Boolean Networks: Fundamentals, Applicability, and Simplification for Biological Systems

Will Gribbin

MS Program in Physiology
June 1, 2018
Computational Physiology, Boolean Networks, Methods

Abstract.

Boolean network models are discrete logic models that have grown in popularity over the past two decades for modeling systems in molecular biology. Boolean network models are popular because they allow for predictions to be made on systems where the exact molecular kinetics are unclear or unknown. Fundamentally, a Boolean network is a series of nodes that can have two values, on (true, 1) or off (false,0). Nodes are connected by transfer functions that describe how they interact and update through time. Eventually, Boolean networks reach steady states, or attractors, which in biological models are often representative of cellular states such as apoptosis or normal progression through the cell cycle. Using Boolean algebra, a published Boolean model of pediatric neuroblastoma was simplified while maintaining model integrity and output. The simplification of the model reduced the amount of time to compute by a 75%. Boolean modeling is a useful tool for scientists seeking to find new hypotheses to test in ill-defined systems Boolean modeling of biological systems is not without its drawbacks, and new modifications to Boolean modeling have been proposed to extend its usability as a modeling platform. Perhaps one day we will have a full physiological model of human physiology.

Introduction.

Computational physiology aims to model and simulate physiological systems to gain a deeper understanding of them. Considering the level of organization within the human body, it is easy to become overwhelmed. We are comprised of tens of organs, arising from tissues from over 200 cell types. The total number of cells in a single person are in the trillions, and each of these cells has access to varied subsets of the approximately 20,000 protein coding genes in humans (1). And yet, from this complexity a regular physiological order emerges, that of homeostasis. Teleologically, the emergence of homeostasis as a ubiquitous biological feature allows individual organisms to exclude extraneous detail from their environment, and at a species level focus on long-term evolutionary adaptation to recurrent pressures. This is a complex phenomenon, with many levels of regulation. Using a simulated computer model, Stern showed that the emergence of homeostasis was positively correlated with species fitness (2). Specifically, the model predicted the ability for ‘species’ to exclude extraneous ‘noise’ signals from the environment and adapt to fitness determining signals. It is incredible that the emergence of homeostasis can be predicted through computation by a simplified model. After all, homeostasis is a highly complex phenomena that means different things to different species at different scales. So, the question is not can we create computational models of human physiology, but where would the modeler prefer to start?

Recently, Boolean models have become popular tools for computational physiologists to investigate molecular biological systems of interest (3,4). One key assumption of Boolean models is that effectors of signaling are discretely on (true, 1) or off (false, 0). An effector can be a gene, the protein that gene encodes, or the macroscopic stimulus that initiates signaling in the first place. While it is easy to understand how a gene or stimulus could be considered on or off, how can proteins be on or off? The effects of many proteins can be accurately described with a simple does-response curve (5). Generally, there is a non-linear dependency between dose and response, or that increases in response do not correspond one-to-one with increases in effector. Additionally, an initial sub-functional concentration of product is followed by a

concentration of protein sufficient for a response (6). While not all dose-response curves are sigmoidal, the point is that the sub- and supra- physiological concentrations can be roughly simplified to discrete, on-off behavior (Figure 1). The simplification works well enough to simulate many biological functions.

It is counterintuitive that simplifying complex cellular dynamics to simple on-off states can still yield results that resemble natural physiological pathways, and yet Boolean Models do wonderfully as simple models of many biological processes. For example, Fauré *et al.* were able to replicate key components of the mammalian cell cycle using a 10 part Boolean network model (7). Replicating an inherently time sensitive process using a method of modeling that does not require kinetic parameters is impressive and a great example of how versatile Boolean networks can be as a modeling platform. Albert and Othmer used Boolean modeling to simulate the expression of segment polarity genes in *D. melanogaster* (8). The model generated by Albert and Othmer revealed the inner dynamics of mutant systems, and was able to recreate the expression patterns of wildtype and mutant systems (8). Davidich *et al.* generated a Boolean model of fission yeast that was predicted the viability of synthetic mutants, and then created the predicted mutants showing that the viability of real mutants was identical to the viability predicted by the model (9). The versatility of Boolean modeling is one of the most exciting aspects of the field. Understanding the fundamental rules underlying the system is critical to construct and make inferences on a model.

Nodes

Fundamentally, a Boolean network is a collection of nodes that interact with each other. In a Boolean network, the node has one property: its state. For a Boolean network with N nodes, there are 2^N possible states for all nodes in that network, referred to as the vertex field. A Boolean state can be either on or off, true or false, one or zero (6). To illustrate, there could be a node ‘blue_sky’. If one were to look out the window now and observe that it was an overcast day the state of the ‘blue sky’ would be off: no blue sky today. If one had observed a sunny day, then the state of the ‘blue sky’ node would be on. Obviously, the state of ‘blue sky’ tells us whether the sky is blue, but nothing else. In a sense, this is a valid model, but it is very simple and of little interest as a system of

study. To generate a proper model, there must be cause and effect. The state of a node means little in isolation but informs how the nodes of a network interact. The interaction between nodes is what makes a model.

Edges – Transfer Functions – Rules

An edge is an interaction between nodes and defines the way nodes in a network interact. Edges are described as being transfer functions. A transfer function is sometimes called an update rule or simply a rule. A transfer function informs the way in which the nodes in the network interact. Two fundamental ways that edges can transmit information are referred to as the identity transfer function and inverse transfer function. An edge with an identity transfer function will instruct the node to which it points to assume the same state as the node from which the edge originates. An edge with an inverse transfer function will instruct the node to which it points to assume the opposite state of the node from which the edge originated. Identity edges are sometimes called ‘activating’, and inverse edges called ‘inhibiting'. In diagrammatic form, identity edges are represented with an arrow and inverse edges are represented with a blunted cap.

To construct a simple Boolean network using only identity and inverse edges, nodes are placed on a figure and edges representing the transfer functions governing the interactions between nodes are drawn (Figure 2). Iterating the network in Fig. 2 will update the nodes within every timestep (Figure 3). Notice, that all nodes are updated at each timestep, not just a singular node of interest. Identity and inverse edges are not the only way in which information is transferred between nodes. Boolean logic gates, also called operators or functoare integrators of edges. Using logic gate, it is possible to generate more complex relationships between nodes than is possible using only identity and inverse transfer functions.

Logic Gates – AND/OR

The ‘and’ logic gate is ideally suited for describing behavior where multiple conditions are necessary to observe a specific outcome. If A and B then C describes a situation where both nodes A and B must be on the turn on node C. A physiological example of this behavior is a receptor and its agonist. For the downstream effect of a cell receptor

to occur (node X) both the receptor (node Y) and its ligand (node B) must be present. When the not logical operation is utilized, the behavior of a receptor, its agonist, and an inhibitor whose action is dominant over the agonist can be generated (Figure 4A).

The ‘or’ transfer function works well for situations where multiple inputs are sufficient to achieve a specific outcome. In the case of “if A or B then C,” both A and B are sufficient to observe outcome C. In the cell, this is an approximation of a promiscuous receptor that can be activated by two ligands, or a gene that can be activated by multiple transcription factors. Amending the previous statement to “if A or not B then C” generates an interaction with behavior similar in nature to a constitutively active enzyme under the control of an activator and inhibitor, where the action of the activator is dominant over the action of the inhibitor (Figure 4B).

Graphical Representation of Networks

It is correct to think of gates as being part of the transfer function that connects two nodes, but it is also fine to consider gates to be separate entities from edges. An intuitive way of visualizing Boolean networks is to create a wiring diagram showing the logical relationship between nodes. However, when the logical functions ‘and’ or ‘or’ are utilized in transfer functions, simplistic wiring diagrams can become confusing. Visualizing transfer functions as being comprised of edges which either are identity or inverse, and logic gates like ‘and’ or ‘or’ make sure that no information is lost in the graphical representation of the system being modeled (Fig. 5).

Representing gates as discrete units of the system is beneficial for two reasons. First, direct representation of ‘and’ or ‘or’ logical functions makes reconstructing the list of transfer functions, or rules, that govern the system very simple. All that a modeler would need to do is transfer the information to a list of rules. Second, explicitly stating the logical function makes understanding what is happening in a system simpler for those reading a publication. There is no ambiguity as to what logic gate is being used. One potential downside to representing a Boolean network in the way described might be confusion about how iterating the model works, because the ‘and’ and ‘or’ logic gates resemble nodes. It is critical to understand that gates are not nodes, so when the model

is iterated only the nodes are updated. For example, in Figure 5 the updating rule *C = not B and A* is observed. One might think that the ‘and’ gate would become true, and on the second timestep this information would transfer to node A, acting as a delay. That delay does not occur, because the representation of ‘and’ in the figure is only there to benefit the reader. If B and C were false and A were true at time zero and the model was iterated by one time step, node C would become true.

Toy Model Construction de novo

Expanding the previously mentioned ‘blue_sky’ node to a three-node network illustrates how the ‘and’ gate can be used to create a model that predicts if the sky is blue or not. For the sky to be blue, it cannot be a cloudy day. Also, the sky also cannot be blue at night. If it is either night or cloudy, then, the sky cannot be blue (Figure 6). So the model’s ‘blue_sky’ node will only be true if both the ‘night’ and ‘clouds’ nodes are false. Imagine that it is a nice sunny day, and you are relaxing outside. You watch as a bank of thunderstorm clouds rolls across the sky, and soon it is overcast. The state of the model before the clouds came was ‘night’ = false, ‘clouds’ = false, and ‘blue_sky’ = true. Once the clouds come and cover the sky, the node ‘clouds’ updates to true. Upon the next timestep, the conditions for ‘blue_sky’ node to be true are no longer satisfied because the ‘clouds’ node is now true. This results in the ‘blue_sky’ node to update to false.

One might wonder at what point the number of clouds is sufficient to change the sky to not be blue? Is one cloud? Twenty? It is important to recognize that the level of clouds required to change the state of the ‘clouds’ node is inherently the amount of clouds sufficient to elicit the response. The benefit of using a Boolean model is that only thing necessary to construct one is knowledge of how the different nodes in the network affect each other. No kinetic parameters are required. With that in mind, Boolean models simply are not meant to make predictions about the kinetic parameters of the system.

Boolean Algebra

Reorganizing update rules using Boolean algebra can lead to model simplification. In the ‘blue_sky’ model, is not saying that the sky is blue if there are no clouds in the sky and it is not nighttime the same as saying if it is not cloudy or night then the sky is blue? In fact this is true, the statements *‘blue_sky’ = not ‘clouds’ and not ‘night’* is in fact identical to ‘*blue_sky’ = not (‘clouds’ or ‘night’)* through De Morgan’s theorem which states (10):

not A and not B = not (A or B)

It may seem as though this is splitting hairs but being able to re-arrange the elements of a Boolean function is important for simplification. Imagine the following Boolean network:

A = not A and not C
B = A or C
C = not A

At first, it may seem as though this network is irreducible, but using De Morgan’s theorem and some basic Boolean algebra, node C can be completely removed as follows:

A = ~~not B and not C = not (A or C)~~ = not B
A = not B
and
C = ~~not A = not (not B)~~ = B
C = B
Thus,

A = not B
B = A or B

Were the crossed-out portion of the formula represent their redundancy to the system. When models are large, reducing the number of nodes is important to minimize the amount of time required to simulate it. Using Boolean algebra to simplify models while maintaining their outcomes and dynamics is therefore incredibly important.

Update Schemes - Synchronous

There are different ways in which nodes can update at time steps. When nodes are updated once per iteration at the same time (as was done in the previous example), it is called a synchronous updating (SU) scheme (11). Additionally, when using a SU scheme, it is assumed that the transition of nodes between states takes the exact same amount of time for each node. The iterations of the model are the method in which time is incorporated into the model: there is a unidirectional progression from cause to effect. In a simple two node model represented as $A^{t+1} = B^t$; $B^{t+1} = not\ A^t$, at time zero imagine that both nodes A and B were true. Now iterate the model one step.. $A^{t=1}$ calls for node $B^{t=0}$, so node at $A^{t=1}$ becomes true and node $B^{t=1}$ updates to the opposite of node $A^{t=0}$ which was true, so $B^{t=1}$ changes state to false. To iterate the model another step, all states of the network are advanced another step. Node $B^{t=1}$ is false, so $A^{t=2}$ will update to match, becoming false. Node $B^{t=2}$ will update to be the opposite of node $A^{t=1}$, becoming false. For a given initial conditions a SU scheme will always lead to the same solution in the exact same number of iterations, and is referred to as deterministic (6).

Update Schemes - Asynchronous

In the complex milieu of the intracellular environment, molecular processes are happening constantly. Proteins are transcribed and degraded; DNA is methylated and transcribed; enzymes are activated and deactivated. To assume that all the elements of a cell change their state at the same time is unrealistic: reality is messier than that. To address this, modelers have created asynchronous updating (AU) schemes. A popular version of AU schemes is the random order asynchronous updating (ROAU) scheme. At every timestep, a ROAU scheme will select nodes at random until every updating rule has been selected. The nodes will then update in the randomly selected order. Imagine there exists a network:

A = not B
B = C or A
C = A and not B

The type of scheme used to update the model plays a large role in how the model behaves. Assume that the starting state of the network was that *A = True, B = False, C*

= *True*. If a ROAU scheme were used to update this model, at the first iteration an order of update would be chosen. Suppose a ROAU chose the order C -> B -> A. the model would then update as follows. Node C will be true, because both node A is true and node B is false. Node B is next, and will update to True because both node C and node A are true, either one being true would have been sufficient to reach this end. Finally, node A is updated to false because it assumes the opposite state of node B. At the end of the first iteration, the new node states are *A = true, B = true, C = false.* Now, imagine that a different order had initially been selected for updating. Instead of the ROAU scheme choosing C -> B -> A, perhaps it chose A -> B -> C. Using the same initial conditions, *A = True, B = False, C = True,* A would become true, because B is false. B would become true because C and A are true, again either would have been sufficient. C would become false because B is true. Using a ROAU scheme, with the same starting conditions the network can arrive at a completely different state due to the order in which nodes update. The benefit of using a ROAU scheme is that it eliminates artificial steady states introduced when using a SU scheme (12,13). A ROAU scheme eliminates that artifacting and ensure that only the most robust limit cycles and steady states are reached (14).

Attractors

Ultimately, a Boolean model progresses from timestep to timestep until it reaches a steady state. These steady states are known as attractors and in models of biological systems generally will correspond to homeostatic steady states if a model is well constructed. For example, in a Boolean network of a fission yeast cell cycle the largest attractor corresponded exactly with a biologically stable state (15). An attractor can be a fixed point or an oscillating set of points called a limit cycle. In Boolean networks, attractors are a single system state for fixed point attractors, or a set of interchanging system states for limit cycles.

In 2003, Samuelson and Troein found that in random Boolean networks as the number of nodes grew, the number of attractors grew at a superpolynomial rate (16). A random Boolean network is a collection of nodes whose interactions are randomly generated. Random Boolean networks were some of tools first used when exploring the use of

Boolean models for biological applications. Two years later Klemm and Bornholdt countered the applicability of the work Samuelson and Troein, as they had studied attractor structure in a SU scheme system (17). Klemm and Bornholdt argued that SU scheme systems are susceptible to overestimation of the number steady states, finding that in asynchronous models the number of attractors grew sublinearly (17). Again, what is clear that when modeling using Boolean networks, it is of the utmost importance to understand the underlying mechanics of the system of study and to choose an appropriate update scheme.

Application of Boolean Models to Neuroblastoma

Utilization of Boolean modeling can provide a dynamic picture of gene regulation and is broadly applicable to predicating outcomes of disease states. An exciting and emerging application for Booelan modeling is simulating the outcomes of varying types of diseases and cancers. The goal of the human genome project was to create a map of the complete genetic identity of the human species (18). An obvious motivation for undertaking this work was to better understand how to cure genetic disease (19). From the results of the human genome project came the understanding that disease states are more than the genes underlying the specific disease. Disease is seldom purely genetic, it a complex, dynamic interaction between genes and environment (20). Should not the tools that are used to predict patient outcomes incorporate a dynamic understanding of genetics? And yet the AJCC, the definitive authority on the staging of cancer and prognostic indicators of cancer, focuses very little on dynamic modeling (21). Instead, the presence of genetic mutations are considered to be the key risk factors indicative disease severity and progression. This intense focus on what genetic mutations a person has or does not have is incomplete. Consider the *BRCA1/2* mutations now understood to be correlated breast cancer susceptibility (22). While women with BRCA1/2 mutations are more likely to develop cancer, what about all the other women who develop cancer despite having regular *BRCA1/2*. Or women who have *BRCA1/2* mutations and never develop breast cancer at all (23)? To reach a truly individualized picture of cancer pathogenesis, incorporation of the dynamic nature of gene regulation and the cell as a whole is needed.

Neuroblastoma is the most deadly and abundant solid tumor in pediatric populations (24). Neuroblastoma arises from peripheral nervous system precursor cells that fail to migrate out of the CNS (25). High tyrosine receptor kinase B (TrkB) expression is correlated to neuroblastoma with high MYCN gene expression which itself is tightly correlated to malignant neuroblastoma with poor prognosis (26). TrkB itself is a receptor for brain-derived neurotrophic factor (BDNF). BDNF promotes survival of TrkB expressing tumors, especially those with high MYCN expression (26). Tyrosine receptor kinase A (TrkA) is a receptor for NGF, and overexpression of TrkA is associated with good prognosis in human neuroblastoma patients, potentially because neuroblastoma cells overexpressing TrkA are reliant on NGF for survival (27). Anaplastic lymphoma kinase (ALK) is a receptor for midkine (MDK) and high expression of ALK is associated with proliferative growth of neuroblastoma tumor tissue (28,29). The TrkA/NGF, TrkB/BDNF, and ALK/MDK (TNTBAM) axes are some of the most predictive elements of prognosis in neuroblastoma, but how these elements dynamically interact is unclear.

Kasemeier-Kulesa *et al.* created a Boolean network model of pediatric neuroblastoma that incorporated the TNTBAM axes as input signals and modeled their downstream elements to create a prognostic model that is better able to predict neuroblastoma progression than traditional, gene-centric methods (30). The end goal of the model was to create a tool to which expression levels of components of the TNTBAM axes could be fed, which would then output how likely it was that a patient's neuroblastoma would progress. Differentiation and apoptosis were outputs that were associated with a good prognosis of neuroblastoma, whereas proliferation and angiogenesis were associated with bad prognostic outcomes. The authors provide two models of neuroblastoma, a more complicated 21 node model and a simplified 18 node model. The most striking topographical difference between the two models proposed by Kasemeier-Kulesa *et al.* is the elimination of the MTOR and MAPK rules, as well as the removal of the DNADamage node (Figure 7). The reason that Kasemeier-Kulesa *et al.* removed these nodes is that they are insulated from the remainder of the model. These nodes have only one input and one output. Importantly, the nodes to which MTOR and MAPK output have no outputs themselves, they are terminal nodes in the model. Directly substituting

the rules that regulate MTOR and MAPK in place of the rules that regulate Angiogenesis and Differentiation respectively results in no change in the predicted outcomes of the model, and no change in the likely hood of entering a steady state in the wildtype model of the model (Equation 1-2).

A hallmark of neuroblastoma progression is the loss of p53 function (31). In neuroblastoma, the p53 gene itself is rarely mutated (32). Functionally, though, p53 tumor suppressor activity is essentially non-existent in neuroblastoma cells. This is because p53 is sequestered in the cytoplasm. Because p53 is a transcription factor it must be in the nucleus for it to act as a tumor suppressor, so exclusion from the nucleus renders it effectively inactive through a MDM2 dependent pathway (33). The final difference between the 21-node model and the 18-node model is the removal of the DNADamage node from the system. By removing the DNADamage node from the model 21-node model, Kasemeier-Kulesa *et al.* (30) are trying to replicate a situation where the P53-MDM2-DNADamage axis of the model is knocked out, or always set to false. Indeed, to replicate the Kasemeier-Kulesa *et al.*(30) results, one is required to artificially knock-out (always set to false) the p53 node. The results of this model were validated on data from real patient samples, and the authors state hat the predictive quality of their Boolean model outperforms more traditional methods of prognostic analysis(30).

Hypothesis

In their 2018 paper in Biophysical Chemistry, Kasemeier-Kulesa *et al.* used a Boolean network model of the signaling underlying neuroblastoma to create a model that could predict the prognosis of patients with higher accuracy than traditional methods (30). Studying the method by which the authors reduced the originally proposed 21 node network to a 18 node network provides insight into what is important to minimize computational intensity while yielding the minimum functioning model of the system of interest (Figure 7). Examining the topology of the model proposed by the authors, it became clear that there may be room for improvement in the model. It was hypothesized that the Kasemeier-Kulesa *et al.* (30) model as published contained redundant nodes, and removal of those nodes would result in a model that would be

functionally identical to the original model while significantly reducing the computational requirements.

Methods

All Boolean network modeling was conducted using the Python coding language (v. 2.7.14) utilizing the BooleanNet module created by Albert *et al.* (4). Development of the code used to calculate fixed point attractors and limit cycles was done in the JetBrains PyCharm IDE. (2018.1.2 Community Edition). Graphical representation of Boolean networks were generated using a modification of the BooleSim online simulation tool created by Bock *et al.* (34).

To reduce the Kasemeier-Kulesa *et al.* model of neuroblastoma, methodology was adapted from Veliz-Cuba (35). The model was able to be reduced as follows:

1. Identify nodes that refer to themselves in the transfer function that regulates them. These nodes are irreducible
2. Identify internal nodes that are directly called upon by output nodes and are directly regulated by input nodes. These are potential pass through nodes, they may be able to be reduced
3. Remove any nodes that are never referenced by another node. These are dead ends, and have no impact upon model outcome or dynamics
4. Test the overall outcome of the model in term of both the number and identity of fixed point attractors and the predicted likelihood of entering the outcome nodes given

Results

The Kasemeier-Kulesa *et al.* (30) model is an exciting tool, able to predict the prognosis of patients with neuroblastoma with great accuracy; however, the model can be improved. By removing the MDM2 node from the model outright and substituting the right-hand side of the IP3 rule wherever it had been called, the model was recreated with the same behavior as the 18-node model (Equation 3). The new condensed model has the same model steady states, identical probability of entering a given steady state from a random initial condition, preserved model dynamic structure, conserved fidelity of

the total probability of reaching a given outcome for a given random starting state (Figure 8).

Discussion

The Importance of a Boolean Model of Neuroblastoma

One of the greatest utilities of prognostic prediction is that those predictions can inform the path of treatment. In traditional cancer staging models, stages of cancer are defined often by morphological features and spread. Utilizing a Boolean model of neuroblastoma, Kasemeier-Kulesa *et al.* have created a mechanistically based prognosis tool. The benefit to this approach is that if clinicians can make individualized treatment plans for a patient with a given TNTBAM axis status, say high likelihood of proliferation and a very low likelihood of angiogenesis. The clinician can target the tumor using anti-proliferative agents and skip the anti-angiogenics. Someday, we will treat cancers based upon the dynamics of the cancer of the individual instead of the general toxic approach of chemotherapeutics. This precision approach to individual cancer detection, incorporating the dynamic elements of the genome, truly is the beginning of a new era in cancer treatment. Like the one envisioned by Hannahan and Weinberg in their seminal ‘The hallmarks of cancer’, developing tools that allow us to predict what the dynamical state a cancer which will allow future clinicians to make more individualized treatment intervention decisions (36,37).

Node Reduction in Kasemeier-Kulesa Model - Efficiency

Reducing the Kasemeier-Kulesa model of neuroblastoma importantly reduced the time it takes to compute the model (30). In the fast-paced healthcare arena, the faster the turnaround time of a test, the more rapidly treatments can be given to patients. Here, an equivalent model of neuroblastoma was generated from a more complex published model. While the amount of time it takes for simulation varies widely across computers based upon their processing power, reducing the neuroblastoma model to 18 to 16 nodes in the authors experience reduced compute time from twenty minutes to five minutes. A large drawback to using Boolean models is that the vertex field, and thus compute time, double with each added node (Figure 9). The methodology used here to

reduce the neuroblastoma model is widely applicable to other Boolean model systems. Reduction of the vertex field will allow future analysis of this model using an ROAU scheme because a ROAU scheme requires averaging multiple simulations. When averaging simulations, many repetitions are needed, so any reduction in compute time is of absolute importance. Examining the system with an ROAU scheme will help elucidate which of the steady-states reported here using a SU scheme are artifacts of the timestep.

Limitations of Update Schemes

A ROAU scheme is an improvement over standard SU schemes, but that does not mean that a ROAU scheme is without flaws. Using a ROAU scheme, the model still assumes that nodes take an equivalent amount of time to turn on or off. When there exist large discrepancies between the rate of node state change the SU scheme and ROAU scheme are inappropriate. While some time scale differences are protected against using a ROAU scheme, other updated schemes should be considered if there exists a many order of magnitude difference between the timescales at which nodes operate.

Ideally, one day there will be validated Boolean models for all diseases, but it is important to recognize that for many patient's, disease states do not exist in a vacuum. Understanding the dynamic interaction between a host of diseases is an important step for Boolean models to become useful clinical tools. After all, Boolean Multiplex Networks (BMN) are one methodology created to help model processes that run in parallel. In BMN, multiple regular Boolean networks are placed into layers and communicate with one another (38). Though little work has been done with BMN in the medical realm thus far, a potential application to medicine is that they would provide a framework for creating models that are able to account for, and make predictions upon, the interaction between diseases. One day perhaps a clinician will be able to select a list of models for their patients with comorbidities, say a patient with lung cancer, COPD, and diabetes, and sync them together into a wholistic model of the patients disease state. A 'plug-and-play' incorporation of individualized disease models would be an revolutionary prospect.

Care when Determining Cause and Effect

The only knowledge necessary to begin to construct a Boolean network model of a physiological system of interest, is a knowledge of the topology of interaction between pieces of the system (4). However, this simplicity can have a major down side. For example, the *lac* operon is responsible for activating genes whose products make lactose catabolism possible in *E. coli* (39). One way that the *lac* promoter, within the *lac* operon and its inducers cAMP and isopropyl-β-D-thiogalactoside (IPTG), could be modeled would be by using a simple "and" gate in the form of "if IPTG and cAMP then *lac"* (40–42). But, Setty *et al.* (42) argued that the control of the *lac* promoter by its inducers IPTG and cAMP is not a pure "and" gate. The authors demonstrate that the response of the operon to IPTG and cAMP was graded. At high levels of either IPTG or cAMP, elevated operon activity was noted; However, to reach full operon activity both high levels of IPTG and cAMP were required. What does this mean for the rule "if IPTG and cAMP then *lac*"? A model that used this rule would be unable to predict the intermediate levels of expression, only "on" when both IPTG and cAMP are present. Setty *et al.* demonstrate that the creator of the Boolean network model must take care when assigning the rules that determine interactions (42). Boolean models of systems must be validated upon real experimental data to tease out issues with rules before they can be implemented in fields like healthcare.

The Limits of Boolean Modeling

Boolean modeling has shown to be a promising tool, but as with all modeling approaches, the scope of its applicability is limited. The benefit of using Boolean models is that they are parameter free, so very little about the specific parameters of the system to be modeled need to be known to get a biologically accurate representation. However, Boolean models make the very bold step of assuming that elements in a biological system can only have two states. This simplification, while great when little is known about a system, will never be able to predict the finer elements of a system with any accuracy. Modeling any continuous variable using Boolean modeling is therefore, not possible. What Boolean models excel at is course-grained predictions where the on-off simplification is justified. They will never work for tasks like predicting the chemical

kinetics of biochemical reactions. Just like one would not use a hammer to affix a screw, the appropriate modeling tool for the question at hand must be used to get meaningful results.

Thoughts Concerning the Future of Boolean Modeling in Healthcare

As computers get more and more powerful, the size of the Boolean networks they can handle will only grow. Perhaps one day automated modeling will be part and parcel with lab results, allowing for accurate, validated, model-based prognoses. We are at an unprecedented time in human history. Since the first production computer was built in 1951, the computational power in commercial computers has increased 152 million-fold. The increase in computational power we have witnessed over the past half century shows no signs of slowing down, and with that power new modeling prospects will become feasible. Perhaps one day, we will have a complete model of human physiology, allowing us to conduct experiments on simulated human models that would be considered too dangerous to risk on a real patient set. Such complex models might allow the scientists of the future to craft their hypotheses to a much more specific question before conducting *in vivo* experimentation. While such an idea may seem fanciful today, it is doubtful that the ability of computers to undertake the complex modeling that they do today seemed any less fanciful in 1951.

References

1. **Ezkurdia I, Juan D, Rodriguez JM, Frankish A, Diekhans M, Harrow J, Vazquez J, Valencia A, Tress ML**. 2014 Multiple evidence strands suggest that there may be as few as 19 000 human protein-coding genes. *Hum. Mol. Genet.* 2014;23(22):5866–5878. PMID: 24939910.
2. **Stern MD**. 1999 Emergence of homeostasis and "'noise imprinting'" in an evolution model. 1999;96:10746–10751.
3. **Bornholdt S**. 2008 Boolean network models of cellular regulation: prospects and limitations. *J. R. Soc. Interface* 2008;5 Suppl 1(Suppl 1):S85-94. PMID: 18508746.
4. **Albert I, Thakar J, Li S, Zhang R, Albert R**. 2008 Boolean network simulations for life scientists. *Source Code Biol. Med.* 2008;3(1):16.
5. **DeLean A, Munson PJ, Rodbard D**. 1978 Simultaneous analysis of families of sigmoidal curves: application to bioassay, radioligand assay, and physiological dose-response curves. *Am. J. Physiol.* 1978;235(2):E97-102. PMID: 686171.
6. **Kauffman SA**. 1993 *The Origins of Order: Self-Organization and Selection in Evolution*. New York: Oxford University Press Available at: https://goo.gl/u648kW. Accessed May 21, 2018.
7. **Faure A, Naldi A, Chaouiya C, Thieffry D**. 2006 Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics* 2006;22(14):e124–e131.
8. **Albert R, Othmer HG**. 2003 The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster. *J. Theor. Biol.* 2003;223(1):1–18.
9. **Davidich MI, Bornholdt S**. 2013 Boolean network model predicts knockout mutant phenotypes of fission yeast. *PLoS One* 2013;8(9):e71786. PMID: 24069138.
10. **De Morgan A**. 1847 *Formal logic, or, The calculus of inference, necessary and probable*. London: Taylor and Walton doi:OL7054286M.
11. **Garg A, Di Cara A, Xenarios I, Mendoza L, De Micheli G**. 2008 Synchronous versus asynchronous modeling of gene regulatory networks. *Bioinformatics* 2008;24(17):1917–1925.
12. **Aracena J, Goles E, Moreira A, Salinas L**. 2009 On the robustness of update schedules in Boolean networks. *Biosystems* 2009;97(1):1–8.
13. **Huberman BA, Glance NS**. 1993 Evolutionary games and computer simulations. *Evolution (N. Y).* 1993;90:7716–7718.
14. **Saadatpour A, Albert R**. 2013 Boolean modeling of biological regulatory networks: A methodology tutorial. *Methods* 2013;62(1):3–12.
15. **Davidich MI, Bornholdt S**. 2008 Boolean Network Model Predicts Cell Cycle Sequence of Fission Yeast. Stolovitzky G, ed. *PLoS One* 2008;3(2):e1672.
16. **Samuelsson B, Troein C**. 2003 Superpolynomial Growth in the Number of Attractors in Kauffman Networks. *Phys. Rev. Lett.* 2003;90(9):098701.
17. **Klemm K, Bornholdt S**. Stable and unstable attractors in Boolean networks. doi:10.1103/PhysRevE.72.055101.
18. **Venter JC, Adams MD, Myers EW, Li PW, Mural RJ, Sutton GG, Smith HO, Yandell M, Evans CA, Holt RA**. 2001 The sequence of the human genome.

Science 2001;291(5507):1304–51. PMID: 11181995.
19. **Collins FS, McKusick VA**. 2001 Implications of the Human Genome Project for Medical Science. *JAMA* 2001;285(5):540.
20. **Cohen IR, Atlan H, Efroni S**. 2016 Genetics as Explanation: Limits to the Human Genome Project. In: *eLS*. Chichester, UK: John Wiley & Sons, Ltd; 2016:1–8.
21. **Amin MB, Greene FL, Edge SB, Compton CC, Gershenwald JE, Brookland RK, Meyer L, Gress DM, Byrd DR, Winchester DP**. 2017 The Eighth Edition AJCC Cancer Staging Manual: Continuing to build a bridge from a population-based to a more "personalized" approach to cancer staging. *CA. Cancer J. Clin.* 2017;67(2):93–99.
22. **Robson M, Gilewski T, Haas B, Levin D, Borgen P, Rajan P, Hirschaut Y, Pressman P, Rosen PP, Lesser ML, Norton L, Offit K**. 1998 BRCA-associated breast cancer in young women. *J. Clin. Oncol.* 1998;16(5):1642–9. PMID: 9586873.
23. **Kuchenbaecker KB, Hopper JL, Barnes DR, Phillips K-A, Mooij TM, Roos-Blom M-J, Jervis S, van Leeuwen FE**. 2017 Risks of Breast, Ovarian, and Contralateral Breast Cancer for *BRCA1* and *BRCA2* Mutation Carriers. *JAMA* 2017;317(23):2402. PMID: 28632866.
24. **Noone A, Howlader N, Krapcho M, Miller D, Brest A, Yu M, Ruhl J, Tatalovich Z, Mariotto A, Lewis D, Chen H, Feuer E, Cronin K**. 2018 *SEER Cancer Statistics Review, 1975-2015*. Bethesda: National Cancer Institute
25. **Brodeur GM**. 2003 Neuroblastoma: biological insights into a clinical enigma. *Nat. Rev. Cancer* 2003;3(3):203–216.
26. **Edsjö A, Lavenius E, Nilsson H, Hoehner JC, Simonsson P, Culp LA, Martinsson T, Larsson C, Påhlman S**. Expression of trkB in Human Neuroblastoma in Relation to MYCN Expression and Retinoic Acid Treatment. doi:10.1097/01.LAB.0000074895.48776.D8.
27. **Nakagawara A, Arima-Nakagawara M, Scavarda NJ, Azar CG, Cantor AB, Brodeur GM**. 1993 Association between High Levels of Expression of the TRK Gene and Favorable Outcome in Human Neuroblastoma. *N. Engl. J. Med.* 1993;328(12):847–854.
28. **Schulte JH, Bachmann HS, Brockmeyer B, Depreter K, Oberthür A, Ackermann S, Kahlert Y, Pajtler K, Theissen J, Westermann F, Vandesompele J, Speleman F, Berthold F, Eggert A, Brors B, Hero B, Schramm A, Fischer M**. 2011 High ALK receptor tyrosine kinase expression supersedes ALK mutation as a determining factor of an unfavorable phenotype in primary neuroblastoma. *Clin. Cancer Res.* 2011;17(15):5082–92. PMID: 21632861.
29. **Kishida S, Kadomatsu K**. 2014 Involvement of midkine in neuroblastoma tumourigenesis. *Br. J. Pharmacol.* 2014;171(4):896–904.
30. **Kasemeier-Kulesa JC, Schnell S, Woolley T, Spengler JA, Morrison JA, McKinney MC, Pushel I, Wolfe LA, Kulesa PM**. 2018 Predicting neuroblastoma using developmental signals and a logic-based model. *Biophys. Chem.* 2018;238:30–38.
31. **Moll UM, LaQuaglia M, Bénard J, Riou G**. 1995 Wild-type p53 protein undergoes cytoplasmic sequestration in undifferentiated neuroblastomas but not

in differentiated tumors. *Proc. Natl. Acad. Sci. U. S. A.* 1995;92(10):4407–11. PMID: 7753819.

32. **Vogan K, Bernstein M, Leclerc JM, Brisson L, Brossard J, Brodeur GM, Pelletier J, Gros P**. 1993 Absence of p53 gene mutations in primary neuroblastomas. *Cancer Res.* 1993;53(21):5269–73. PMID: 8221661.
33. **Rodriguez-Lopez AM, Xenaki D, Eden TO, Hickman JA, Chresta CM**. 2001 MDM2 mediated nuclear exclusion of p53 attenuates etoposide-induced apoptosis in neuroblastoma cells. *Mol. Pharmacol.* 2001;59(1):135–43. PMID: 11125034.
34. **Bock M, Scharp T, Talnikar C, Klipp E**. 2014 BooleSim: an interactive Boolean network simulator. *Bioinformatics* 2014;30(1):131–132.
35. **Veliz-Cuba A**. 2009 REDUCTION OF BOOLEAN NETWORKS. 2009. Available at: https://arxiv.org/pdf/0907.0285.pdf. Accessed May 27, 2018.
36. **Hanahan D, Weinberg RA**. 2000 The hallmarks of cancer. *Cell* 2000;100(1):57–70. PMID: 10647931.
37. **Hanahan D, Weinberg RA**. 2011 Hallmarks of Cancer: The Next Generation. *Cell* 2011;144(5):646–674.
38. **Wu Y, Xu J, Sun X-M, Wang W**. 2017 Observability of Boolean multiplex control networks. *Sci. Rep.* 2017;7:46495.
39. **Jacob F, Monod J**. 1961 On the Regulation of Gene Activity. *Cold Spring Harb. Symp. Quant. Biol.* 1961;26(0):193–211.
40. **Griffiths A, Miller J, Suzuki D**. 2000 Catabolite repression of the lac operon: positive control. In: *An Introduction to Genetic Analysis*. 7th ed. New York: W.H. Freeman; 2000. Available at: https://www.ncbi.nlm.nih.gov/books/NBK22065/.
41. **Marbach A, Bettenbrock K**. 2012 lac operon induction in Escherichia coli: Systematic comparison of IPTG and TMG induction and influence of the transacetylase LacA. *J. Biotechnol.* 2012;157(1):82–88. PMID: 22079752.
42. **Setty Y, Mayo AE, Surette MG, Alon U**. 2003 Detailed map of a cis-regulatory input function. *Proc. Natl. Acad. Sci. U. S. A.* 2003;100(13):7702–7. PMID: 12805558.