On the number of biologically permitted logics

Thomas M. A. Fink
London Institute for Mathematical Sciences, Royal Institution, 21 Albermarle St, London W1S 4BS, UK

Networks of gene regulation are responsible for such complex feats as morphogenesis and cell programming. At a molecular level, they rely on local updates rules, or logics. But quantitative insights into how many and which logics are used have proved elusive. We analyze the number of biologically permitted logics, which we calculated in a recent paper using the composition of Boolean functions, and bound it with a simple function of the local connectivity. It confirms that the range of biological logics is highly restricted, which makes it easier to infer them through experiments.

Can the study of life lead to new mathematical theorems? One reason to hope so is that the very existence of life remains deeply mysterious [1]. Darwin's theory tells us that evolution is the result of mutation, selection and inheritance. But, from a physics point of view, we have essentially no understanding of how life got started in the first place [2]. Studies of artificial digital life that possess these key ingredients exhibit intricate dynamics and have even led to new approaches to optimization. But they have yet to exhibit the sorts of innovative leaps found in biological evolution.

Coming back down to earth and turning to the only life we actually know—biology—we are confronted with serious gaps in our knowledge. How do networks of gene regulation achieve complex feats such as morphogenesis [5] and cell programming? The economy of viruses suggest the existence of modular subroutines, but we have little understanding of the operating system of life.

We would like the study of life to lead to new types of mathematics for describing it, but how likely is this in practice? One reason to be hopeful is that mathematics tends to progress along the lines that mathematicians attend to [4]. Research areas that require new mathematics to advance, and for which there is a demand for progress, tend to influence what mathematics gets done.

The problem of identifying biologically permitted logics is just such an example. It is simple to state in biological terms, yet translates into a precise and seemingly important mathematical problem, namely, the number and degeneracy of Boolean functions under composition.

Why are these two problems equivalent? In genetic regulatory networks, genes talk to transcription factors and transcription factors talk to genes, but neither group talks to itself. This type of network structure is called bipartite, and it shows up in many contexts, such as the network of academic papers and their authors.

Genetic regulatory networks have been extensively modeled using Boolean networks [6]. In this model, genes are either on or off and the state of a node at time $t+1$ is a Boolean function of the states of its inputs at time t. A Boolean function is just a fixed update rule for what to do in response to what your inputs are doing, and throughout this paper I use the phrases Boolean function and logic interchangeably.

A major drawback of Boolean networks is that they assume that there is one species of node when in fact there are two. Genes can only interact indirectly via transcription factors. To capture this feature, a new model for modeling networks of gene regulation has emerged, called bipartite Boolean networks [12, 13]. To describe how a gene talks to other genes via the transcription factor middlemen, it's necessary to keep track of the number of second-nearest neighbors as well as neighbors. Then the Boolean functions of the transcription factors, which are themselves Boolean functions of genes, can be expressed as a composition of such functions. This is denoted with the shorthand

$$\{t_1, t_2, \dots, t_n\},$$

which I call the composition structure. For example, $\{2, 2\}$ is shorthand for $h(a, b, c, d) = f(g_1(a, b), g_2(c, d))$, where a, b, c and d are binary-valued input variables.

In a recent paper, my coauthor and I showed that a bipartite Boolean networks can be decomposed into two ordinary Boolean networks. Based on this, we derived an exact expression for the number of logics that are biologically valid, based on the composition of Boolean functions [3].

In this paper, we explore the properties of this quantity and what it tells us about the number and type of biologically permitted logics. Specifically, we show that the number of logics is tightly bound from below and above by

$$\frac{a(n)}{2^n}\prod_{i=0}^{n}(2^{2^{t_i}}-2) \leq c(t_1, \dots, t_n) \leq \frac{a(n)}{2^n}\prod_{i=0}^{n}2^{2^{t_i}}, \quad (1)$$

where $a(n)$ is the number of Boolean functions of n variables which depend on all n variables, that is,

$$a(n) = \sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}2^{2^i}. \quad (2)$$

We use these bounds to confirm that the fraction of biologically permitted logics is small, and that many compositions of logics lead to the same logic.

Boolean functions and two stepping stones

Before we derive our main result, let's review some general properties of Boolean functions and work out two stepping stones which will prove useful later. There are 2^{2^n} Boolean functions of n variables. For $n = 2$, for example, these are true, false, a, b, $\overline{a}$, $\overline{b}$ ab, $a\overline{b}$, $\overline{a}b$, $\overline{a}\overline{b}$, $a+b$, $a+\overline{b}$, $\overline{a}+b$, $\overline{a}+\overline{b}$, $ab+\overline{a}\overline{b}$ and $a\overline{b}+\overline{a}b$. In this notation, $\overline{a}$ means NOT a, ab means a AND b, and $a+b$ means a OR b. Two of these 16 functions depend on no variables, four depend on one variable, and 10 depend on two variables

For our first stepping stone, let $a(n)$ be the number of Boolean functions of n variables that depend on all n variables. By the principle of inclusion and exclusion, we can write $a(n)$ as the inverse binomial transform of 2^{2^n}, which is the expression given in (2). The first several $a(n)$ are 2, 2, 10, 218, 64594 (OEIS A000371 [15]). Our first stepping stone is that we can bound $a(n)$ from below and above:

$$2^{2^n} - n\,2^{2^{n-1}} \leq a(n) \leq 2^{2^n}. \tag{3}$$

The right bound follows from the definition of $a(n)$. The left bound can be deduced by showing that the magnitude of the $i-1$th term in (**??**) is less than half that of the ith term. Why is this helpful? Because it implies that the sum of the third ($i = 2$) term onwards, even were they to all have the same sign, could never add up to more than the second term. So, we need to show that

$$\binom{n}{i-1} 2^{2^{i-1}} \leq \frac{1}{2}\binom{n}{i} 2^{2^i},$$

which implies that

$$2i \leq (n-i+1) 2^{2^{i-1}},$$

where $i \leq n$. Since the smallest that $n-i+1$ can be is 1, we need only that $2i \leq 2^{2^{i-1}}$. It is indeed for all $i \geq 1$, establishing the left bound in (3).

Now let's turn to our second stepping stone. It turns out, as we shall see, that it is also useful to consider the inverse binomial transform of $a(n)$, namely,

$$u(n) = \sum_{i=0}^{n} (-1)^{n-i} \binom{n}{i} a(i). \tag{4}$$

The first several $u(n)$ are 2, 0, 8, 192, 63776 (OEIS A193247 [15]). Our second stepping stone is that, as with $a(n)$, we can bound $u(n)$ from below and above:

$$a(n) - n\,a(n-1) \leq u(n) \leq a(n). \tag{5}$$

To prove this, again we rely on showing that the magnitude of the $i-1$th term in (4) is less than half that of the ith term. We need to show that

$$\binom{n}{i-1} a(i-1) \leq \frac{1}{2}\binom{n}{i} a(i), \tag{6}$$

that is,

$$2i\,a(i-1) \leq (n-i+1) a(i), \tag{7}$$

where $i \leq n$. Inserting the upper and lower bounds from (3) into the left and right sides and rearranging, we find

$$2i \leq (n-i+1)\left(2^{2^{i-1}} - i\right).$$

Since the smallest that $n-i+1$ can be is 1, we need only $2i \leq 2^{2^{i-1}} - i$. It is for all $i \geq 3$, and inserting $i = 1$ and $i = 2$ into eq. (6) confirms the result for those values too.

Derivation of bounds

We are now ready to derive eq. (1), our main result. Consider a Boolean function of n inputs, which are themselves Boolean functions of $t_1, \ldots, t_n$ inputs. Our starting point is the main result of my recent paper on biologically permitted logics [3], namely, that the number of permitted logics is

$$c(t_1, \ldots, t_n) = \sum_{m=0}^{n} a(m) \sum_{\sigma_1 \ldots \sigma_m} \alpha_{\sigma_1} \ldots \alpha_{\sigma_n}, \tag{8}$$

where

$$\alpha_i = (2^{2^i} - 2)/2.$$

The second sum adds up the product of all m-tuples of the α_i. For $m = 0$, the sum is over the null set and is taken to be 1. For example,

$$\begin{aligned} c(i) &= 2 + 2\,\alpha_i, && (9)\\ c(i,j) &= 2 + 2(\alpha_i + \alpha_j) + 10\,\alpha_i\alpha_j, && (10)\\ c(i,j,k) &= 2 + 2(\alpha_i + \alpha_j + \alpha_k) && (11)\\ &+ 10\big(\alpha_i\alpha_j + \alpha_j\alpha_k + \alpha_i\alpha_k\big) + 218\,\alpha_i\alpha_j\alpha_k. \end{aligned}$$

Composition $(k_1, \ldots, k_n)$	*Lower bound*	*True value*	*Upper bound*	*Permitted fraction*
$\{1,1\}$	10	16	40	1
$\{1,2\}$	70	88	160	0.34
$\{1,3\}$	1,270	1528	2560	0.023
$\{2,2\}$	490	520	640	0.0079
$\{2,3\}$	8,890	9160	10,240	2.1×10^{-6}
$\{3,3\}$	161,290	161,800	163,840	8.8×10^{-15}
$\{1,1,1\}$	218	256	1744	1
$\{1,1,2\}$	1526	1696	6976	0.026
$\{1,1,3\}$	27,686	30,496	111,616	7.1×10^{-6}
$\{1,2,2\}$	10,682	11,344	27,904	2.6×10^{-6}
$\{1,2,3\}$	193,802	204,304	446,464	1.1×10^{-14}
$\{2,2,2\}$	74,774	76,288	111,616	4.1×10^{-15}
$\{2,2,3\}$	1,356,614	1,375,168	1,785,856	4.0×10^{-33}
$\{2,3,3\}$	2.46×10^7	24,792,448	2.86×10^7	2.1×10^{-70}
$\{3,3,3\}$	4.46×10^8	447,032,128	4.57×10^8	3.3×10^{-146}

TABLE I: The composition structure $\{t_1, \ldots, t_n\}$ indicates a Boolean function of n inputs, which are themselves Boolean functions of $t_1, \ldots, t_n$ inputs. The true value of the number of pemitted logics is given by eq. (8) and the bounds are given by eq. (1). The ratio of the true value and the number of Boolean functions of $t_1 + \ldots + t_n$ variables is very small for most structures.

Explicit values of these are given in Table I for i, j and k ranging from 1 to 3.

Eq. (8), while exact, does not provide much intuition for how this function behaves over different composition structures because it involves a sum over all subsets of a set. We would like to express it in simpler terms to get a better feel for its behavior. Our goal is to tightly bound it from below and above, as shown in eq. (1).

We can immediately bound eq. (8) from below by taking just the last ($m = n$) term. This gives

$$\frac{a_n}{2^n}\prod_{i=0}^{n}(2^{2^{t_i}} - 2) \le c(t_1, \dots, t_n). \tag{12}$$

But is this bound any good? This depends on how much the other terms contribute to the total, which is not at all clear.

To find out, we need to bound $c(t_1, \dots, t_n)$ from above, which is harder. Our first step is to re-express eq. (8) in terms of products of $\beta_i = 2^{2^i}/2$ instead of $\alpha_i = (2^{2^i} - 2)/2$. For example, with a little algebra, we find

$$\begin{aligned} c(i) &= 2\,\beta_i, && (13)\\ c(i,j) &= 8 - 8\,(\beta_i + \beta_j) + 10\,\beta_i\beta_j, && (14)\\ c(i,j,k) &= -192 + 200\,(\beta_i + \beta_j + \beta_k) && (15)\\ &\quad -208\,(\beta_i\beta_j + \beta_j\beta_k + \beta_i\beta_k) + 218\,\beta_i\beta_j\beta_k. && \end{aligned}$$

Notice that the coefficients in eqs. (13–15) are different for each line, unlike the case for eqs. (9–11). These new expressions of c in terms of the β_i are less "natural" than the ones in terms of the α_i. Nevertheless, let us put up with the extra complexity in the hope of finding our upper bound.

By analogy with the expansion of $(1-x)^n$ into an alternating series of powers of x, we can express $c(t_1, \dots, t_n)$ as

$$c(t_1, \dots, t_n) = \sum_{m=0}^{n}(-1)^n u(n,m) \sum_{\sigma_1 \dots \sigma_m} \beta_{\sigma_1} \dots \beta_{\sigma_m}, \tag{16}$$

where the coefficients $u(n,m)$ satisfy

$$u(n,m) = \sum_{i=m}^{n}(-1)^{n-i+m}\binom{n-m}{i-m}a(i). \tag{17}$$

Notice that $u(n,m)$ is similar to $u(n)$ in our second stepping stone, but it differs in that it sums only the top $n-m$ of the $a(i)$ terms, and the binomial coefficients are shifted downwards. The first several $u(n,m)$ are

2;
0, −2;
8, −8, 10;
192, −200, 208, −218;
63,776, −63,968, 64,168, −64,376 64,594.

Note that $u(n,0) = u(n)$ and $u(n,n) = (-1)^n a(n)$. Just like we were able to bound $u(n)$ from below and above, we can do the same for the magnitude of $u(n,m)$:

$$a(n) - n\,a(n-1) \le u(n,m) \le a(n). \tag{18}$$

This proof of this is along similar lines to that of the bounds for $u(n)$. We will make use of (18) soon.

But before we do, we can make (16) tidier. Let B_m be the average value of the product of m randomly selected choices of the β_i. Then the sum on the right of (16) can be expressed as $\binom{n}{m}B_m$. Aesthetics isn't the only reason for doing this. While the sum of the products of the m-tuples is not an increasing function of m, B_m is. This will prove useful in a moment. But for now we can write

$$c(t_1, \dots, t_n) = \sum_{m=0}^{n}(-1)^n u(n,m)\binom{n}{m}B_m. \tag{19}$$

We want to show that this is bounded from above by the last ($m = n$) term. To prove this, we need to show that all but the last terms sum to at most zero, that is,

$$\sum_{\substack{m=0\\ n-m \text{ even}}}^{n-1} u(n,m)\binom{n}{m}B_m \le \sum_{\substack{m=0\\ n-m \text{ odd}}}^{n-1} u(n,m)\binom{n}{m}B_m.$$

Inserting the upper and lower bounds from (18) into the left and right sides and rearranging, we find

$$a(n)\sum_{\substack{m=0\\ n-m \text{ even}}}^{n-1}\binom{n}{m}B_m \le (a(n) - n\,a(n-1))\sum_{\substack{m=0\\ n-m \text{ odd}}}^{n-1}\binom{n}{m}B_m. \tag{20}$$

Recall that B_m increases with m. This means that we can set the β_{σ_i} to 1, the validity of which implies the validity of the above. So we need only show

$$a_n\sum_{\substack{m=0\\ n-m \text{ even}}}^{n-1}\binom{n}{m} \le (a_n - n\,a_{n-1})\sum_{\substack{m=0\\ n-m \text{ odd}}}^{n-1}\binom{n}{m}$$

Since the even binomial coefficients and the odd binomial coefficients both sum to 2^{n-1}, we need only show

$$a(n)(2^{n-1} - 1) \le (a_n - n\,a(n-1))\,2^{n-1},$$

that is,

$$a(n+1) \ge (n+1)2^n a(n). \tag{21}$$

Inserting the upper and lower bounds from (3) into the left and right sides and rearranging, we find

$$2^{2^n} \ge (2^n + 1)(n+1),$$

which is true for $n \ge 2$. For $n = 1$, (21) is true by inspection, completing the proof of the right side of (1).

There are two limiting regimes for $c(t_1, \ldots, t_n)$ which provide some intuition for how the function behaves. The first is a composition with few inputs, each of which has many inputs. The second is one with many inputs, each of which has few inputs.

For $n = 2$ and arbitrary i and j, by eq. (8)

$$c(i, j) \leq 10/2^2 \cdot 2^{2^i} 2^{2^j}.$$

For arbitrary n and $t_i = 2$,

$$c(2, \ldots, 2) = \sum_{m=0}^{n} 7^m \binom{n}{m} a(n, m) \leq 8^n a(n).$$

The equality is by eq. (8) and the inequality by eq. (1).

Discussion

Our lower and upper bounds, which are described in Table I, are very tight. Dividing the left and right sides of eq. (1) by the right side, the bounds are within

$$\prod_{i=1}^{n} \left(1 - 2/2^{2^{t_i}}\right)$$

of the true result. For example, for the composition structure $\{3, 3\}$, the logarithm of the lower and upper bounds differ by 0.1%.

There are two aspects of our upper bound to notice. The first is that the composition of logics is highly restricted. The number of logics of $t_1 + \ldots + t_n$ variables is $2^{2^{t_1+\ldots+t_n}}$. But the number of distinct logics is $2^{2^n} 2^{2^{t_1} \ldots 2^{t_n}}$. The base-2 logarithm of the fraction of logics that are valid is less than 2^n plus the sum of 2^{t_i} minus the product of 2^{t_i}, that is, $2^n + 2^{t_1} + \ldots + 2^{t_n} - 2^{t_1+\ldots+t_n}$.

The second thing to notice is that the composition of logics is a many-to-one input-output map. To see why, consider all of the ways of assigning Boolean functions to f and $g_1, \ldots, g_n$. This is $2^{2^n} 2^{2^{t_1}} \ldots 2^{2^{t_n}}$. But we know from our upper bound in eq. (1) that these map to at most $a(n)/2^n\, 2^{2^{t_1}} \ldots 2^{2^{t_n}}$ Boolean functions of the $t_1, \ldots, t_n$ variables. So the average degeneracy—the number of logic compositions that map to the same logic—is at least $2^n 2^{2^n}/a(n)$. Since $a(n)$ is bounded from above by 2^{2^n}, the average degeneracy is at least 2^n. Such redundancy in how a particular logic is coded could make them resistant to mistakes [20].

But what makes this more interesting is that this degeneracy appears to be far from uniform. Some logics show up much more frequently than others. Computer enumeration suggests that simpler logics, in the sense of depending on fewer of the $t_1 + \ldots + t_n$ inputs, tend to show up more. For example, for the composition structure $\{2, 2\}$, 22% of the 4,096 compositions map to true and false, and 30% map to 28 other relatively simple logics. If this effect applies generally to other composition structures, it would suggest that not only are biologically permitted logics restricted, but they tend to be simple as well. Both properties would make the reverse engineering of genetic regulatory logics through experiments easier.

The study of the composition of Boolean functions is motivated by how biology processes information at a genetic level. But it is also a beautiful and fundamental mathematical question in its own right. That it seems to have received little attention until motivated by the study of life suggests that new theorems are indeed around the corner.

This research was supported by a grant from Bit.Bio. I acknowledge hospitality from the King Ranch near Bandera, Texas and R. Hannam for useful discussions.

[1] E. Schroedinger *What is life?* (Cambridge University Press, Cambridge, 2008).

[2] J. England. The thermodynamics of reproducing systems, J Chem Phys 90, 1332 (2021).

[3] T. M. A. Fink and R. Hannam, Composition of Boolean functions restricts biologically permitted logics, submitted to Phys Rev Lett (2021).

[4] M. Reed, Why is mathematical biology so hard?, Notices AMS 51, 338 (2004).

[5] A. M. Turing, The chemical basis of morphogenesis, Philos T Roy Soc B 237, 37 (1952).

[6] S. Huang, G. Eichler, Y. Bar-Yam, and D. E. Ingber, Cell fates as high-dimensional attractor states of a complex gene regulatory network, Phys Rev Lett 94, 128701 (2005).

[7] S. Bilke and F. Sjunnesson, Stability of the Kauffman model, Phys Rev E 65, 016129 (2001).

[8] J. E. Socolar and S. A. Kauffman, Scaling in ordered and critical random Boolean networks, Phys Rev Lett 90, 068702 (2003).

[9] B. Samuelsson and C. Troein, Superpolynomial growth in the number of attractors in Kauffman networks, Phys Rev Lett 90, 098701 (2003).

[10] I. Shmulevich and S. A. Kauffman, Activities and sensitivities in Boolean network models, Phys Rev Lett 93, 048701 (2004).

[11] C. Buccitelli and M. Selbach, mRNAs, proteins and the emerging principles of gene expression control, Nat. Rev. Genet 21,630 (2020).

[12] R. Hannam, R. Kuhn, and A. Annibale, Percolation in bipartite Boolean networks and its role in sustaining life, J Phys A 52, 334002 (2019).

[13] R. Hannam, Cell states, fates and reprogramming, Ph.D. thesis, King?s College London (2019).

[14] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, Random graphs with arbitrary degree distributions and their applications, Phys Rev E 64, 026118 (2001).

[15] N. J. A. Sloane, editor, The On-Line Encyclopedia of Integer Sequences, published electronically at https://oeis.org, 2021.

[16] S. J. Engle and D. Puppala, Perspective Integrating Human-Pluripotent Stem Cells into Drug Development, Stem Cell 12,669 (2013).

[17] J. L. Payne and A. Wagner, Mechanisms of mutational robustness in transcriptional regulation, Front Genet 6, 322 (2015).

[18] S. E. Ahnert and T. M. A. Fink, Form and function in generegulatory networks J. R. Soc. Interface13, 20160179 (2016).

[19] K. Dingle, C. Q. Camargo, and A. A. Louis, Input-output maps are strongly biased towards simple outputs, Nat. Commun. 9 (2018).

[20] K. Raman and A. Wagner, The evolvability of programmable hardware, J. R. Soc. Interface 8, 269 (2011).