical regime, or near the so-called critical threshold. The evolution of canalizing Boolean networks was studied in [SD07]. Fourier analysis has shown that canalizing Boolean networks maximize mutual information [KKBS14]. An exact formula was derived for the number of Boolean canalizing functions in [JSK04]. Canalizing functions have been generalized from Boolean to over general finite fields in [ML12].

Nested canalizing functions (NCFs) have also gained significant attention. In [Pei10] and [KLAL14], the authors study the phase diagram of Boolean networks with NCFs. A recursive formula for the number of NCFs was derived in [JRL07], where they were shown to be what the electrical engineering community calls unate cascade functions [BB78]. NCFs have been studied algebraically through the lens of toric

2010 *Mathematics Subject Classification.* 06E30.

Key words and phrases. Boolean function, Boolean network, canalizing depth, canalizing function, enumeration, extended monomial layer, nested canalizing function.

Partially supported by NSF grant DMS-1211691.

varieties [JL07], and in [LAM$^+$13], where the authors obtained a unique algebraic form by writing an NCF in extended monomial layers. This allowed the authors to enumerate the number of NCFs. It also provided the tools for the development of an algorithm in [HJ12] to reverse-engineering a nested canalizing Boolean network from partial data. In [LDM12], the authors generalized the notion of both canalizing and nested canalizing functions by introducing the class of partially nested canalizing functions. Loosely speaking, these are the functions that are "somewhat recursively canalizing." The dynamics of Boolean networks built with these functions has been studied in [LDM12] and [JM13].

In this article, we carry out a detailed mathematical study on canalization of Boolean functions. Instead of thinking of partially (or fully) nested canalizing functions as a subclass of Boolean functions, we consider canalization as a property of *all* Boolean functions. We modify the notion of *canalizing depth* from [LDM12] to quantify the degree to which a function exhibits a recursive canalizing structure. From here, we show that every Boolean function has a unique algebraic form using extended monomial layers, generalizing what was done for NCFs in [LAM$^+$13]. Once one "peels off" these layers, a unique non-canalizing *core polynomial* remains. This gives a well-defined stratification of *all* Boolean functions by canalizing depth and monomial layers, which includes the canalizing, non-canalizing, and NCFs as special cases. We say that a function is k-*canalizing* if it has canalizing depth at least k. Our stratification allows us to derive exact formulas for the number the k-canalizing functions on n variables. The special cases of $k = 1$ and $k = n$ yield the enumeration results of canalizing, and nested canalizing functions from [JSK04] and [LAM$^+$13], respectively.

This paper is organized as follows. After introducing necessary preliminaries in Section 2, we define k-canalizing functions, canalizing depth and core functions in Section 3. Next, we characterize Boolean functions by a unique polynomial form in Section 4 and use this to stratify all Boolean functions by extended monomial layers and their core polynomials, which are slighly different from the aforementioned core functions. In Section 5, we use this structure to derive exact enumeration formulas for the number of functions with a fixed canalizing depth. Finally, we end in Section 6 with some concluding remarks and directions of current and future research.

2. Canalizing and nested canalizing functions

To make this paper self-contained we will restate some well-known definitions; see, e.g., [KPST03]. This is also needed because there are slight variations in certain definitions throughout the literature. Let $\mathbb{F}_2 = \{0, 1\}$ be the binary field, and let $f\colon \mathbb{F}_2^n \to \mathbb{F}_2$ be an n-variable Boolean function.

Definition 2.1. A Boolean function $f(x_1, \ldots, x_n)$ is *essential* in the variable x_i if there exists a sequence $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n \in \mathbb{F}_2$ such that

$$f(a_1, \ldots, a_{i-1}, 0, a_{i+1}, \ldots, a_n) \neq f(a_1, \ldots, a_{i-1}, 1, a_{i+1}, \ldots, a_n).$$

In this case, we say that x_i is an *essential variable* of f. Variables that are non-essential are *fictitious*.

S. Kauffman defined canalizing Boolean functions in [Kau74] to capture the general stability of gene regulatory networks. In that paper, a Boolean function f is canalizing in variable x_i, with canalizing input a and canalized output b, if, whenever x_i takes on the value a, the output of f is b, regardless of the inputs of other variables. As a consequence, constant functions are trivially canalizing. We will soon see why it is more mathematically natural to exclude these functions, among others. This is done by the following small adjustment to the original definition that does not change the overall idea.

Definition 2.2. A Boolean function $f\colon \mathbb{F}_2^n \to \mathbb{F}_2$ is *canalizing* if there exists a variable x_i, a Boolean function $g(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, and $a, b \in \mathbb{F}_2$ such that

$$f(x_1, \ldots, x_n) = \begin{cases} b & x_i = a, \\ g \not\equiv b & x_i \neq a. \end{cases}$$

In this case, x_i is a *canalizing variable*, the input a is the *canalizing input*, and the output value b when $x_i = a$ is the corresponding *canalized output*.

The only difference of our definition is the added restriction that g can not be the constant function b. In other words, *we require a canalizing function to be essential in its canalizing variable*. The original definition was motivated by the stability of canalizing functions while our definition tries to capture the

dominance of the canalizing variable. At first glance, our additional restriction might seem artificial or insignificant. However, it is unequivocally more natural when considering the algebraic structure of Boolean functions, which is at the heart of the stratification derived in this paper.

In Definition 2.2, when the canalizing variable does not receive its canalizing input a, the function g obtained by plugging in $x_i = \overline{a}$ can be an arbitrary Boolean function. To better model a dynamically stable network, in [KPST03] Kauffman proposed that in this case, there should be another variable x_j that is canalizing for a particular input, and so on. This leads to the following definition, where σ is a total ordering, or permutation, of $[n] := \{1, \ldots, n\}$. We write this as $\sigma = \sigma(1), \sigma(2), \ldots, \sigma(n)$, and say that $\sigma \in \mathfrak{S}_n$, the symmetric group on $[n]$.

Definition 2.3. A Boolean function $f\colon \mathbb{F}_2^n \to \mathbb{F}_2$ is *nested canalizing* with respect to the permutation $\sigma \in \mathfrak{S}_n$, inputs a_i and outputs b_i, for $i = 1, 2, \ldots, n$, if it can be represented in the form:

$$f(x_1, \ldots, x_n) = \begin{cases} b_1 & x_{\sigma(1)} = a_1, \\ b_2 & x_{\sigma(1)} \neq a_1, x_{\sigma(2)} = a_2, \\ b_3 & x_{\sigma(1)} \neq a_1, x_{\sigma(2)} \neq a_2, x_{\sigma(3)} = a_3, \\ \vdots & \vdots \\ b_n & x_{\sigma(1)} \neq a_1, \ldots, x_{\sigma(n-1)} \neq a_{n-1}, x_{\sigma(n)} = a_n, \\ \overline{b_n} & x_{\sigma(1)} \neq a_1, \ldots, x_{\sigma(n-1)} \neq a_{n-1}, x_{\sigma(n)} \neq a_n. \end{cases} \tag{1}$$

The idea of nested canalizing is in that some sense, it is "recursively canalizing" for exactly n steps. As an analogy, one can consider a nested canalizing function as an onion. We can peel off variables one at a time by not taking the canalizing input of each variable (i.e., by plugging in $x_i = \overline{a_i}$). Before we peel off the 'inner' variables, we need to peel off the 'outer' variables first. In the end, we are left with the constant function $\overline{b_n}$. We will return to this onion analogy several times throughout this paper to highlight our main ideas.

Remark 2.4. Since $b_n \neq \overline{b_n}$, a nested canalizing function is essential in all n variables.

If a Boolean function is nested canalizing, then at least one (of all $n!$) ordering of the variables yields an equation in the form of Eq. (1). Note that such variable orderings are not unique, and the number of such orderings depends on the function f. For example, we can write the function $f_1(x, y, z) = xyz$ as in Eq. (1) using any of the 6 orderings of the variables $\{x, y, z\}$. In contrast, for $f_2(x, y, z) = x(yz + 1)$, only 2 orderings would work, namely (x, y, z) and (x, z, y).

3. k-Canalizing Functions

Nested canalizing functions have a very restrictive structure and become increasingly sparse as the number of input variables increases [JRL07]. In a real network model, it is often the case that not all variables exhibit nested canalizing behavior. Moreover, the first several canalizing variables play more central roles than the remaining variables. Thus, it is natural to consider functions that are canalizing, but not nested canalization. For example, one function in the segment polarity gene in by Albert and Othmer's seminal paper [AO03] is canalizing but not nested canalizing. For another example, one can look at the lactose (*lac*) operon, which regulates the transport and metabolism of lactose in *Escherichia coli*. In [RH13], a simple Boolean network model of the *lac* operon was proposed, where the regulatory function for lactose was

$$f_L(t+1) = \overline{G_e} \wedge \left[(L \wedge \overline{E}) \vee (L_e \wedge E) \right].$$

In a sentence, this means "internal lactose (L) will be present the following timestep if there is no external glucose (G_e), *and* at least one of the following holds:

- there already is internal lactose present, but the enzyme β-galactosidase (E) that breaks it down is absent;
- there is external lactose (L_e) available and the *lac* permease transporter protein (also represented by E since it is transcribed by the same gene) is present.

The variable $\overline{G_e}$ (though sometimes considered a parameter) is canalizing because it acts as a "shut-down" switch: if $G_e = 1$, then $f_L = 0$ regardless of the other variables. In other words, we can write this as

$$f_L(G_e, L_e, L, E) = \begin{cases} 0 & G_e = 1, \\ (L \wedge \overline{E}) \vee (L_e \wedge E) & G_e \neq 0. \end{cases}$$

The function $g = (L_e \wedge E) \vee (L \wedge \overline{E})$ is not canalizing, and so the 5-variable function f_L is canalizing but not nested canalizing. In the framework that we are about to define, this function has canalizing depth 1.

Due to both theoretical and practical reasons, a relaxation of the nested canalizing structure is often necessary. This was done in [LDM12], where there authors defined partially nested canalizing functions, and then distinguished between the "active depth" and "full depth" of a function. Our definition of k-canalizing functions is similar to what it means in their paper to be "partially nested canalizing of active depth at least k." As before, the small differences are motivated by the desire to have a natural unique algebraic form.

Definition 3.1. A Boolean function $f(x_1, \ldots, x_n)$ is *k-canalizing*, where $0 \leq k \leq n$, with respect to the permutation $\sigma \in \mathfrak{S}_n$, inputs a_i, and outputs b_i, for $1 \leq i \leq k$, if

$$f(x_1, \ldots, x_n) = \begin{cases} b_1 & x_{\sigma(1)} = a_1, \\ b_2 & x_{\sigma(1)} \neq a_1, x_{\sigma(2)} = a_2, \\ b_3 & x_{\sigma(1)} \neq a_1, x_{\sigma(2)} \neq a_2, x_{\sigma(3)} = a_3, \\ \vdots & \vdots \\ b_k & x_{\sigma(1)} \neq a_1, \ldots, x_{\sigma(k-1)} \neq a_{k-1}, x_{\sigma(k)} = a_k, \\ g \not\equiv b_k & x_{\sigma(1)} \neq a_1, \ldots, x_{\sigma(k-1)} \neq a_{k-1}, x_{\sigma(k)} \neq a_k. \end{cases} \tag{2}$$

where $g = g(x_{\sigma(k+1)}, \ldots, x_{\sigma(n)})$ is a Boolean function on $n - k$ variables. When g is not a canalizing function, the integer k is the *canalizing depth* of f. Furthermore, if g is not a constant function, then we call it a *core function* of f, denoted by f_C.

As with canalizing and nested canalizing functions, the $g \not\equiv b_k$ condition ensures that f is essential in the final variable, $x_{\sigma(k)}$.

Remark 3.2. Since $g \not\equiv b_k$, a function f that is k-canalizing with respect to $\sigma \in \mathfrak{S}_n$, inputs a_i and outputs b_i is essential in each $x_{\sigma(i)}$ for $i = 1, \ldots, k$.

The representation of a k-canalizing function f in the form of Eq. (2), even when k is the canalizing depth, is generally not unique since it depends on the variable ordering. However, we will prove that several key properties, such as the canalizing depth and core function $f_C = g$ (if there is one), are independent of representation. It is worth noting that if g is constant, then g need not be unique, i.e., both $g \equiv 0$ and $g \equiv 1$ can arise. This is why we do not allow constant core functions. The following observation is elementary.

Remark 3.3. If f is k-canalizing with respect to $\sigma \in \mathfrak{S}_n$, inputs a_i and outputs b_i, then any initial segment $x_{\sigma(1)}, \ldots, x_{\sigma(j)}$ with the same canalized output $b_1 = \cdots = b_j$ can be permuted to yield an equivalent form as in Eq. (2).

Definition 3.4. If $f(x_1, \ldots, x_n)$ is k-canalizing with respect to $\sigma \in \mathfrak{S}_n$, inputs a_i and outputs b_i, then for each $j \leq k$, define the Boolean function $g_j^\sigma(x_{\sigma(j+1)}, \ldots, x_{\sigma(n)})$ to be the result of plugging in $x_{\sigma(i)} = \overline{a_i}$ for $i = 1, \ldots, j$.

In plain English, the function g_j^σ is the result of when the first j canalizing variables do *not* get their canalizing inputs. We can now show that the canalizing depth k and the core function f_C are independent of the order of the variables. Moreover, the ambiguity of variable orderings is well-controlled in that they are partitioned into blocks called *layers* via extended monomials, and variables can be permuted arbitrarily if and only if they lie in the same layer. This generalizes the observation in Remark 3.3.

Proposition 3.5. *Suppose an n-variable Boolean function f is k-canalizing with respect to the permutation σ, inputs a_i and outputs b_i, for $1 \leq i \leq k$, and k'-canalizing with respect to the permutation σ', inputs a'_j and outputs b'_j, for $1 \leq j \leq k'$, such that both g and g', obtained by substituting $\overline{a_i}$ for $x_{\sigma(i)}$ and $\overline{a'_j}$ for*

$x_{\sigma'(j)}$ respectively, are not canalizing. Then $k = k'$ and the resulting core functions, if they exist, are the same.

Proof. Assume f is canalizing, because otherwise, $k = k' = 0$ and the result is trivial. Without losing generality we can assume $\sigma(1) \neq \sigma'(1)$, since if this were not the case, we could simply input $\overline{a_1} = \overline{a_1'}$ for $x_{\sigma(1)} = x_{\sigma'(1)}$ and consider $g_1^{\sigma} = g_1^{\sigma'}$. (Note that if $\sigma(1) = \sigma'(1)$ and $a_1 \neq a_1'$, then $b_1 \neq b_1'$, which means that f is completely determined by the input to $x_{\sigma(1)} = x_{\sigma'(1)}$. In this case, f has only one essential variable, and so $k = 1$. Moreover, both g_1^{σ} and $g_1^{\sigma'}$ are constant functions. Thus f has no core function.)

Since g is non-canalizing, it is not essential in $x_{\sigma(1)}$, and thus $\sigma(1) = \sigma'(j^*)$ for some $1 < j^* \leq k'$. We claim that we may assume without loss of generality that $a'_{j^*} = a_1$ and $b'_{j^*} = b_1$. To see why, first suppose that $a'_{j^*} = \overline{a_1}$ and consider the two possible inputs to $x_{\sigma'(j^*)} = x_{\sigma(1)}$ in the function $g^{\sigma'}_{j^*-1}$. If this variable takes its canalizing input $\overline{a_1}$, then the output is b'_{j^*}. However, since f is canalizing in $x_{\sigma'(j^*)} = x_{\sigma(1)}$, then the other input a_1 would yield the output b_1. In other words, $g^{\sigma'}_{j^*-1}$ is completely determined by the input to $x_{\sigma'(j^*)}$, so all subsequent variables are fictitious. Therefore, $g^{\sigma'}_{j^*} = g'$ must be constant, hence $j^* = k'$. Moreover, this function must be $g' \equiv b_1$ because it only arises when $x_{\sigma'(j^*)} = x_{\sigma(1)}$ takes the canalizing input a_1. Since f is essential in $x_{\sigma'(j^*)} = x_{\sigma(1)}$, then Remark 3.2 implies that $b'_{j^*} = \overline{b_1}$, the opposite value of $g' \equiv b_1$. Thus, we have two equivalent ways to represent $g^{\sigma'}_{j^*-1} = g^{\sigma'}_{k'-1}$:

$$g^{\sigma'}_{k'-1} = \begin{cases} \overline{b_1} & x_{\sigma'(k')} = \overline{a_1}, \\ g' \equiv b_1 & x_{\sigma'(k')} = a_1. \end{cases} = \begin{cases} b_1 & x_{\sigma'(k')} = a_1, \\ g' \equiv \overline{b_1} & x_{\sigma'(k')} = \overline{a_1}. \end{cases} \tag{3}$$

In other words, switching the triple of values $(a'_{k'}, b'_{k'}, g')$ from $(\overline{a_1}, \overline{b_1}, b_1)$ to $(a_1, b_1, \overline{b_1})$ in the original representation of f with respect to $\sigma' \in \mathfrak{S}_n$ does not change the function, so we may assume that $a'_{j^*} = a_1$ and $b'_{j^*} = b_1$, as claimed. The proof for the case when $b'_{j^*} = \overline{b_1}$ is almost the same.

Since f is canalizing in $x_{\sigma'(j^*)} = x_{\sigma(1)}$ with input a_1 and output b_1, we must also have $b'_j = b_1$ for all $1 \leq j \leq j^*$. By Remark 3.3, we can create a new permutation σ'' by swapping the order of $x_{\sigma'(1)}$ and $x_{\sigma'(j^*)}$ in σ'. Clearly, f is k'-canalizing with respect to σ'' and $g^{\sigma'}_{k'} = g^{\sigma''}_{k'}$. Since $x_{\sigma(1)} = x_{\sigma''(1)}$, the result follows from induction on $g_1^{\sigma} = g_1^{\sigma''}$. We conclude that $k = k'$.

Finally, we need to show that when f has a core function f_C, it is unique. The non-canalizing functions g and g' are essential in the same set of variables. If they are both constant functions, then they actually need not be the same, due to the different ways to write g' as in Eq. (3). Otherwise, they are core functions for f, and are obtained by substituting the same set inputs for the same set of variables, thus we must have $f_C = g = g'$. □

It is worth noting that Definition 3.1 is similar to the definition of k-partially nested canalizing functions (k-PNCFs) in [LDM12]. In fact, these two definitions hold the same motivation but are from different perspectives. In [LDM12], the authors treat k-PNCFs as a subclass of Boolean functions. While we prefer to consider canalization as a property of Boolean functions and different functions have different extent of canalization. This provides us a well-defined way to classify all Boolean functions on n variables.

Returning to our onion analogy, now we can think of all Boolean functions as onions. For each Boolean function, we can try to peel off its variables as we did for nested canalizing functions. We will have to stop once we get to a non-canalizing function. In this sense, nested canalizing functions would be the 'best' onions since we can peel off all the variables and non-canalizing would be the 'worst'. The k-canalizing functions would be those for which one can be peeled off at least k variables. Though a unique core function $f_C = g$ only exists when g is non-constant, we will soon see how every Boolean function, whether or not it has a core function, has a unique *core polynomial* that extends the notion of a core function.

Example 3.6. The Boolean function $f(x, y, z, w) = xy(z + w)$ has canalizing depth 2 and core function $f_C = z + w$.

Remark 3.7. In our framework, if we consider the set of all Boolean functions on n variables, then:

- The canalizing depth of a k-canalizing function is at least k.
- A non-canalizing function has canalizing depth 0, and if it is non-constant, then its core function is itself.

- Every Boolean function is 0-canalizing.
- The 1-canalizing functions are precisely the canalizing functions.
- The n-canalizing functions are precisely the nested canalizing functions.
- If a function f has canalizing depth k and the resulting g is constant, then f has $n-k$ fictitious variables, and it is a nested canalizing function on its k essential variables.

4. Characterizations of k-Canalizing Functions

4.1. **Polynomial Form of k-Canalizing Functions.** It is well-known [LNC96] that any Boolean function f can be uniquely expressed as a square-free polynomial, called its *algebraic normal form*. Equivalently, the set of Boolean functions on n variables is isomorphic to the quotient ring $R := \mathbb{F}_2[x_1, \ldots, x_n]/I$, where $I = \langle x_i^2 - x_i : 1 \leq i \leq n \rangle$. Henceforth in this section, when we speak of Boolean polynomials, we assume they are square-free. Additionally, we will define $\hat{x}_i := (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ for notational convenience. In this section, we will extend work on NCFs from [LAM$^+$13] to general k-canalizing Boolean functions.

Lemma 4.1. *A Boolean function $f(x_1, \ldots, x_n)$ is canalizing in variable x_i, for some $1 \leq i \leq n$, with input a_i and output b_i, if and only if*

$$f = (x_i + a_i)g(\hat{x}_i) + b_i\,,$$

for some polynomial $g \not\equiv 0$.

Proof. Suppose f is canalizing in x_i. Write f in its algebraic normal and factor it as

$$f = x_i\, q(\hat{x}_i) + r(\hat{x}_i)\,,$$

where q and r are the quotient and remainder of f when divided by x_i. Note that $b_i = a_i q(\hat{x}_i) + r(\hat{x}_i)$, and since $a_i + a_i = 0$ in $\mathbb{F}_2$,

$$f = (x_i + a_i)q(\hat{x}_i) + \big[r(\hat{x}_i) + a_i q(\hat{x}_i)\big] = (x_i + a_i)q(\hat{x}_i) + b_i\,.$$

The function $g(\hat{x}_i) := q(\hat{x}_i)$ is nonzero because f is essential in x_i. This establishes necessity, and sufficiency is obvious. □

By applying the above lemma recursively, we get the following theorem.

Theorem 4.2. *A Boolean function $f(x_1, \ldots, x_n)$ is k-canalizing, with respect to permutation $\sigma \in \mathfrak{S}_n$, inputs a_i and outputs b_i, for $1 \leq i \leq k$, if and only if it has the polynomial form*

$$f(x_1, \ldots, x_n) = (x_{\sigma(1)} + a_1)g(\hat{x}_i) + b_1\,,$$

where

$$g(\hat{x}_i) = (x_{\sigma(2)} + a_2)\Big[\ldots\big[(x_{\sigma(k-1)} + a_{k-1})[(x_{\sigma(k)} + a_k)\bar{g} + \Delta b_{k-1}] + \Delta b_{k-2}\big]\ldots\Big] + \Delta b_1$$

for some polynomial $\bar{g} = \bar{g}(x_{\sigma(k+1)}, \ldots, x_{\sigma(n)}) \not\equiv 0$, where $\Delta b_i := b_{i+1} - b_i = b_{i+1} + b_i$. □

4.2. **Dominance Layers of Boolean Functions.** One weakness of Theorem 4.2 is that given a Boolean function f, the representation of f into the above form, even when k is exactly the canalizing depth, is not unique. In a k-canalizing function, some variables are "more dominant" than others. We will classify all variables of a Boolean function into different layers according to the extent of their dominance, extending work from [LAM$^+$13] from NCFs to general Boolean functions. The "most dominant" variables will be precisely those that are canalizing. Recall that we are always working in the quotient ring $R = \mathbb{F}_2[x_1, \ldots, x_n]/I$, though at times it is helpful to consider the algebraic normal form of a polynomial as an element of $\mathbb{F}_2[x_1, \ldots, x_n]$.

Definition 4.3. A Boolean function $M(x_1, \ldots, x_m)$ is an *extended monomial* in variables $x_1, \ldots, x_m$ if

$$M(x_1, \ldots, x_m) = \prod_{i=1}^{m}(x_i + a_i),$$

where $a_i \in \mathbb{F}_2$ for each $i = 1, \ldots, m$.

An extended monomial in R is an extended monomial of a subset of $\{x_1, \ldots, x_n\}$. In other words, it is simply a product $\prod_{i=1}^{n} y_i$, where each y_i is either x_i, $\overline{x_i}$, or 1. Using extended monomials, we can refine Theorem 4.2 to obtain a unique *extended monomial form* of any Boolean function.

Proposition 4.4. *Given a Boolean function $f(x_1, \ldots, x_n)$, all variables are canalizing if and only if $f = M(x_1, \ldots, x_n) + b$, where M is an extended monomial in all variables.*

Proof. Suppose all n variables are canalizing if f, and so f is essential in every variable. Since x_1 is canalizing, Lemma 4.1 says that $f = (x_1 + a_1)g(\hat{x}_1) + b$ for some $a_1, b \in \mathbb{F}_2$, and $g \not\equiv 0$. In particular, this means that $(x_1 + a_1) \mid (f + b)$ in $\mathbb{F}_2[x_1, \ldots, x_n]$. Since x_2 is also canalizing, $f(x_1, a_2, \ldots, x_n) \equiv b'$ for some a_2 and b'. Plugging in $x_1 = a_1$ yields $f(a_1, a_2, x_3, \ldots, x_n) \equiv b = b'$, and so

$$(x_2 + a_2) \mid (f + b) = (x_1 + a_1)g(x_2, \ldots, x_n)\,.$$

Since $x_1 + a_1$ and $x_2 + a_2$ are co-prime, we get $(x_2 + a_2) \mid g(x_2, \ldots, x_n)$. Note that $g(\hat{x}_1) \not\equiv 0$, hence, we have $g(\hat{x}_1) = (x_2 + a_2)g'(x_3, \ldots, x_n)$ where $g'(x_3, \ldots, x_n) \not\equiv 0$. Thus we have $f = (x_1 + a_1)(x_2 + a_2)g'(x_3, \ldots, x_n) + b$. Necessity of the proposition now follows from induction, and sufficiency is obvious. □

We are now ready to prove the main result of this section. This is a generalized version of Theorem 4.2 in [LAM$^+$13]. We will obtain a new *extended monomial form* of a Boolean function f by induction. In this form, all variables will be classified into different layers according to their dominance. The canalizing variables are the *most dominant* variables. Thus, a Boolean function may have one, none, or many "most dominant" variables. As in [LAM$^+$13], variables in the same layer will have the same level of dominance, with the variables in the outer layers being "more dominant" than those in the inner layers.

Theorem 4.5. *Every Boolean function $f(x_1, \ldots, x_n) \not\equiv 0$ can be uniquely written as*

$$f(x_1, \ldots, x_n) = M_1(M_2(\cdots(M_{r-1}(M_r p_C + 1) + 1)\cdots) + 1) + b, \tag{4}$$

where each $M_i = \prod_{j=1}^{k_i}(x_{i_j} + a_{i_j})$ is a nonconstant extended monomial, $p_C \not\equiv 0$ is the core polynomial *of f, and $k = \sum k_i$ is the canalizing depth. Each x_i appears in exactly one of $\{M_1, \ldots, M_r, p_C\}$, and the only restrictions on Eq.* (4) *are the following "exceptional cases":*

(i) If $p_C \equiv 1$ and $r \neq 1$, then $k_r \geq 2$;
(ii) If $p_C \equiv 1$ and $r = 1$ and $k_1 = 1$, then $b = 0$;

When f is a non-canalizing function, we simply have $p_C = f$.

Before we prove Theorem 4.5, we will define some terms and examine a few details, such as the subtle difference between the core function and core polynomial, and the "exceptional cases", by simple examples. This should help elucidate the more technical parts of the proof.

Definition 4.6. A Boolean function f written in its unique form from Eq. (4) is said to be in *standard monomial form*, and r is its *layer number*. The i^{th} *dominance layer* of f, denoted L_i, is the set of essential variables of M_i. The set of essential variables of p_C is denoted L_∞, and these are called the *recessive variables* of f.

As we will see, when f has a core function f_C, its core polynomial is either $p_C = f_C$ or $p_C = f_C + 1$. When the number of "+1"s that appear in Eq. (4), possibly including b, is even, we have $p_C = f_C$. Otherwise, we have $p_C = f_C + 1$. When a Boolean function f with canalizing depth $k > 0$ fails to have a core function, i.e., the remaining function is either $g \equiv 0$ or $g \equiv 1$, then f is in fact a nested canalizing function on k variables, and its core polynomial is simply $p_C = 1$.

Finally, we will examine the two "exceptional cases". Both of these are necessary to avoid double-counting certain functions and ensure uniqueness, as claimed in Theorem 4.5.

(i) *If $p_C \equiv 1$ and $r \neq 1$.* In this case, if $k_r = 1$, that is $M_r = x_i$ or $\overline{x_i}$, for some i. In either case, this innermost layer can be "absorbed" into the extended monomial M_{r-1}. For example, if $M_r = x_i$, then the inner two layers are

$$M_{r-1}(M_r + 1) + 1 = M_{r-1}(x_i + 1) + 1 = (x_i + 1)\prod_{j=1}^{k_{r-1}}(x_{i_j} + a_{i_j}) + 1, = \hat{M}_{r-1} + 1\,,$$

where $\hat{M}_{r-1} = \overline{x_i} M_{r-1}$ is an extended monomial. Thus, in this case we may assume that the innermost layer has at least two essential variables, hence $k_r \geq 2$.

(ii) If $p_C \equiv 1$ and $r = 1$ and $k_1 = 1$, then for some i, either $f = x_i + b$, or $f = \overline{x_i} + b$. Clearly, there are only two such functions, either $f = x_i$ or $f = \overline{x_i}$, and so allowing both $b = 0$ and $b = 1$ would double-count these. Thus, we may assume that $b = 0$.

Proof of Theorem 4.5. For any non-canalizing function $f \not\equiv 0$, $f = p_C$ and the uniqueness is obvious.

When f is canalizing, we induct on n. When $n = 1$, there are 2 canalizing functions, namely $x = (x)1$ and $x + 1 = (x + 1)1$, both satisfying Eq. (4). For the these 2 functions, since $p_C \equiv 1$, $r = 1$ and $k_1 = 1$, we must have $b = 0$, so the previous representation is also unique.

When $n = 2$, there are 12 canalizing functions, 4 of which are essential in 1 variable, and thus can be uniquely written as in Eq. (4). Now let us consider the 8 canalizing functions that are essential in 2 variables. It is easy to check for all these, both variables x_1 and x_2 are canalizing. Then by Proposition 4.4, all of them are of the form

$$(x_1 + a_1)(x_2 + a_2) + b = M_1 p_C + b\,,$$

where $M_1 = (x_1 + a_1)(x_2 + a_2)$ and $p_C \equiv 1$. In this case, we have $r = 1$ and $k_1 = 2$. Note that when $p_C \equiv 1$, the innermost layer must have at least two essential variables, so uniqueness holds. We have proved that Eq. (4) holds for $n = 1$ and $n = 2$.

Assume now that Eq. (4) is true for any canalizing function that is essential in at most $n - 1$ variables. Consider a canalizing function $f(x_1, \ldots, x_n)$. Suppose that x_{1_j} for each $j = 1, \ldots, k_1$ are all canalizing in f. With the same argument as in Proposition 4.4, we get $f = M_1 g + b$, where $M_1 = (x_{1_1} + a_{1_1}) \cdots (x_{1_{k_1}} + a_{1_{k_1}})$ and $g \not\equiv 0$. If g is non-canalizing, then Eq. (4) holds with $p_C = g$ and $r = 1$. If g is canalizing, then it is a canalizing function that is essential in at most $n - k_1 < n - 1$ variables. By our induction hypothesis, it can be uniquely written as

$$g = M_2(M_3(\cdots(M_{r-1}(M_r p_C + 1) + 1)\cdots) + 1) + b'\,.$$

Note that b' must be 1, otherwise all variables in M_2 will also be most dominant variables of f. This completes the proof. □

Remark 4.7. For any Boolean function f:

(i) Variables in two consecutive layers have different canalized outputs.

(ii) L_1 consists of all the most dominant variables (canalizing variables) of f.

Let us return to our onion analogy, where previously we were peeling off one variable at a time. Furthermore, imagine that each individual variable layer is white if the canalized output $a_i = 0$, and black if $a_i = 1$. Thus, we can think of an extended monomial layer L_i as a maximal block of variable layers of the same color. We can "peel off" an entire L_i at once by plugging in the non-canalizing input $x_{i_j} = \overline{a_{i_j}}$ for each variable in L_i. In other words, we can peel off all black layers, then all white layers, then all black layers, and so on. Moreover, we can read off the colors directly off of the function if it is written in the form of Eq. (2). However, recall that this form of a k-canalizing function, where g is non-canalizing, is not unique. By Theorem 4.5, the order of consecutive variables, $x_{\sigma(i)}$ and $x_{\sigma(i+1)}$, can be transposed if and only if they are in the same L_j. Based on this property, we can enumerate Boolean functions on n variables with canalizing depth k. Roughly speaking, we will do this by counting the number of different layer structures, and then counting the number of (non-canalizing) core polynomials. This last set is just the complement of the set of canalizing functions on those variables, which were enumerated in [JSK04].

Example 4.8. The Boolean function $f(x_1, \ldots, x_7) = x_1\overline{x_2}(x_3x_4(x_5 + x_6 + x_7 + 1) + 1)$ has canalizing depth 4. With respect to the permutation $\sigma = 1, 2, 3, 4$, its canalizing inputs are $(a_i)_{i=1}^4 = (0, 1, 0, 0)$, outputs $(b_i)_{i=1}^4 = (0, 0, 1, 1)$ and the core polynomial is $p_C = x_5 + x_6 + x_7$.

5. Enumeration of k-canalizing functions

Let $B(n, k)$ be the number of Boolean functions on n variables with canalizing depth exactly k. Exact formulas are known for $B(n, k)$ in a few special cases. The number of nested canalizing functions is $B(n, n)$. A recurrence for this was independently derived in the 1970s by engineers studying unate cascade functions [BB78, SK79], and then a closed formula was found by mathematicians studying NCFs

[LAM+13]. The quantity $B(4,k)$ was recently computed in [RDC14]. In this section, we will present a general formula for $B(n,k)$.

Theorem 4.5 indicates that we can construct a Boolean functions with canalizing depth k by adding extended monomial layers to a non-canalizing function on $n-k$ variables. Moreover, the complement of the set of non-canalizing functions are the canalizing functions. Hence, let us begin with a formula for C_n, the number of canalizing functions on n variables. This result was derived in [JSK04] using a probabilistic method. We will include an alternative combinatorial proof using the *truth table* of a Boolean function f. This is the length-2^n vector $(f(x_i))_i$, given some fixed ordering of the elements of $\mathbb{F}_2^n$.

Lemma 5.1. *The number C_n of canalizing Boolean functions on $n \geq 0$ variables is*

$$C_n = 2((-1)^n - n - 1) + \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k+1} 2^{2^{n-k}}.$$

Proof. We wish to count the number of Boolean functions that are canalizing in at least 1 variable. We can construct a truth table of a Boolean function that is canalizing in at least k variables by doing the following. First, pick k variables to be canalizing; there are $\binom{n}{k}$ ways to do this. Next, pick the canalizing input for each canalizing variable; there are 2^k ways to do that. Then, fill out the entries in the truth table of these canalizing inputs with the same canalized output; there are 2 ways to do that. The remaining table has 2^{n-k} entries, so there are $2^{2^{n-k}} - 1$ ways to fill it out such that the corresponding function is non-constant. By inclusion-exclusion, we have $\sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k+1} (2^{2^{n-k}} - 1)$. Note that in this process, there are $2n$ functions of the form $x_i + a_i$, each being counted exactly twice, since we can pick either input as canalizing input. Therefore we have

$$C_n = \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k+1} (2^{2^{n-k}} - 1) - 2n = 2((-1)^n - n - 1) + \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k+1} 2^{2^{n-k}}.$$

□

As examples, one can check that $C_n = 0, 2, 12, 118, 3512, \ldots$ for $n = 0, 1, 2, 3, 4, \ldots$. This is consistent with the results in [JSK04], though it should be noted that all numbers differ by 2 because we do not consider the constant functions to be canalizing.

Recall that there are 2^{2^n} Boolean functions on n variables. Since the non-canalizing functions are the complement of the set of canalizing functions, the following is immediate.

Corollary 5.2. *The number $B^*(n,0)$ of non-constant core polynomials on n variables is*

$$B^*(n,0) = B(n,0) - 2 = (2^{2^n} - C_n) - 2 = 2^{2^n} - 2((-1)^n - n) + \sum_{k=1}^{n} (-1)^k \binom{n}{k} 2^{k+1} 2^{2^{n-k}}.$$

One can check that $B^*(n,0) = 0, 0, 2, 136, 62022, \ldots$, for $n = 0, 1, 2, 3, 4, \ldots$.

Before we derive the general formula for $B(n,k)$, let us first look at the special case when $k = n$. This was computed in [LAM+13], but we include a self-contained proof. Recall that a *composition of* n is a sequence $k_1, \ldots, k_r$ of non-empty integers such that $k_1 + \cdots + k_r = n$. By Theorem 4.5, the standard extended monomial form of a Boolean function with canalizing depth k involves a size-r composition of k with the additional property that $k_r \geq 2$.

Lemma 5.3. *For $n \geq 2$, the number $B(n,n)$ of nested canalizing functions on n variables is given by:*

$$B(n,n) = 2^{n+1} \sum_{r=1}^{n-1} \sum_{\substack{k_1+\ldots+k_r=n \\ k_i \geq 1,\ k_r \geq 2}} \binom{n}{k_1, \ldots, k_r}, \tag{5}$$

where $\binom{n}{k_1,\ldots,k_r} = \frac{n!}{k_1! k_2! \ldots k_r!}$.

Proof. If a Boolean function is nested canalizing on n variables, then by Theorem 4.5, we know its core polynomial must be $p_C = 1$. Let us first fix the layer number r. Then for each choice of $k_1, \ldots, k_r$, with $k_1 + \ldots + k_r = n$, $k_i \geq 1$ and $k_r \geq 2$, there are $\binom{n}{k_1,\ldots,k_r}$ different ways to assign n variables to these r layers. For each variable x_j, we can pick either x_j or $x_j + 1$ to be in its corresponding extended monomial.

Note that we also have 2 choices for b. So the number of nested canalizing functions on n variables with exactly r layers is given by:

$$2^{n+1} \sum_{\substack{k_1+\ldots+k_r=n \\ k_i\geq 1,\ k_r\geq 2}} \binom{n}{k_1,\ldots,k_r}.$$

Then by summing over all possible layer numbers r, for $1 \leq r \leq n-1$, we get the formula in Eq. (5) for $B(n,n)$. □

According to our definition, $B(1,1) = 2$. One also can check that $B(2,2) = 8$, $B(3,3) = 64$, $B(4,4) = 736, \ldots$.

Now we are ready to derive the general formula for $B(n,k)$.

Theorem 5.4. *The number $B(n,k)$ of Boolean functions on n variables with canalizing depth k, for $1 \leq k \leq n$, is*

$$B(n,k) = \binom{n}{k}\left[B(k,k) + B^*(n-k,0)\cdot 2^{k+1}\sum \binom{k}{k_1,\ldots,k_r}\right],$$

where the sum is taken over all compositions of k, and the closed from of $B(k,k)$ is given by Lemma 5.3.

Proof. We can construct a Boolean function f on n variables with canalizing depth k by doing the following. First, pick k variables that are not in the core polynomial p_C. There are $\binom{n}{k}$ different ways to do that. Once we fixed the variables that are not in p_C, we need to consider the following two cases:

Case 1: $p_C \equiv 1$. Then f is actually a nested canalizing function on these k variables. There are $B(k,k)$ of them in total.

Case 2: $p_C \not\equiv 1$. Then p_C is a non-constant core polynomial on $n-k$ variables, so there are $B^*(n-k,0)$ different choices for p_C. Using the same argument as in Lemma 5.3, there are

$$2^k \sum \binom{k}{k_1,\ldots,k_r}$$

different ways for those k variables to form the extended monomials in Equation (4), where the sum is taken over all compositions of k. Note that we also have 2 ways to pick b. Therefore, in this case, there are

$$B^*(n-k,0)\cdot 2^{k+1}\sum \binom{k}{k_1,\ldots,k_r}$$

different Boolean functions.

By combining the above two cases, we get the formula for $B(n,k)$. □

Example 5.5. As previously mentioned, the quantities $B(4,k)$ for $k = 0,\ldots,4$ were computed in [RDC14]. It is easy to check that these values are consistent with our general formula. There are $2^{2^4} = 65536$ Boolean functions on 4 variables. The number of functions with canalizing depth exactly k, for $k = 1,2,3,4$ is

$$B(4,4) = \binom{4}{4}(736+0) = 736$$
$$B(4,3) = \binom{4}{3}(64+0) = 256$$
$$B(4,2) = \binom{4}{2}(8+2\cdot 8\cdot 3) = 336.$$
$$B(4,1) = \binom{4}{1}(2+136\cdot 4\cdot 1) = 2184.$$

Summing these yields the total number of canalizing functions on 4 variables,

$$C_4 = 3512 = 736+256+336+2184 = B(4,4)+B(4,3)+B(4,2)+B(4,1).$$

Thus, there are $B(4,0) = 65536 - 3512 = 62024$ non-canalizing functions on four variables, including the two constant functions.

Note that k-canalizing functions are simply Boolean functions with depth at least k, therefore we immediately get the following equality.

Corollary 5.6. *The number of k_0-canalizing Boolean functions on n variables, $1 \leq k_0 \leq n$, is given by:*

$$\sum_{k=k_0}^{n} B(n,k) = \sum_{k=k_0}^{n} \binom{n}{k} \left[B(k,k) + B^*(n-k,0) \cdot 2^{k+1} \sum \binom{k}{k_1, \ldots, k_r} \right].$$

In particular, the canalizing functions are counted by the following identity:

$$C_n = \sum_{k=1}^{n} B(n,k) = \sum_{k=1}^{n} \binom{n}{k} \left[B(k,k) + B^*(n-k,0) \cdot 2^{k+1} \sum \binom{k}{k_1, \ldots, k_r} \right].$$

In both equations, the last sum is taken over all compositions of k.

6. Concluding remarks and future work

Canalizing Boolean functions were inspired by structural and dynamic features of biological networks. In this article, we extended results on NCFs and derived a unique extended monomial form of *arbitrary* Boolean functions. This gave us a stratification of the set of n-variable Boolean functions by canalizing depth. In particular, this form encapsulates three invariants of Boolean functions: canalizing depth, dominance layer number and the non-canalizing core polynomial. By combining these three invariants, we obtained an explicit formula for the number of Boolean functions on n variables with depth k. We also introduced the notion of k-canalizing Boolean functions, which we believe to be a promising framework for modeling gene regulatory networks. Our stratification yielded closed formulas for the number of n-variable Boolean functions of canalizing depth k. Perhaps more valuable than the exact enumerations is the fact that now it is straightforward to derive asymptotics for the number of such functions as n and k grow large.

In future work, we will investigate well-known Boolean network models and compute the canalizing depth of the proposed functions. We are working on reverse-engineering algorithms that construct Boolean network models from partial data. In particular, how can one find the function with the maximum canalizing depth that fits that data, and whether the set of k-canalizing functions in the model space has an inherent algebraic structure. Progress has been made on these problems for general Boolean functions without paying attention to canalizing depth, and for NCFs. For example, for the general reverse-engineering problem, the set of feasible functions (i.e., the "model space") is a coset $f + I$ in the polynomial ring $\mathbb{F}_2[x_1, \ldots, x_n]$, where I is the ideal of functions that vanish on the data-set; see [HJ12]. Can we get more refined results by restriction to k-canalizing functions? The set of nested canalizing functions can be parametrized by a union of toric algebraic varieties [JL07]. It is relatively straightforward to show that the set of k-canalizing functions admits a similar parametrization, but it is not clear whether this has any actual utility for modeling.

Another avenue of current research extends the work in the electrical engineering community on the unate cascade functions. Recall that these are precisely the NCFs, and they are precisely the functions whose binary decision diagrams have minimum average path length, and this can be explicitly computed. Similarly, we can compute the minimum average path length of a binary decision diagram of a k-canalizing function.

Finally, much of the work in this paper should be able to be extended to multi-state (rather than Boolean) functions. As long as K is a finite field, then n-variable functions over K are polynomials in the ring $K[x_1, \ldots, x_n]$. The definition of an NCF was extended from Boolean to multi-state functions in [ML12], where the authors also enumerated these functions. Some of the proof techniques in this current paper specifically use the fact that $K = \mathbb{F}_2$, and it is not clear how well they would extend to general finite fields. However, there should absolutely be a stratification of multi-state functions by canalizing depth. The problem of enumerating k-canalizing multi-state functions seems to be challenging but still within reach.

References

[AO03] R. Albert and H.G. Othmer. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster. *J. Theor. Biol.*, 223(1):1–18, 2003.

[BB78] E.A. Bender and J.T. Butler. Asymptotic aproximations for the number of fanout-free functions. *IEEE T. Comput.*, 27(12):1180–1183, 1978.

[HJ12] F. Hinkelmann and A.S. Jarrah. Inferring biologically relevant models: nested canalyzing functions. *ISRN Biomathematics*, 2012, 2012.

[JL07] A.S. Jarrah and R. Laubenbacher. Discrete models of biochemical networks: The toric variety of nested canalyzing functions. In *Algebraic Biology*, pages 15–22. Springer, 2007.

[JM13] K. Jansen and M.T. Matache. Phase transition of boolean networks with partially nested canalizing functions. *Eur. Phys. J. B*, 86(7):1–11, 2013.

[JRL07] A.S. Jarrah, B. Raposa, and R. Laubenbacher. Nested canalyzing, unate cascade, and polynomial functions. *Physica D*, 233(2):167–174, 2007.

[JSK04] W. Just, I. Shmulevich, and J. Konvalina. The number and probability of canalizing functions. *Physica D*, 197(3):211–221, 2004.

[Kau69] S.A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, 22(3):437–467, 1969.

[Kau74] S. Kauffman. The large scale structure and dynamics of gene control circuits: an ensemble approach. *J. Theor. Biol.*, 44(1):167–190, 1974.

[KH07] F. Karlsson and M. Hörnquist. Order or chaos in boolean gene networks depends on the mean fraction of canalizing functions. *Physica A*, 384(2):747–757, 2007.

[KKBS14] J.G. Klotz, D. Kracht, M. Bossert, and S. Schober. Canalizing boolean functions maximize mutual information. *IEEE T. Inform. Theory*, 60(4):2139–2147, 2014.

[KLAL14] C. Kadelka, Y. Li, J.O. Adeyeye, and R. Laubenbacher. Nested canalizing functions and their networks. *arXiv:1411.4067*, 2014.

[KPST03] S. Kauffman, C. Peterson, B. Samuelsson, and C. Troein. Random boolean network models and the yeast transcriptional network. *Proc. Natl. Acad. Sci.*, 100(25):14796–14799, 2003.

[KPST04] S. Kauffman, C. Peterson, B. Samuelsson, and C. Troein. Genetic networks with canalyzing boolean rules are always stable. *Proc. Natl. Acad. Sci.*, 101(49):17102–17107, 2004.

[LAM+13] Y. Li, J.O. Adeyeye, D. Murrugarra, B. Aguilar, and R. Laubenbacher. Boolean nested canalizing functions: A comprehensive analysis. *Theor. Comput. Sci.*, 481:24–36, 2013.

[LDM12] L. Layne, E. Dimitrova, and M. Macauley. Nested canalyzing depth and network stability. *Bull. Math. Biol.*, 74(2):422–433, 2012.

[LLL+04] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang. The yeast cell-cycle network is robustly designed. *Proc. Acad. Natl. Sci.*, 101(14):4781–4786, 2004.

[LNC96] R. Lidl, H. Niederreiter, and P.M. Cohn. Encyclopedia of mathematics and its applications 20: Finite fields, 1996.

[MA05] A.A. Moreira and L.A.N. Amaral. Canalizing Kauffman networks: Nonergodicity and its effect on their critical behavior. *Phys. Rev. Lett.*, 94(21):218702, 2005.

[ML12] D. Murrugarra and R. Laubenbacher. The number of multistate nested canalyzing functions. *Physica D*, 241(10):929–938, 2012.

[Pei10] T.P. Peixoto. The phase diagram of random boolean networks with nested canalizing functions. *Euro. Phys. J. B*, 78(2):187–192, 2010.

[RDC14] C. Ray, J.K. Das, and P.P. Choudhury. On analysis and generation of some biologically important boolean functions. *arXiv:1405.2271*, 2014.

[RH13] R. Robeva and T. Hodge. *Mathematical concepts and methods in modern biology: using modern discrete models*. Academic Press, 2013.

[SD07] A. Szejka and B. Drossel. Evolution of canalizing boolean networks. *Euro. Phys. J. B*, 56(4):373–380, 2007.

[SK79] T. Sasao and K. Kinoshita. On the number of fanout-free functions and unate cascade functions. *IEEE T. Comput.*, 100(1):66–72, 1979.

[SK04] I. Shmulevich and S.A. Kauffman. Activities and sensitivities in boolean network models. *Phys. Rev. Lett.*, 93(4):048701, 2004.

DEPARTMENT OF MATHEMATICAL SCIENCES, CLEMSON UNIVERSITY, CLEMSON, SC 29634-0975, USA
E-mail address: qhe@g.clemson.edu, macaule@clemson.edu