Article

Highly-Sensitive Measure of Complexity Captures Boolean Networks Regimes and Temporal Order More Optimally

Manuel de J. Luevano and Alejandro Puga *

Posted Date: 24 July 2024

doi: 10.20944/preprints202403.1569.v2

Keywords: Random Boolean Networks; Entropy; Algorithmic Complexity; Compressibility

Article

Highly-Sensitive Measure of Complexity Captures Boolean Networks Regimes and Temporal Order More Optimally

Manuel de J. Luevano †,‡ **and Alejandro Puga** *,†,‡

Unidad Académica de Física, Universidad Autónoma de Zacatecas, Zacatecas, Mexico; manuel.luevano@fisica.uaz.edu.mx

* Correspondence: apuga@fisica.uaz.edu.mx; Tel.: +52-492-924-1314

† Current address: Calzada Solidaridad esq. Paseo, La Bufa s/n C.P. 98060 Zacatecas, Zac. México.

‡ These authors contributed equally to this work.

Abstract: In this work, several random Boolean networks (RBN) are generated and analyzed from two characteristics: their time evolution diagram and their transition diagram. For this purpose, its randomness is estimated using three measures, of which Algorithmic Complexity is capable of both a) revealing transitions towards the chaotic regime in a more marked way, and b) disclosing the algorithmic contribution of certain states to the transition diagram and their relationship with the order they occupy in the temporal evolution of the respective RBN. The results obtained from both types of analysis are useful for the introduction of both Algorithmic Complexity and Perturbation Analysis in the context of Boolean networks, and their potential applications in regulatory network models.

Keywords: random Boolean networks; entropy; compressibility; algorithmic complexity; perturbation analysis

1. Introduction

Discrete-time models are used in biology to capture the dynamic nature of biological systems. In particular, *Boolean networks* are models that also assume discrete values for their variables, and allow us to have a qualitative view of the dynamics of a system. The origin of Boolean networks can be traced to Kauffman's work [1], where he introduces *random genetic nets* to test the hypothesis that modern organisms are randomly assembled molecular automata. Later works, such as that of Kauffman [2] where maps continuous biochemical control networks to their discrete analogue, and that of Thomas [3] who attempts to formalize genetic situations through models of complex control circuits, established the usefulness of Boolean network models to study biological regulatory systems.

Many studies highlight the usefulness of models of Boolean networks to analyze regulatory networks. Some examples are: a) the dynamic properties of the core network that controls the mammalian cell cycle [4], b) simulation of the stomatal response in plants with a signal transduction network model [5], c) virulence and response to pathogens resulting from the interactions between bacteria and the immune components of an infected host [6], d) the modeling of the *lac* operon in *Escherichia coli* [7], and e) the modeling of the network of regulatory interactions that occur through various stages of embrionic development of the *Drosophila* [8]. Most importantly, such type of models of regulation and signalling serves to study a multitude of factors in an integrated way, rather than studying them in isolation.

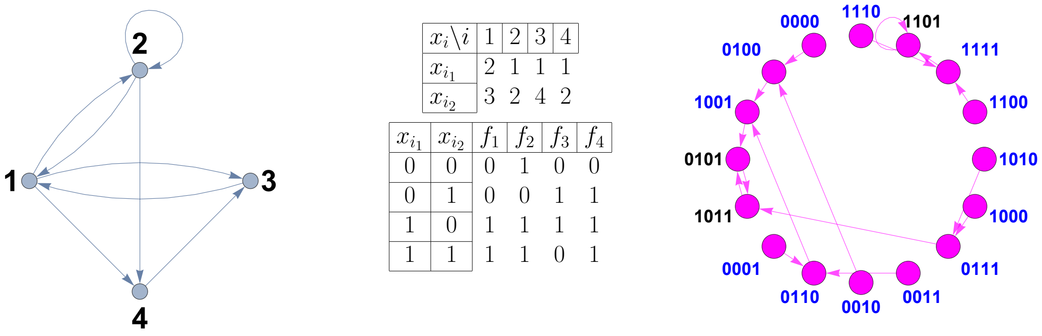
In this paper, we present and compare various methodologies for analyzing the dynamics of some Boolean networks under the simplest of assumptions. First, we analyze the regimes of certain Boolean networks; then, we study the transitions between states in some simple Boolean networks applying perturbations. This work intends to introduce some useful tools for potential applications in the modeling and analysis of genetic regulatory networks, as dynamical systems with a causal regulatory network that can be found via gene perturbations.

2. Background

Boolean networks, also known as N-k networks, are characterized by two parameters: the number of nodes N and the in-degree k. The state that each node can acquire is only one of two possible states -0 and 1, and is determined by a set of rules called *boolean functions*,

$$x_i(t+1) = f_i\big(x_{j_1}(t), x_{j_2}(t), \cdots, x_{j_k}(t)\big) \tag{1}$$

where the states x_i of the nodes at time $t+1$ are determined by the Boolean functions f_i that depend on certain nodes at time t (in this work, we make the assumption of *synchronous update*; but there exist more flexible modeling frameworks that incorporate randomness in the duration of processes, see [9]). The arguments of Boolean functions, as well as their order, are specified by a *connectivity graph* and a *connectivity matrix*. Boolean functions are represented by a *truth table*. A Boolean network has a *transition diagram* between all 2^N possible states (see Figure 1).



$x_i \backslash i$	1	2	3	4
x_{i_1}	2	1	1	1
x_{i_2}	3	2	4	2

x_{i_1}	x_{i_2}	f_1	f_2	f_3	f_4
0	0	0	1	0	0
0	1	0	0	1	1
1	0	1	1	1	1
1	1	1	1	0	1

Figure 1. A specific Boolean network. Left: connectivity graph with $N = 4$ and $k = 2$. Medium: connectivity matrix and truth table. Right: transition diagram between all $2^4 = 16$ possible states, constructed from the connectivity matrix and the truth table; steady states or attractors are depicted in black.

2.1. Random Boolean Networks

When Boolean functions are extracted from a random sample of 0s and 1s they become *random boolean functions*, and the network becomes a *random boolean network* (RBN) [10]. Boolean functions can be constructed with a uniform probability distribution. Or they can be constructed from a binomial probability distribution with parameter p, this parameter being therefore the probability that each of the Boolean functions returns a "1" as output. This parameter will also determine the type of system:

- Chaotic, if $k \cdot 2p \cdot (1-p) > 1$
- Ordered, if $k \cdot 2p \cdot (1-p) < 1$
- At the edge of chaos, if $k \cdot 2p \cdot (1-p) = 1$

In the last case, $k_c = \frac{1}{2p\cdot(1-p)}$ is known as the *critical in-degree*. Since these are probabilities, this is only true when the number of nodes N is very large. Figure 2 shows time evolution diagrams for three RBNs with 500 nodes and in-degree $k = 5$; two of them are in the ordered regime ($p = 0.1$ and $p = 0.9$) and the other one ($p = 0.5$) is in the chaotic regime. For the RBN with $p = 0.1$ there is a 90 percent probability that the Boolean functions deactivate their respective node; this can be confirmed in its time evolution diagram where there are many more deactivated nodes and in the rapid fall in a limit cycle attractor. Similarly, for the RBN with $p = 0.9$ there is a 90 percent probability that the Boolean functions activate their respective node, and a rapid fall in a limit cycle attractor is observed. On the other hand, it can be seen that the RBN with $p = 0.5$ is a chaotic system.

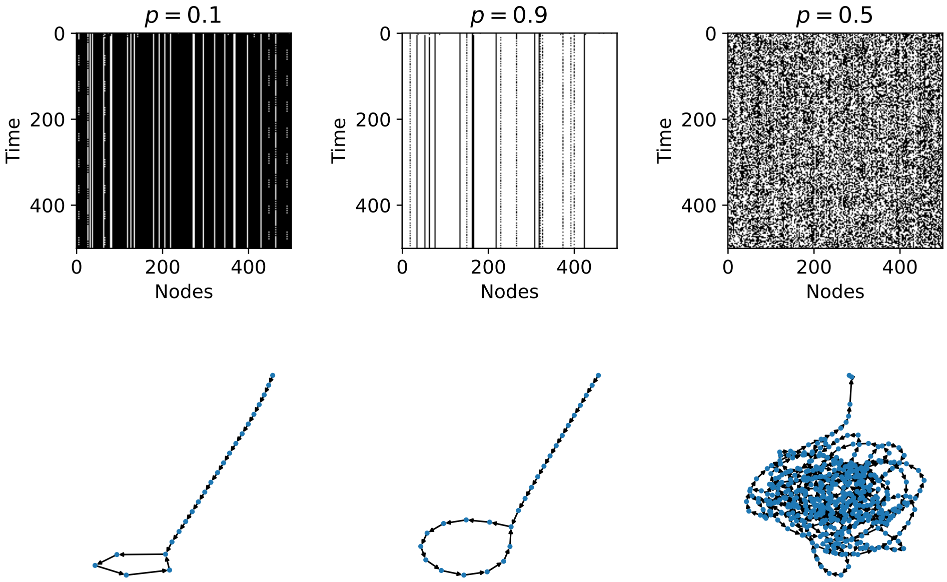


Figure 2. Time evolution diagrams (top) and their respective state transition graphs (bottom) for some RBNs with $N = 500$ nodes and $k = 5$. The RBNs share the same initial state and the same connectivity matrix, both generated randomly from uniform probability distributions.

The time evolution and transition diagrams such as in Figure 2 allow us to analyze qualitatively the regimes in RBNs; for a more quantitative approach, we need a measure that is capable of capturing such regimes and the transitions between them. In the next section, we introduce some methods of measure on space-time patterns as candidates for the task at hand (for an alternative classification of chaotic/ordered RBNs, which is made from measurements on the basins of attraction of the state transition graphs, see [11]).

3. Methodology

The act of explaining what our eyes see has always been a human need. The act of trying to understand reality, which some have called science, common sense, etc., is nothing more than classifying randomness. When we discover the mechanism behind a phenomenon, we determine that it is not random and has a specific cause. To do this, a phenomenon can be abstracted to a chain of symbols and we call that "the object". However, measuring randomness is a challenge, and there are several measures at our disposal. In the next subsections, we review some of these measures.

3.1. Information Theory and Entropy

Information Theory [12] is a branch of mathematics that deals with the quantification and communication of information. One of the fundamental concepts in information theory is the concept of entropy. In particular, the Shannon entropy is a measure of the amount of uncertainty or information content in a random variable.

The Shannon entropy $H(X)$ of a discrete random variable X with probability mass function $p(x)$ is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \tag{2}$$

where $\mathcal{X}$ is the set of all possible values of X. The logarithm is usually taken to base 2, so the entropy is measured in bits.

The Shannon entropy has a number of important properties, including:

- It is always non-negative, $H(X) \geq 0$.
- It is maximized when all the possible values of X are equally likely, i.e., when $p(x) = 1/|\mathcal{X}|$ for all $x \in \mathcal{X}$. In this case, the entropy is $\log_2|\mathcal{X}|$.
- It is minimized when X is a deterministic function of another random variable Y, i.e., when there exists a function f such that $X = f(Y)$ with probability 1.

In the last case, $H(X|Y) = 0$, since there is no uncertainty or information content in X beyond what is already contained in Y.

Shannon entropy is a useful tool in a variety of applications, including coding theory, data compression, and cryptography [13]. The Shannon entropy indicates how much "surprise" or "unexpectedness" there is in the outcomes of a random variable. The higher the entropy, the more uncertain or unpredictable the random variable is, and the more information is needed to describe or communicate its outcomes. Conversely, the lower the entropy, the more certain or predictable the random variable is, and the less information is needed to describe or communicate its outcomes.

3.2. Compression Algorithms and Compressibility

A *lossless compression algorithm* is any encoding procedure that aims to represent a certain amount of information using or occupying a smaller space, making an exact reconstruction of the original data possible. An example is the Lempel-Ziv-Welch (LZW) algorithm [14], a dictionary-based lossless compressor. The algorithm uses a dictionary to map the sequence of symbols to codes that take up less space. The LZW algorithm performs well, especially with files that have repetitive sequences. Another example of a lossless compression algorithm is the free and open-source file compression program called BZip2, which uses the Burrows-Wheeler algorithm [15], which only compresses individual files; It relies on standalone external utilities for tasks such as handling multiple files, encryption, and file splitting.

From the lossless compression of a piece of data we can define its *compressibility rate* as the length of the codified data divided by the length of the original data. This is an intuitive measure of randomness because we expect that the more random a piece of data or string is, the more difficult to compress it.

3.3. Algorithmic Complexity and BDM

Program size complexity, also known as *algorithmic complexity*, is a measure that quantifies algorithmic randomness. The algorithmic complexity (AC) or *algorithmic information content* of a string s is defined as the length of the shortest computer program p that is executed in a prefix-free Universal Turing Machine and generates s as an output [16,17]:

$$K(s) = \min\{|p| : U(p) = s\} \tag{3}$$

The difference between the original length and the minimum program size determines the complexity of data s. A string s is said to be random if $K(s)$ (in bits) $\sim |s|$ (for a more intuitive measure of complexity that combines the notions of algorithmic information content and time, see the concept of Charles Bennett's logical depth [18]).

A technical drawback is that K is not computable due to the Halting problem [19], since we can't always find the shortest programs in a finite time without having to run all the computer programs, which means having to wait forever in case they never stop. Nevertheless, it is possible to randomly produce programs that produce a specific result to estimate its AC from above; for this purpose, the *algorithmic probability* $m(s)$ of a binary string s is defined as the sum over all (prefix-free) programs with which a (prefix-free) Universal Turing Machine generates s and stops:

$$m(s) = \sum_{p:U(p)=s} \frac{1}{2^{|p|}} \tag{4}$$

The *Coding Theorem Method* (CTM) [20,21] further establishes the connection between $m(s)$ and $K(s)$,

$$|-\log_2 m(s) - K(s)| < c \tag{5}$$

where c is a fixed constant, independent from s. This theorem implies [22] that the output frequency distribution of random computer programs to approximate $m(s)$ can be converted to estimates of $K(s)$ using

$$K(s) = -\log_2(m(s)) + O(1) \tag{6}$$

Finally, the *Block Decomposition Method* (BDM) [23] is a powerful tool to estimate algorithmic complexity. This method allows the decomposition of a large system in smaller and more manageable pieces, facilitating the estimation of its total complexity. By dividing the data into fragments and precomputing CTM calculations on these fragments [24,25], BDM provides a measure that lies between entropy and algorithmic complexity, allowing a better understanding of the structure and behavior of the system under study. In general, for w-dimensional objects, BDM is defined as

$$\text{BDM}(X, \{x_i\}) = \sum_{(r_i, n_i) \in \text{Adj}(X)_{\{x_i\}}} [\text{CTM}(r_i) + \log n_i] \tag{7}$$

where the set $\text{Adj}(X)_{\{x_i\}}$ is composed of the pairs (r_i, n_i), r_i is an element from the decomposition of X (specified by a partition $\{x_i\}$, where x_i is a submatrix of X) and n_i is the multiplicity of each r_i.

3.4. Perturbation Analysis

It is worth asking whether the randomness measures outlined above can serve us for more than simply determining how random an object is or how chaotic its temporal evolution is. Specifically, we can ask if a robust measure of randomness can help us characterize or even control a dynamic system.

The *perturbation analysis* or *reprogrammability analysis* is based on the change in the randomness of a dynamic system subjected to perturbations. Its purpose is to track in detail the changes in algorithmic information content (estimated by local observations) produced by natural or induced perturbations in evolving complex open systems. Algorithmic models can be derived from partial observations, and the algorithmic probability that these models produce the behavior of the system under analysis can be estimated. Perturbation analysis can also help us reconstruct space-time diagrams of dynamic systems [26,27].

A simple example will allow us to illustrate the basic concepts of perturbation analysis. Let G be a graph with a set of nodes $V(G)$ and a set of vertices $E(G)$. The dynamics of G can be defined by transitions between different states. Let $G' = G \backslash e$ denote the operation of removing a vertex e from G. The difference $\text{AID} := K(G) - K(G \backslash e)$ would be the shared mutual algorithmic information, or *algorithmic information dynamics*, of G and $G \backslash e$, where it is considered that G is a time-dependent system and that G' can be the result of the temporal evolution of G. We have the following cases:

- $|\text{AID}| \leq \log_2 |V(G)|$. The vertex e is contained in the algorithmic description of G, and can be recovered from the latter.
- $|\text{AID}| \sim \log_2 |V(G)|$. The vertex e does not contribute to the description of G. The relationship between G and G' is causal.
- $|\text{AID}| > \log_2 |V(G)|$. G and G' do not share causal information. Removing e results in a loss.
- $|\text{AID}| > |V(G)|$. There are two possibilities: a) G' cannot be explained by G alone, that is, G' is not algorithmically contained in G, and therefore e is a fundamental part or generative causal mechanism of G; and b) e is not part of G but it is noise.

In the first two cases, it is said that element e is not essential in the explanation of G but it is an element that is produced by the natural course of evolution of G. In the last two cases, if $\text{AID} < 0$, element e is said to make a high *causal contribution* since its removal moves G more toward randomness. And in the last case, if AID > 0, then the removal of element e moves G away from randomness (or towards simplicity), so it is unlikely that element e is part of the generating mechanism of G, i.e., it is more likely that element e is noise or part of another system.

In this work, a basic perturbation analysis will be carried out on the state transition graph of some RBNs using the randomness measures already outlined in the previous subsections.

4. Results and Discussion

In this section we present and discuss some results about the time evolution of RBNs whose connectivity graphs were generated like this: for each of the N nodes, select one node k times as an adjacent node following a uniform probability distribution over all N nodes. We used the randomness measures of Section 3 over boolean functions (truth tables) and time evolution diagrams. Entropy was calculated from the relative frequencies of 0s and 1s. For the calculation of compressibility, an array was first converted to a one-dimensional string concatenating the rows from top to bottom, and then such string was compressed using the LZW algorithm. Finally, Algorithmic Complexity for an array was estimated through the Block Decomposition Method (for the estimation of AC by BDM we used the `pybdm` library [28])

4.1. RBNs with $N = 500$ Nodes

Figure 3 presents results on simulations of RBNs with $N = 500$ and $k = 5$ that share the same initial state and the same connectivity matrix, both generated from uniform probability distributions. At the top of Figure 3 there are randomness estimates, for both the Boolean functions and the time evolution diagrams, as a function of parameter p. The results concerning the Boolean functions show "causality gaps" between the entropy/LZW estimates and the AC estimates using the BDM, and could be anticipated since the Boolean functions were constructed from a binomial probability distribution with parameter p (see [23]). The causality gaps are more noticeable in the randomness estimates of the time evolution diagrams; it can even be seen that the AC estimation is capable of revealing the transition to the chaotic regime on both sides of the ordered regime ($p = 0.113$ and $p = 0.887$), while both entropy and LZW are insensitive to such transition (beyond some small fluctuations) and give results from which it can only be suggested that the evolution diagrams arise from a binomial probability distribution. For greater statistical rigor, estimates of average algorithmic complexity from samples of size 10 of RBNs generated with a fixed value of p are shown at the bottom of Figure 3; as in the upper part, we can see the transitions to the chaotic regime.

Next, Figure 4 shows some series of BDM (Boolean function) vs p, and BDM (evolution diagram) vs p, obtained for each specific value of in-degree k. In this case, p takes on 41 values evenly spaced in the interval $[0.0, 0.5]$. Simulations were carried out with connectivity matrices generated randomly from uniform probability distributions, with the RBNs for a specific k value sharing the same connectivity matrix. In each of the series, we can notice a sudden growth of the time evolution diagram's BDM starting from a certain p value (marked with vertical lines). Such p values are the critical p values that point out the transition to the critical regime in each series, as can be confirmed by comparison with the theoretical p values obtained from the chaos condition in RBNs (see Table 1). We can also notice that the critical p value increases when going from one series to another with lower k, following the chaos condition in RBNs. Furthermore, as expected, in each series it is observed that an increase in BDM of the evolution diagrams goes hand in hand with an increase in BDM of the Boolean functions; it can also be observed that the series with greater k have greater BDM values in the time evolution diagrams.

The results presented up to this point were obtained from simulations that consider connectivity matrices generated from uniform probability distributions. However, it is possible that non-uniform probability distributions introduce a change in the dynamics and therefore also introduce changes in the critical points. Although considerations about non-uniform distributions are beyond the objectives of this work, Figure 5 shows the results obtained by repeating the simulations carried out for Figure 3, but this time the connectivity matrix was generated from a binomial probability distribution with $P = 0.5$ and $n = 499$ (that is, the nodes of the Boolean network are labeled from 0 to 499 and the quantity that follows the binomial distribution is the probability that one of these nodes is chosen to determine each of the entries in the connectivity matrix; the condition $P = 0.5$ implies that the nodes with the highest probability of being chosen are nodes 249 and 250). In this case, transitions towards the chaotic regime continue to be observed and these can be revealed more noticeably by AC estimates; although, unlike the results presented in Figure 3, the critical values of p do not coincide with the theoretical ones (indicated with dotted lines in Figure 5) expected from a connectivity matrix constructed from a uniform probability distribution. The latter shows a change in the dynamics and the critical points when taking into account non-uniform probability distributions.

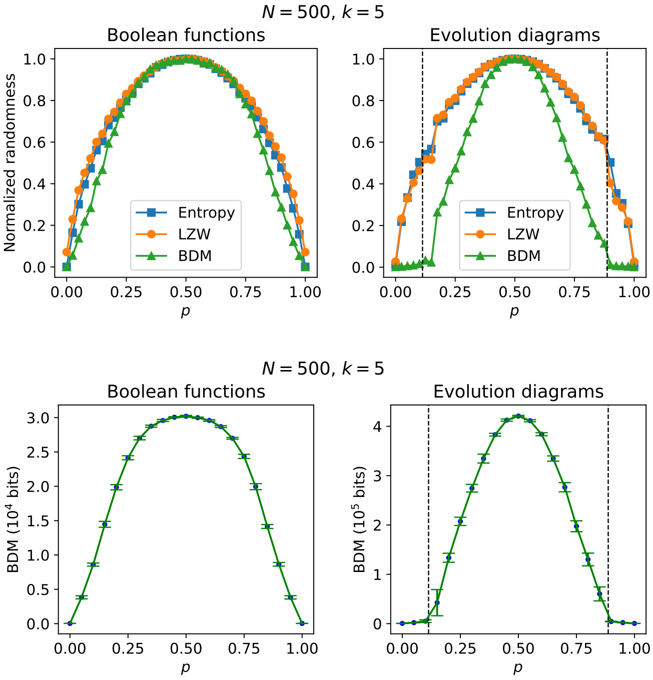


Figure 3. Top: randomness estimates of RBNs with $N = 500$ and $k = 5$ as a function of the parameter p. Bottom: estimates of average algorithmic complexity from samples of size 10. Vertical dotted lines represent the theoretical critical values of $p = 0.113$ and $p = 0.887$ at the edge of chaos. All RBNs share the same initial state and the same connectivity matrix, both generated randomly from uniform probability distributions.

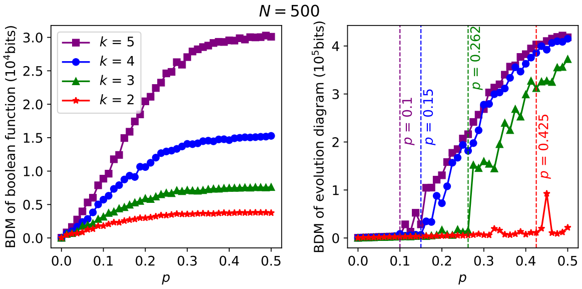


Figure 4. Left: Plots of boolean functions' BDM vs p. Right: Plots of time evolution diagrams' BDM vs p. Vertical dotted lines mark the critical p values in each series. All RBNs share the same initial state, and all RBNs for a given in-degree k share the same connectivity matrix; both types of objects were generated randomly from uniform probability distributions.

Table 1. Comparison of critical p values for the RBNs simulated in Figure 4.

k	**Critical p value in Figure 4**	**Theoretical critical p value**
5	0.100	0.113
4	0.150	0.146
3	0.262	0.211
2	0.425	0.50

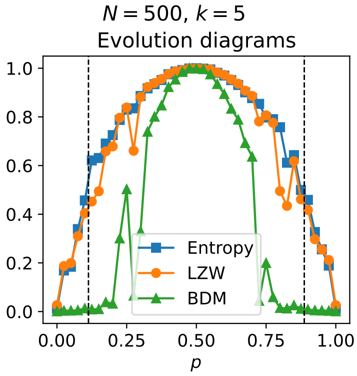


Figure 5. Randomness estimates of RBNs with $N = 500$ and $k = 5$ as a function of the parameter p. Vertical dotted lines represent the theoretical critical values of $p = 0.113$ and $p = 0.887$ at the edge of chaos. All RBNs share the same initial state and the same connectivity matrix. The connectivity matrix was generated randomly from a binomial probability distribution with $p = 0.5$ and $n = 499$.

4.2. RBNs with $N = 10$ and $k = 5$

In this subsection, some RBNs with 10 nodes and *in-degree* of 5 are analyzed from their complete transition diagram - which have $2^N = 1024$ possible states. In Figure 6 there are three transition diagrams, two for RBNs corresponding to the opposite values $p = 0.1$ and $p = 0.9$, and one for an RBN with a value of $p = 0.5$. The *internal eigenvector centrality* measure or *prestige* of the nodes [29,30] in each diagram was calculated. In the transition diagram of the RBN with $p = 0.1$ a single fixed point attractor is observed, in which all states fall after a few time steps. In the transition diagram of the RBN with $p = 0.9$ a similar behavior is observed with two limit cycle attractors. On the other hand, in the RBN with $p = 0.5$ several attractors with more time steps are observed. It can be noted that the attractors have the greatest prestige in each transition diagram.

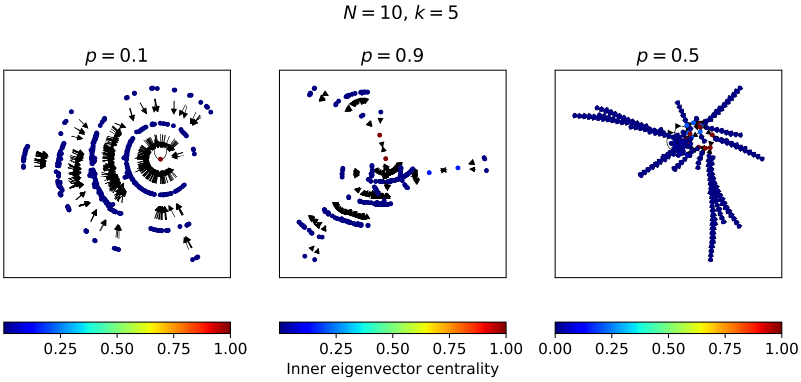


Figure 6. Transition diagrams for three RBNs with 10 nodes and *in-degree* 5. The transition diagrams correspond to the values $p = 0.1$, $p = 0.9$ and $p = 0.5$. The color of the nodes denotes their inner eigenvector centrality.

The transition diagrams obtained are conducive to performing perturbation analysis. In this work, perturbations were carried out on our RBNs according to the following procedure:

1. Perform randomness measurements of the unperturbed transition diagram.
2. Create a list of nodes with the highest prestige.
3. Perturbation: locate the node with the highest prestige in the list, disconnect it from the transition diagram, and perform randomness measurements.
4. Remove the most prestigious node from the list.
5. Restore disconnected node to transition diagram.
6. Repeat steps 3, 4, and 5 until the list of most prestigious nodes is exhausted.

Let's call this perturbation method *perturbation of most relevant nodes*. Similarly, let's call *perturbation of less relevant nodes* the perturbation method where nodes with lower prestige are disconnected (modifying step 2).

Figure 7 shows the results of performing the aforementioned perturbations. The 20 most relevant-prestigious nodes and the 20 least relevant-prestigious nodes in each transition diagram in Figure 6 were perturbed. In each perturbation, the *relative randomness change* (defined here as $[R(G) - R(G \backslash e)]/R(G)$) of the respective transition diagram was calculated from its adjacency matrix. The perturbations series obtained show that the compressibility by LZW is blind to the perturbations, as is the entropy (except for a few perturbations), while the algorithmic complexity - estimated by BDM - is sensitive in the face of these perturbations.

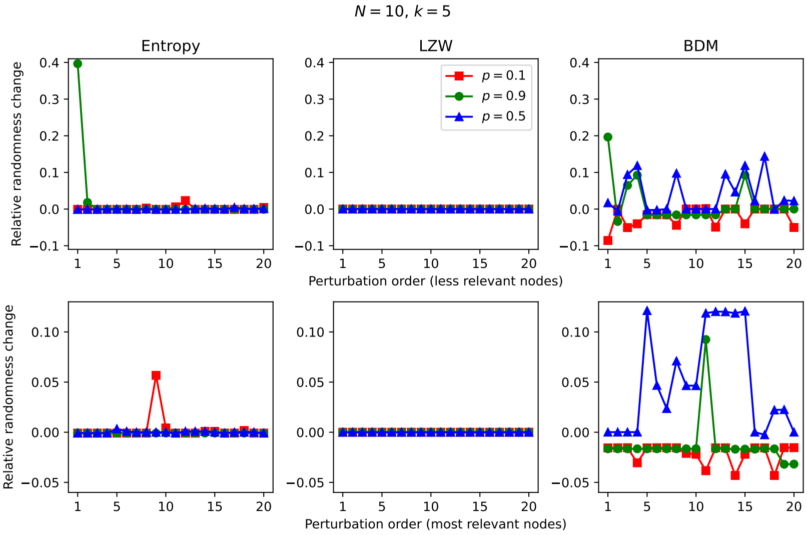


Figure 7. Relative randomness change produced by perturbations of the transition diagrams from Figure 6. The randomness measures were applied to the adjacency matrices of the transition diagrams.

In the graphs at the right of Figure 7, values of relative shared mutual algorithmic information (AID) between the perturbed and unperturbed transition diagrams of Figure 6 are shown. Here, we can observe that for the transition diagram with $p = 0.1$ the first least relevant node makes a relatively high causal contribution to the transition diagram (because AID < 0 for that perturbation). This is to be expected since the least relevant nodes are found on the periphery of the transition diagram to which they belong, which makes them more preceding nodes in the temporal evolution of the respective RBN. Regarding the perturbation of more relevant nodes, we see a few AID inverted peaks under a constant low absolute value; this tells us that the majority of less relevant nodes are not essential in the explanation of the system, so it is likely that they are elements produced by the natural course of evolution of the system. The latter is to be expected since the most relevant-prestigious nodes are usually close to the attractors. A similar analysis follows from the results for the transition diagram with $p = 0.9$, although in this case the first least relevant node does not take part in the algorithmic content of the transition diagram (because AID > 0). Finally, for the transition diagram with $p = 0.5$, there are more nodes with low causal contribution (AID > 0) with respect to the other two transition diagrams; the existence of such nodes with low causal contribution is perhaps due to the fact that the relevant nodes were selected from among the several basins of attraction in the transition diagram, which are separated from each other.

5. Conclusions

We used three measures of randomness to analyze the dynamics of RBNs with synchronous updates. Two main results were shown in this work: a) the randomness estimation by BDM is capable of showing jumps from null values of randomness and "causality gaps", which makes BDM the randomness measure capable of detecting regime changes in a more noticeable way with respect to the other measures of randomness; b) BDM is the only randomness measure that is capable of revealing the causal contribution of certain states according to the order they occupy in the temporal evolution of an RBN. The theoretical results presented in subsection 4.2 introduce the application of reprogrammability analysis in Boolean networks, potentially leading to practical applications. Specifically, we believe that a future work may be the extension of the analysis to RBN models with asynchronous updates [9], or the application of perturbation analysis on connectivity graphs in regulatory network models to help simplify such graphs and their Boolean rules/functions —i.e., to perform gene knock-outs in order to find the causal regulatory network.

Finally, a brief explanation is in order for the accuracy of the Block Decomposition Method. For larger networks, some effects may be lost, but precisely the sensitivity of the method holds for small perturbations which popular lossless compression would fail [23].

Funding: This research received no external funding.

Acknowledgments: (Remains pending).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RBN	Random Boolean Networks
AC	Algorithmic Complexity
CTM	Coding Theorem Method
BDM	Block Decomposition Method

References

1. Kauffman, S. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology* **1969**, *22*, 437–467. doi:https://doi.org/10.1016/0022-5193(69)90015-0.
2. Glass, L.; Kauffman, S.A. The logical analysis of continuous, non-linear biochemical control networks. *Journal of Theoretical Biology* **1973**, *39*, 103–129. doi:https://doi.org/10.1016/0022-5193(73)90208-7.
3. Thomas, R. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology* **1973**, *42*, 563–585. doi:https://doi.org/10.1016/0022-5193(73)90247-6.
4. Fauré, A.; Naldi, A.; Chaouiya, C.; Thieffry, D. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics* **2006**, *22*, e124–e131, [https://academic.oup.com/bioinformatics/article-pdf/22/14/e124/48839020/bioinformatics_22_14_e124.pdf]. doi:10.1093/bioinformatics/btl210.
5. Li, S.; Assmann, S.; Albert, R. Predicting essential components of signal transduction networks: A dynamic model of guard cell abscisic acid signaling. *PLoS biology* **2006**, *4*, 1732–1748. doi:10.1371/journal.pbio.0040312.
6. Thakar, J.; Parette, M.; Kirimanjeswara, G.; Harvill, E.; Albert, R. Modeling Systems-Level Regulation of Host Immune Responses. *PLoS computational biology* **2007**, *3*, e109. doi:10.1371/journal.pcbi.0030109.

7. Veliz-Cuba, A.; Stigler, B. Boolean models can explain bistability in the lac operon. *Journal of computational biology : a journal of computational molecular cell biology* **2011**, *18*, 783—794. doi:10.1089/cmb.2011.0031.
8. Albert, R.; Othmer, H.G. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster. *Journal of Theoretical Biology* **2003**, *223*, 1–18. doi:https://doi.org/10.1016/S0022-5193(03)00035-3.
9. Chaves, M.; Albert, R.; Sontag, E.D. Robustness and fragility of Boolean models for genetic regulatory networks. *Journal of Theoretical Biology* **2005**, *235*, 431–449. doi:https://doi.org/10.1016/j.jtbi.2005.01.023.
10. Shmulevich, I.; Dougherty, E.R. *Probabilistic Boolean Networks*; Society for Industrial and Applied Mathematics, 2010; [https://epubs.siam.org/doi/pdf/10.1137/1.9780898717631]. doi:10.1137/1.9780898717631.
11. Wuensche, A. Classifying cellular automata automatically: Finding gliders, filtering, and relating space-time patterns, attractor basins, and theZ parameter. *Complexity* **1999**, *4*, 47–66. doi:10.1002/(SICI)1099-0526(199901/02)4:33.0.CO;2-V.
12. Shannon, C.E. A Mathematical Theory of Communication. *The Bell System Technical Journal* **1948**, *27*, 379–423.
13. Stone, J.V. *Information Theory: A Tutorial Introduction*, 1st ed.; Sebtel Press, 2015.
14. Ziv, J.; Lempel, A. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory* **1978**, *24*, 530–536. doi:10.1109/TIT.1978.1055934.
15. Snyder, D. bzip2. https://gitlab.com/bzip2/bzip2/, 2023. Accessed: (January 15, 2024).
16. Kolmogorov, A.N. Three approaches to the quantitative definition of information *. *International Journal of Computer Mathematics* **1968**, *2*, 157–168, [https://doi.org/10.1080/00207166808803030]. doi:10.1080/00207166808803030.
17. Chaitin, G.J. On the Length of Programs for Computing Finite Binary Sequences. *J. ACM* **1966**, *13*, 547–569. doi:10.1145/321356.321363.
18. Bennett, C.H. Logical Depth and Physical Complexity. In *The Universal Turing Machine. A Half-Century Survey*; Herken, R., Ed.; Presses Universitaires de France, 1992; pp. 227–257.
19. Copeland, B. *The Essential Turing*; Clarendon Press, 2004.
20. Solomonoff, R. A formal theory of inductive inference. Part I. *Information and Control* **1964**, *7*, 1–22. doi:https://doi.org/10.1016/S0019-9958(64)90223-2.
21. Levin, L.A. Laws of information conservation and problems of foundation of probability theory. *Probl. Peredachi Inf.* **1974**, *10*, 30–35.
22. Calude, C.; Chaitin, G.; Salomaa, A. *Information and Randomness: An Algorithmic Perspective*; Monographs in Theoretical Computer Science. An EATCS Series, Springer Berlin Heidelberg, 2013.
23. Zenil, H.; Hernández-Orozco, S.; Kiani, N.A.; Soler-Toscano, F.; Rueda-Toicen, A.; Tegnér, J. A Decomposition Method for Global Evaluation of Shannon Entropy and Local Estimations of Algorithmic Complexity. *Entropy* **2018**, *20*. doi:10.3390/e20080605.
24. Delahaye, J.P.; Zenil, H. Numerical evaluation of algorithmic complexity for short strings: A glance into the innermost structure of randomness. *Applied Mathematics and Computation* **2012**, *219*, 63–77.
25. Soler-Toscano, F.; Zenil, H.; Delahaye, J.P.; Gauvrit, N. Calculating Kolmogorov complexity from the output frequency distributions of small Turing machines. *PloS one* **2014**, *9*, e96223.
26. Zenil, H.; Kiani, N.A.; Marabita, F.; Deng, Y.; Elias, S.; Schmidt, A.; Ball, G.; Tegnér, J. An Algorithmic Information Calculus for Causal Discovery and Reprogramming Systems. *iScience* **2019**, *19*, 1160–1172. doi:https://doi.org/10.1016/j.isci.2019.07.043.
27. Zenil, H.; Kiani, N.A.; Tegnér, J., Algorithmic Information Dynamics (AID). In *Algorithmic Information Dynamics: A Computational Approach to Causality with Applications to Living Systems*; Cambridge University Press, 2023; p. 228–240.
28. Talaga, S. PyBDM: Python interface to the Block Decomposition Method. https://pybdm-docs.readthedocs.io/en/latest/, 2019. Accessed: (February 04, 2024).
29. Bonacich, P. Power and Centrality: A Family of Measures. *American Journal of Sociology* **1987**, *92*, 1170–1182.
30. Newman, M. *Networks*; OUP Oxford, 2018; p. 159.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.