The Pennsylvania State University
The Graduate School
Eberly College of Science

NETWORK CONTROL AND DAMAGE MITIGATION IN COMPLEX NETWORKED SYSTEM

A Dissertation in
Physics
by
Gang Yang

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2018

The dissertation of Gang Yang was reviewed and approved* by the following:

Réka Albert
Professor of Physics
Dissertation Advisor, Chair of Committee

Dezhe Jin
Professor of Physics

Timothy Reluga
Professor of Mathematics

Lingzhou Xue
Professor of Statistics

Richard W. Robinett
Professor of Physics
Graduate Program Chair

*Signatures are on file in the Graduate School.

Abstract

Dynamical models have been successfully employed to study how different molecular components give rise to to cellular functions in biological systems. Such models are of great importance in understanding the underlying mechanisms of complex disease and designing preventive or therapeutic strategies. Complex diseases often start with abnormal mutations of the system, which can be modeled as network damage in network dynamical model. Network control problems design strategy to influence or drive the system to a desired state. Thus network control and damage mitigation strategies are a promising avenue toward developing disease intervention and therapies.

Developing network dynamical models and network control strategies are challenging tasks as numerous components interact in a diverse ways in a biological system and we often have incomplete information including the dynamical mechanism and precise quantitative parameters from experimental data. Logic dynamical models, such as Boolean network models, demonstrate their value in such situations, including having considerable dynamic richness and capacity to capture emergent characteristic of real biological systems. I contributed to developing two framework to solve network control problem in Boolean network models. I design compensatory interactions to mitigate multiple network deregulations and stabilize the system as disease prevention or immediate treatment method. I also applied a heuristic algorithm to solve the target control problem in Boolean network models, which can be used to design disease treatment. This heuristic algorithm is based on a concept called domain of influence of node states, which describes the stabilization effect of a long-term intervention. These two frameworks complement each other in their method and purpose. Another way to proceed with incomplete information is structure-based control of continuous models. I test and compare two established methods, structural controllability and feedback-vertex control to understand their differences and elucidate relationship between network topology and network dynamics.

I applied the above framework to several real biological models, including dynamical models involved in complex disease such as cancer and T-LGL leukemia. These analytical and computational tools not only generate solutions consistent with estab-

lished experimental results and previous established tools, but also make predictions to help guide experimentalist to design real solutions to the challenging complex diseases. These developed frameworks in my dissertation also points out new directions for future research in network control.

The dissertation is organized in the following way. In chapter 1, I introduce the background, concepts and methods involved in network modeling and network control problems. In chapter 2, I report the work to design immediate damage mitigation strategies through compensatory interactions in Boolean network models. In chapter 3, I report the work to solve the target control problem in Boolean network models through the concept of domain of influence. In chapter 4, I applied the structure-based network control strategies for continuous models to real biological systems. In chapter 5, I discuss possible future works and some preliminary results, especially those inspired by combining ideas from multiple previous chapters.

Table of Contents

Chapter 3

List of Figures

List of Tables

Acknowledgments

This work was supported by NSF grants PHY-1205840, IIS-1161007 and PHY 1545832. The findings and conclusions do not necessarily reflect the view of the funding agency.

This research or portions of this research were conducted with Advanced CyberInfrastructure computational resources provided by The Institute for CyberScience at The Pennsylvania State University (http://ics.psu.edu).

I would like to express my sincere gratitude to my advisor Prof. Réka Albert for the continuous support of my Ph.D study and research, for her patience, motivation and immense knowledge. She has been a role model who I look up to and keep learning from.

I also would like to express my appreciation to my collaborators and co-authors, Colin Campbell and Jorge G. T. Zañudo, for their help and generosity in sharing their knowledge and work. My sincere thanks also go to the rest of group members, Zhongyao Sun, Xiao Gan, Parul Maheshwari and Jordan Rozum, for their useful discussions.

I would like to express my thanks to the rest of my thesis committee: Prof. Dezhe Jin, Prof. Timothy Reluga and Prof. Lingzhou Xue, for their insightful comments and encouragement.

Dedication

I dedicate my dissertation work to my parents, MingChen Yang and Chen Liang, and my grandparents, QiZhong Liang, GuoQin Wang, GuangMing Yang and GuiFen Ma. My family have always been trying to provide the best education resources for me. They have been respective and supportive to each important life decision made by me. I could not express more appreciation for their support, patience and being along my road to explore my interests.

Chapter 1 | Background and concepts

This Chapter is primarily based on a submitted work under review at Springer. The submitted work is a book chapter titled "Modelling of Molecular Networks", where I am the first author, under a book to be published titled "The Dynamics of Biological Systems". This chapter was reproduced with permission from Gang Yang and Réka Albert.

1.1 Introduction

Decades of research in molecular biology established a large amount of information about the structure and function of individual molecules in cells. It is now known that various non-identical (macro)molecules such as DNA, RNA, proteins, small molecules interact in diverse ways. [7,8] The totality of interactions among various molecular components give rise to cellular functions such as movement or proliferation. Thus, cells are an example of complex interacting systems, as are organs, individuals, or populations. In order to understand such systems, researchers are increasingly using networks to represent the components of the system and their interactions. [4, 8–11]

A network (or graph) is a mathematical abstraction, consisting of nodes, which represent different elements, and edges, which specify the pairwise relationships between the elements. [10, 11] In molecular biological networks, nodes are genes, RNA, proteins and small molecules; edges indicate interactions and regulatory relationships. [4,9] Edges can be symmetrical (representing a mutual relationship) or directed (representing mass or information flow from a source to a target). The latter type of edges can also have a sign, representing positive (activating) or negative (inhibitory) influences. The network representation allows the use of graph measures to characterize the organization of the molecular interaction networks. In Sec. 1.2 of this chapter, we will introduce different types of molecular (biological) networks and present informative graph measures.

Complex systems demonstrate several emergent dynamical properties, such as homeostasis, multi-stability or synchronization. [4, 8, 9] To understand and explain these emergent behaviors of the system, the network needs to be complemented by a dynamical model. In Sec. 3, we briefly compare different approaches of dynamical modeling and mainly discuss the procedure to build a discrete dynamical model of molecular networks and how to use the model to make predictions. In Sec. 1.4, we explore established methods to connect the topological properties of the interaction network with the emergent dynamics of the complex system in logic dynamical models. In Sec. 1.5, we introduce the network control problem and briefly discuss its motivations, applications and history of key results. We also outline the structure of the remaining chapters, consisting of three network control projects with each having a different setting.

1.2 The structure of molecular networks

1.2.1 Introduction to molecular networks : classifications and examples

Let us review the kinds of interactions possible inside a cell. Genes are transcribed into mRNAs, which are translated into proteins. Proteins called transcription factors can activate or inhibit the transcription (also called expression) of genes. Proteins interact with each other and may form protein complexes. Proteins called enzymes catalyze chemical reactions of the metabolism. Chemicals from the environment are metabolized or are sensed by receptor proteins. [4, 8, 9] Biologists usually try to group these interactions and separately define four types of networks, namely gene regulation, protein-protein interaction, signal transduction and metabolic networks, but they are in fact interconnected. [4, 8, 9] In the following we exemplify three types of intra-cellular networks, in the order of increasing diversity.

Protein-protein interaction networks are formed by biochemical events and/or electrostatic interactions between proteins. Several methods now exist to detect such interactions on a large scale, such as such as two-hybrid screening [12], biomolecular fluorescence complementation (BiFC) [13] and co-immunoprecipitation (Co-IP) [14]. Such networks have been built for several organisms including S. cerevisiae, Drosophila, C. elegans and human beings. [8, 15–17] For example, 1870 proteins and 2240 identified direct physical interactions between them are mapped in the S. cerevisiae protein-protein interaction network. The network is built through studying combined, non-overlapping data, obtained by systematic two-hybrid analyses. [15, 18]

A gene regulatory network is a set of genes and gene products (mRNA and proteins), that interact with each other and with other substances in the cell to regulate gene expression levels. For example, genes and their interactions involved in embryonic pattern formation in the fruit fly Drosophila melanogaster are mapped into the Drosophila segment polarity network, as shown in Fig. 1.1. [1] Various dynamical models have been built to understand the embryonic development process. [1, 19]

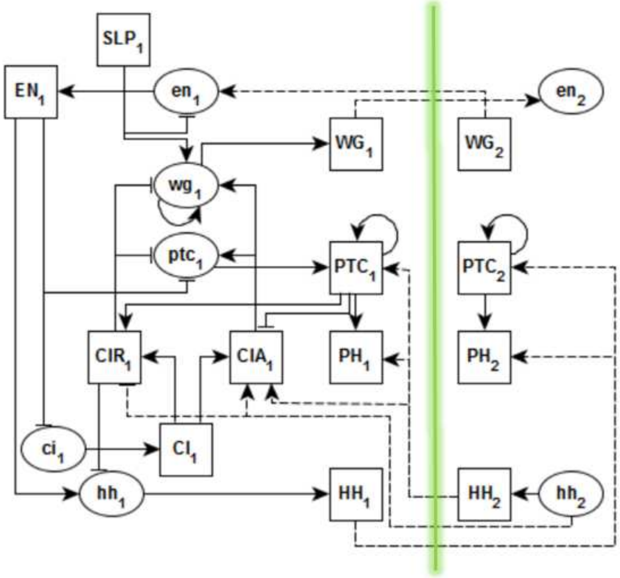


Figure 1.1: Drosophila segment polarity gene network model. 4 cells with periodic boundary conditions are considered and mainly the first cell is shown. The green line indicates a cell boundary. Ellipses represent mRNAs and squares represent proteins. Positive edges terminate in arrow-heads and negative edges terminate in blunt segments. Solid lines indicate intra-cellular regulation and dashed lines indicate inter-cellular regulation. Figure is adapted from [1].

Signal transduction is the process through which living cells receive and respond to various external stimuli. A diverse set of interacting (macro)molecules participate in this process, such as enzymes, other types of proteins, and small molecules. Signal transduction is crucial in the maintenance of cellular homeostasis, in a cell′s communications with its surroundings and in cell behavior such as growth, survival, apoptosis and movement. [20] Many complex diseases, such as developmental disorders, diabetes and cancer, arise from mutations or alterations in the expression of signal transduction pathway components. [5, 21] Fig. 1.2 depicts an example of a real signal transduction network, describing the activation induced cell death of white blood cells called cytotoxic T cells. [2, 3] This network was used to study the disruption of activation induced cell death in the disease T-LGL leukemia, causing the survival of a fraction of activated T cells, which later start attacking healthy cells. The network has 60 nodes and 142 edges. In Fig. 1.2, cellular location is indicated by the shape of the node: rectangles indicate intracellular components, ellipses indicate extracellular components, and diamonds indicate receptors. In addition, hexagonal nodes are conceptual nodes used to summarize connections with other signal transduction mechanisms or cell behaviors. [2, 3]

1.2.2 Network Topological Properties

The totality of the nodes and edges of a network is referred to as the *network structure* or *network topology*. The structural (topological) analysis enables us to trace the propagation of information in the network and determine the key mediators etc. This initial analysis invokes graph theoretical measures, such as centrality measures, shortest paths and network motifs, to describe the organization of the network. [10, 11, 22]

Centrality measures were introduced to describe the importance of individual nodes in the network. The simplest centrality measure is the node *degree*, which is the number of edges connected to the node. For directed networks, the *in-* and *out-degree* of a node is defined as the number of edges coming into or going out of the node, respectively. [10, 11, 22] For example, in the T-LGL leukemia network shown in Fig. 1.2, node CREB (bottom left corner) has in-degree 2 and out-degree 2. In some molecular networks, especially signal transduction networks, it is possible that nodes have an auto-regulatory *loop*, an edge that both starts and ends at the same node. This loop usually represents a stabilizing, or on the contrary, destabilizing, self-influence. For example, the conceptual node Apoptosis has a self-loop, indicating that after commitment to apoptosis (programmed cell death) the process is self-sustaining.

In directed networks, nodes with in- or out-degree of zero are given special names. The nodes with only outgoing edges (with the potential exception of loops) are called *sources*, and nodes with only incoming edges (again, with the potential exception of loops) are *sinks* of the network. In signal transduction networks, source nodes generally

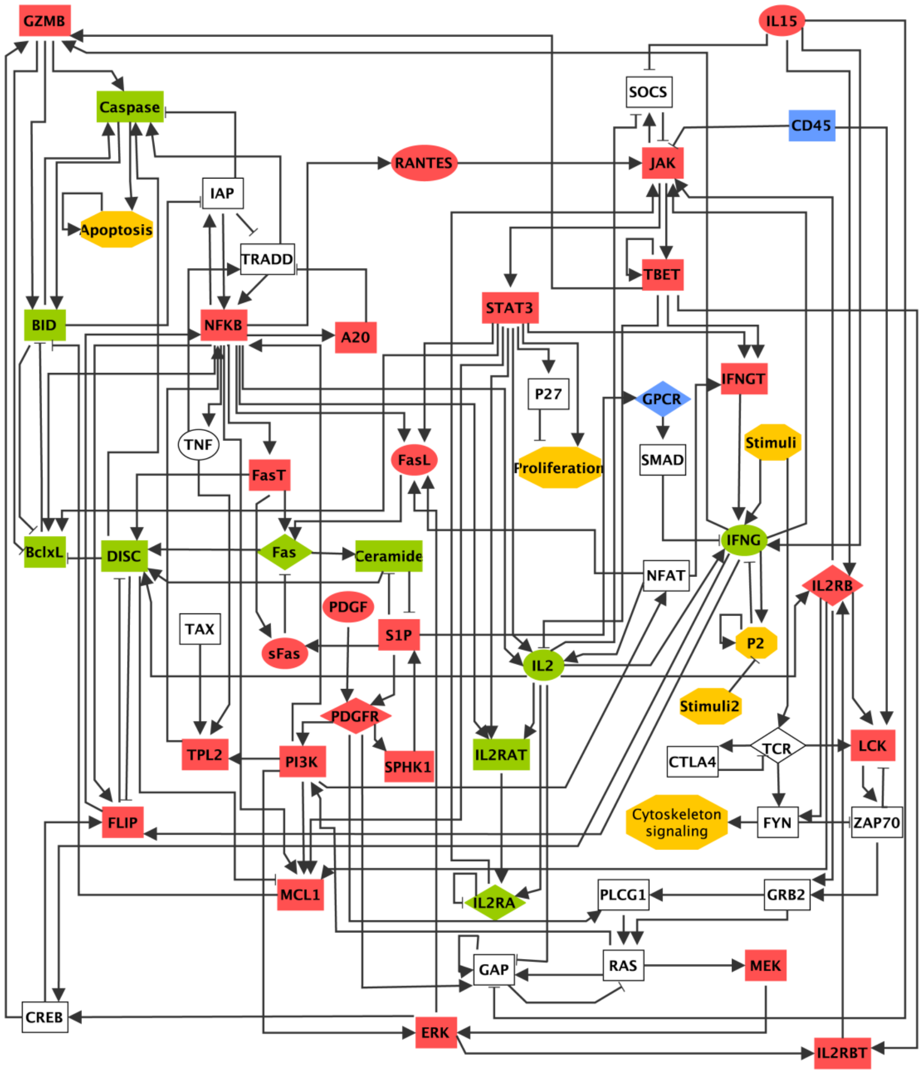


Figure 1.2: A signal transduction network involved in activation induced cell death of white blood cells called cytotoxic T cells. The key signals are Stimuli (representing stimulus of the cell by the presence of pathogens) together with the external molecules interleukin 15 (IL15) and platelet derived growth factor (PDGF). These signals correspond to source nodes, which only have outgoing edges. The key output node of the network is Apoptosis, expressing programmed cell death. Nodes that, like Apoptosis, have no outgoing edges are called sink nodes. The shape of the nodes indicates their cellular location: rectangles indicate intracellular components, ellipses indicate extracellular components, and diamonds indicate receptors. Conceptual nodes are represented by yellow hexagons. The color coding of the nodes indicates the known status of these nodes in abnormally surviving T-LGL cells as compared to normal T cells: red indicates abnormally high expression or activity, green means abnormally low expression or activity, and blue indicates inconclusive or contradictory evidence. An arrow-head or a short perpendicular bar at the end of an edge indicates activation or inhibition, respectively. Details about the name of the nodes can be found in [2, 3]. Figure is reproduced from [3]
.

correspond to external signals, while sink nodes denote responses or outcomes of the process. [4] For example, in Fig. 1.2, the nodes Stimuli, IL15 and PDGF are source nodes and have no incoming edges, and indeed they represent external signals acting on T cells. Proliferation, Cytoskeleton signaling and Apoptosis are sink nodes and have no outgoing edges except the loop of Apoptosis, and indeed they represent outcomes of the signal transduction process: the increase in the number of cells due to cell growth and division, the reorganization of the cytoskeleton necessary for movement, and the genetically determined process of cell-destruction. [2, 3]

Statistical quantities, such as the degree distribution, can be formed to summarize the information of all nodes in the network. [10, 11, 22] The node degree distribution $p(k)$ is a function that, for each degree k, gives the fraction of nodes that have k edges. Similarly, we can define an in-degree and out-degree distribution for directed networks. The degree distribution reveals a lot of information about the structure of the network. For example, in a random network, where the probability of having an edge between each pair of nodes is the same, the node degree distribution will be close to a binomial distribution. [10, 11, 22] However, a variety of molecular networks have a degree distribution that follows a power law, for example, the metabolites in the E. coli metabolic network have an in-degree distribution $P(k) \sim k^{\gamma_{in}}$, where $\gamma_{in} = 2.2$. [23] The heterogeneity encompassed in this so-called scale-free degree distribution has a significant impact on the network′s dynamical properties, such as its controllability and stability with respect to perturbation. [24–26]

The nodes whose degree is in the top 1-5% of the nodes are termed *hubs*. [10, 11] These hub nodes often play an important role in the network. For example, the node representing the NFκB protein has an out-degree of 11 and an in-degree of 4, and is a hub of the T-LGL network on Fig. 1.2. This is expected since NfκB is a transcription factor that is known to be important in cellular responses to various stimuli and in cell survival. [27]

A *path* exists between two nodes if there is a sequence of adjacent edges connecting them. In directed networks, the adjacency needs to be directional as well. [10, 11] Thus in a directed network the existence of a path from A to B does not guarantee that a path from B to A exists. For example, as shown in Fig. 1.2, there is a path from Caspase to the conceptual node Apoptosis, however there is no path from Apoptosis to Caspase.

In networks that can have both positive and negative edges, the *sign of a path* is positive if there are no or an even number of negative edges in the path and is negative if there is an odd number of negative edges. [4, 9] For example, as shown in Fig. 1.2, the path from Stimuli2 to P2 is negative and the path from Stimuli2 to IFNG is positive since the path consists of two negative edges.

A path containing two or more edges that begins and ends at the same node is called a *circuit* or *cycle* (if it does not repeat nodes or edges). The *length* of a path or a cycle is defined to be the number of its edges (loops can be considered as cycles of length one). A directed cycle is also called feedback loops. The sign of a cycle is defined the same way as the sign of a path. For example, as shown in Fig. 1.2, the cycle between S1P, PDGFR and SPHK1 is a positive feedback loop, while the cycle between TCR and CTLA4 is a negative feedback loop.

An undirected network is connected if there is a path between any two nodes. A disconnected network is made up by two or more connected components (sub-graphs). A directed network is *strongly connected* if for any two nodes u and v in the network, there is a directed path both from u to v and from v to u. If a network is not strongly connected, it is informative to identify *strongly connected components* of the network. Having no strongly connected components (SCCs) indicates that the network has an acyclic structure (*i.e.*, it does not contain feedback loops), while having a large SCC implies that the network has a central core. The core can be obtained by iteratively removing source and sink nodes until no nodes can be removed from the network. A directed network is weakly connected if it is connected when we disregard the edge directions. Signaling networks tend to have a strongly connected core of considerable size. [28] For example, the network on Fig. 1.2 has a strongly connected component of 44 nodes, which represents 75% of all nodes.

We can define the *in-component* of a SCC as the nodes that can reach the SCC, and the *out-component* of a SCC as the nodes that can be reached from the SCC. In biological networks, nodes in each of these subsets tend to have a common task. In signaling networks, the nodes of the in-component represent signals or their receptors and the nodes of the out-component are usually responsible for the transcription of target genes or for phenotypic changes. [28] For example, the in-component of the T-LGL network on Fig. 1.2 includes 6 source nodes, while its out-component consists of three sink nodes and P27.

Another useful centrality measure is betweenness centrality. The *betweenness centrality* of node k is given by

$$g_k = \sum_{i \neq j \neq k} \frac{C_k(i,j)}{C(i,j)}, \tag{1.1}$$

where $C(i,j)$ is number of shortest paths between node i and j and $C_k(i,j)$ is how many of these pass through node k. [29] For example, if we want to calculate the betweenness centrality of CIA, then i and j could be CI and wg, thus $C_{CIA}(CI,wg) = 1$ and $C(CI,wg) = 2$, the ratio of the two quantities is 1/2. The betweenness centrality is the sum of such ratios among all possible pairs. Betweenness centrality tends to be a better importance measure than node degree.

A network module has many inside edges but few edges going outside the module. There are several possible more specific definitions of modules, and many methods to identify network modules. [10,11] One method of module detection is based on adjacent k-cliques, where a k-clique is a complete undirected network of k nodes. [30] Two k-cliques are adjacent if they share $k-1$ nodes. The k clique module is the union of all k-cliques that can be reached from each other through a series of adjacent k-cliques. Palla *et. al* applied this method to detect modules in the protein-protein interactions network of S. cerevisiae, and demonstrated that the proteins in the detected modules have a shared functional classification. [30]

Network motifs are recurring patterns of interconnection with well-defined topologies. [7] Among these motifs are *feed-forward loops* (in which a pair of nodes is connected by both an edge or short path and a longer path) and *feedback loops* (directed cycles). For example, in the T-LGL leukemia network shown in Fig. 1.2, nodes STAT3, P27 and Proliferation form an incoherent feed-forward loop, since the two paths from STAT3 to Proliferation have different signs. Feed-forward loops are more abundant in transcriptional regulatory and signaling networks of different organisms compared to randomized networks that keep each node′s degree. They were found to support several functions such as filtering of noisy input signals, pulse generation, and response acceleration. [7] Positive feedback loops were found to support multi-stability while negative feedback loops can cause pulse generation or oscillations. [31]

Software packages for network visualization and analysis include yEd Graph Editor, Cytoscape [32], NetworkX [33] and Pajek [34].

1.3 Logic modeling of the dynamics of molecular networks

1.3.1 Introduction

Network representation and analysis provide insight into the connectivity between inputs and outputs and the importance of mediator nodes in the molecular system. However, as each node represents a specific molecular species in the molecular network, it also has an abundance associated with it and this abundance can change in time. Thus we need a second, dynamic layer in addition to the static network representation to model the cell behavior. We assign each node a variable x_i to represent its state or abundance. The value of this state variable (or, simply said, the state of the node) will depend on the state of the node$'$s regulators (which are specified by the network). Then the states of the nodes (or of a subset) can be used to represent a certain cell function or behavior. [4, 9] For example, in the T-LGL network a high value for the state variable of Apoptosis indicates that the cell committed to the cell death process, and a zero or low value of Apoptosis, coupled with abnormal values of other nodes (shown as node colors in Fig. 1.2), indicates the abnormal survival state of leukemic cells.

Dynamical models can be classified into continuous or discrete depending on whether the state variables are continuous or discrete. In continuous dynamical models, the rate of change (time derivative) of each node state x_i is expressed as a function of other variables in the molecular network. Thus the regulatory relationships are described by a system of ordinary differential equations (ODE). [35, 36] Continuous models are optimal for well characterized systems, where the mechanistic details for each interaction, the regulatory functions$'$ form and their parameters$'$ values are well known through collecting a sufficient amount of quantitative information (usually through decades of experimental work). However, this is usually not the case in molecular systems involving large numbers of heterogeneous chemical substances: not all interactions have been established, the underlying mechanisms are not known and the kinetic parameters are difficult to measure or estimate. Thus continuous modeling is not fit for these types of systems.

Discrete dynamical models use discrete variables to represent logic categories of node abundance and describe the future state of each node as a function of the states of its regulators in the molecular networks. The discrete models only require qualitative or relative measurements, demand no or very few kinetic parameters and yet can provide a qualitative dynamic description of the system. [9] Also, there is increasing evidence that the responses to signals in molecular networks (the so-called dose-response curves) show sigmoidal functional forms, which provides a rationale to describe the responses with discrete variables. For example, the MAP kinase cascade has sigmoidal regulatory functions at each level, and overall leads to a step-like input-output relationship. [37] Certain network motifs show parameter-independent input-output characteristics or outcomes that are robust to changes in parameter values. [19, 37] Taken together, this evidence makes it possible for us to use discrete models to capture the characteristics of real molecular systems. These discrete dynamical models, including Boolean network models [38], multi-valued logical models [39] and Petri nets [40] have been employed to study various systems in unicellular organisms, plants, animals and humans. [1–3, 5, 41–45]

Choosing the right dynamical model involves striking a balance between modeling detail and scalability. The hypothesis behind discrete dynamical models is that for certain classes of systems, the kinetic details of individual interactions are less important than the organization of the regulatory network. [1, 37, 38] Boolean networks are the simplest discrete dynamic models. In the following, we introduce the definitions of a Boolean network, sketch the steps in constructing a Boolean model of a molecular network, and discuss several obstacles and possible solutions.

In a Boolean network, each node state σ_i is a *binary variable*, either 0 or 1. The value $\sigma_i = 1$ (ON) represents that the node (*i.e.* gene, protein or molecule) is active or expressed, or is above certain concentration threshold; while the value $\sigma_i = 0$ represents that the node is inactive, not expressed, or is below certain concentration threshold. The threshold may not need to be specified as long as it is clear that such threshold exists, above which the component will effectively regulate the downstream nodes. The state of the entire system will be represented as a vector $(\sigma_1, \ldots, \sigma_N)$. The regulation relationships are described by the governing equations $\sigma_i^* = f_i$, which means that the future node state σ_i^* is determined by the Boolean regulatory function f_i (also called Boolean rule) of its regulators. There are two ways to specify the Boolean function. The first intuitive way is to write it in terms of the logic operators AND, OR and NOT. For example, $\sigma_4^* = f_4 = (\sigma_1 \text{ OR } \sigma_2) \text{ AND } (\text{NOT } \sigma_3)$ means that σ_4 will be ON when σ_3 is OFF and simultaneously at least one of σ_1 or σ_2 is ON. The implicit order of precedence of logical operators may be used: NOT has higher precedence than AND, and AND

Table 1.1: Truth tables illustrating the NOT, OR and AND operators. The first two columns list all the possible configurations for the two input nodes A and B. The third to fifth columns give the output value of the corresponding input configuration in the same row for the three functions NOT σ_A, σ_A OR σ_B, σ_A AND σ_B respectively.

σ_A	σ_B	f_C = NOT σ_A	f_D = σ_A OR σ_B	f_E = σ_A AND σ_B
0	0	1	0	0
0	1	1	1	0
1	0	0	1	0
1	1	0	1	1

has higher precedence than OR. Thus the above Boolean rule can also be written as $f_4 = (\sigma_1$ OR $\sigma_2)$ AND NOT σ_3, but it is different from $f_4 = \sigma_1$ OR σ_2 AND NOT σ_3. The second way to express a Boolean function is through a truth table, where we specify the output value for each possible input configuration. If node σ_i has k regulators, we will have 2^k input configurations since each regulator has two possible states. For example, the three basic logic operators, NOT (third column), OR (fourth column), and AND (last column), can be written as shown in Table 1.1.

1.3.2 Procedures to construct Boolean networks

We first outline the whole procedure to develop a Boolean model of a molecular network then give the details in the following paragraphs. One starts to build the Boolean model by establishing the list of nodes and of the known interaction and regulatory relationships among these nodes. One then needs to determine the Boolean regulatory function of each node. One also needs to determine the relevant initial conditions and choose an updating scheme to model the passing of time. Model construction is followed by model analysis, including determining the long-term behavior of the model. The analytical results need to be compared with established experimental results. If there are discrepancies, one needs to iteratively revise the Boolean model, including the network topology or the Boolean regulatory functions until the model is consistent with known behavior. Then one can use the Boolean model to make novel predictions awaiting experimental confirmation.

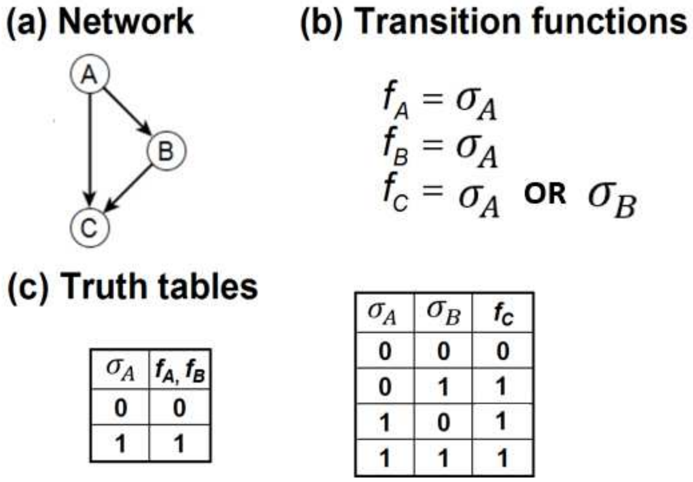
The first step in constructing the Boolean network is to collect information about the network nodes and interactions. We would need to integrate and assemble information from several experiments, for example high-throughput gene expression, proteomics and metabolomics data or detailed studies of individual interactions. [9, 46] High-throughput phosphoproteomics, protein-DNA interaction and genetic interaction studies can be used for two purposes: to determine the meaning of the binary states of components in known conditions (in a comparative manner, or by using a threshold), or to infer casual relationships between components. These casual relationships can be represented by a directed edge from one node to another in the network. Often the sign of the edge, positive (activating) or negative (inhibitory), can also be inferred. We can construct the molecular network if the totality of relevant information is sufficient. [9,46] Readers interested in how to deal with incomplete information can refer to [47–49].

The next step is to determine the Boolean regulatory function for each node. When there are multiple regulators for a node, we select the function that best represents the existing knowledge about their action. The OR function would be used if the node can be activated by any of its regulators. The AND function would be used if the node needs all of its regulators to be activated. If the Boolean regulatory function involving several regulators cannot be fully determined, one needs to take the trial and error approach to select the function that can successfully reproduce the existing experimental result (both at the node and at the whole network level).

For example, let us determine a compatible Boolean regulatory function for the three-node feed-forward motif shown in Fig. 1.3. A natural choice for source node A is $\sigma_A^* = f_A = \sigma_A$ as it represents that the signal of the system maintains a certain state for a certain period of time. As A positively regulates B, $\sigma_B^* = f_B = \sigma_A$. Node A and B positively regulates C. Then there are two compatible choices for the Boolean regulatory function of node C: $f_C = \sigma_A$ OR σ_B, and $f_C = \sigma_A$ AND σ_B. The results of knockout experiments (wherein one node is set into the OFF state) can help us determine which one is more appropriate. Let′s assume that providing A and simultaneously knocking out B resulted in the activation of C. This means that A alone can activate C, and thus $f_C = \sigma_A$ OR σ_B.

The next step is to determine the relevant initial condition for the system, *e.g.* the system′s natural resting state. When the relevant initial condition is not accessible, one can sample from different initial conditions in the state space. We note that the biologically relevant initial conditions may occupy a small region in the state space.

One also needs to choose a time implementation and *updating regime* for the system to evolve. Time is often implemented as a discrete variable, that is, the node states are updated at fixed time steps and their values are kept the same between time steps. [9,46] The timescale of the processes represented as edges can vary from fractions of a second to



σ_A	f_A, f_B
0	0
1	1

σ_A	σ_B	f_C
0	0	0
0	1	1
1	0	1
1	1	1

Figure 1.3: A Boolean model of a simple signal transduction network. (a) The graphical representation of the network. The edges with arrow-heads represent positive regulations. Note that the Boolean regulatory function for node C is not uniquely determined by the network representation. (b) The Boolean regulatory functions for each node in the model. (c) The truth tables of the Boolean regulatory functions given in (b).

several hours depending on the biological process. [7] Mathematically, we use the vector $(\sigma_1(t),\ldots,\sigma_n(t))$ to represent the state of the system at time t. Then we determine the value of each node state $\sigma_i(t+\tau_i)$ in the next time step based on the Boolean regulatory function, that is, $\sigma_i(t+\tau_i) = f_i(\sigma_{k_1}(t),\ldots,\sigma_{k_i}(t))$, where the τ_i is the time step for node i and $k_1,\ldots,k_i$ are the regulators of node i.

As we need to update all the nodes to obtain the system′s evolution trajectory, we also need to specify the order of updating each node. The simplest updating regime is synchronous updating, wherein all the nodes are updated simultaneously. This is equivalent to setting $\tau_1 = \cdots = \tau_n$ as a time step. [9,46] Thus the synchronous updating regime implicitly assumes that the timescales of all biological events are approximately the same so that the state change of each node is synchronized. In biological systems that include biological events of different timescales (*e.g.* include both transcriptional and post-translational regulation) it is not appropriate to use synchronous updating.

In order to take into account variations in timescale, different asynchronous updating regimes were developed. In deterministic asynchronous updating, a fixed timescale or time delay is used for each node. In the stochastic asynchronous regime, the system is updated in a random way. To be specific, in random order asynchronous update, a random permutation of a sequence of all the nodes is generated for each round and the nodes are updated in the order of the simulated permutation; this process is repeated until convergence. [9,46] Thus, in this regime, every node will be updated once during each round.

Another popular stochastic asynchronous update method is general asynchronous update, where a randomly selected node is updated in each time step. [9,46] Thus, in contrast with random order asynchronous regime, it is possible that one node is updated several times before another node gets updated next. However, since the node is randomly selected, the expected number of updates is the same for all nodes. If we know that nodes should be updated with different frequencies, we can use an update probability distribution.

Let us continue with the three node motif in Fig. 1.3 to illustrate two deterministic and two stochastic updating regimes. In synchronous updating, the state transitions will be $\sigma_A^* = \sigma_A(t+1) = f_A(t), \sigma_B^* = \sigma_B(t+1) = f_B(t), \sigma_C^* = \sigma_C(t+1) = f_C(t)$, where the nodes$'$ future states (at time $t+1$) are determined simultaneously, using their current node state at time t. In a deterministic asynchronous updating regime, say $\tau_A = 1$, $\tau_B = 2$, $\tau_C = 3$, the system will be updated in a pattern with period of 6: A alone, A and B together, A and C together, A and B together, A alone, A, B and C together. For random order asynchronous updating, there are 3!=6 ways of ordering these three nodes, at each time one ordering will be randomly selected. The order of update in our example can be $A,C,B;A,B,C;B,C,A;A,C,B;\ldots$ where semicolons indicate the end of a time step. In general asynchronous update, a possible update order for the three nodes system could be $A,B,C,B,B,C,B,A\ldots$. Notice that node B has been updated 4 times until A was updated again in this particular realization.

After the model is completely specified, we need to determine its long-term behavior. Since the Boolean network is a finite system, the state of the system will evolve into a single state (steady state) or a set of recurring states (a complex attractor). These steady states or recurring states are collectively called as attractors. [9,46] Attractors of molecular networks have corresponding biological meanings. A steady state or a group of steady states with similar function can be associated to a cell state or phenotype. Complex attractors can be interpreted as cyclic or oscillatory behavior such as the cell cycle, circadian rhythms or Ca^{2+} oscillations. [41,47,50]

A compact visualization of all possible trajectories is given by the *state transition graph* (STG), wherein each node is a possible state of the system, and each directed edge represents a possible transition from one state to another state in one update. [9,46] The

STG will contain 2^N nodes for a Boolean network with N nodes as it contains all the possible states in the state space. For example, the state transition graph of the three node Boolean model in synchronous updating regime is shown in Fig. 1.4.

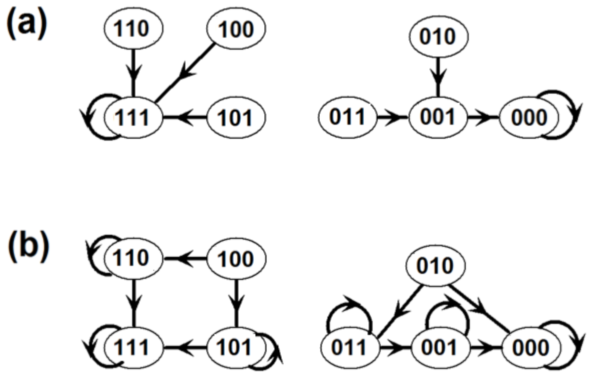


Figure 1.4: State transition graphs of the Boolean model presented in Fig. 1.3. A node represents a state of system, written in the order A, B, C; thus 111 represents $\sigma_A = 1$, $\sigma_B = 1$, $\sigma_C = 1$. A directed edge between two states indicates a possible transition from the first state to the second in one update specified in the updating scheme. A loop (an edge that starts and ends at the same state) indicates that the state does not change during update. (a) The state transition graph under synchronous update. The two states that have loops are the fixed points of the system. (b) The state transition graph under general asynchronous update (update one random node at a time). Though several states have loops, only the two states that have no outgoing edges are fixed points of the system.

In the state transition graph, a steady state will be a node with a loop and no other out-going edge; a complex attractor will be a strongly connected component without an outgoing edge. For example, in Fig. 1.4, the state 111 and state 000 only have a loop and no other out-going edges, indeed they are the steady state of the three node system in Fig. 1.4. Notice that this criterion can be used to identify steady states, however, it won′t be an efficient way as it requires to map the entire state transition graph first.

For each attractor, all the states that can reach the attractor in the state transition graph are called the basin of attraction. For example, in Fig. 1.4, the basin of the steady state 111 includes state 100, 110, 101, and 111; while the basin of the steady state of 000 includes state 000, 010, 001 and 011.

It′s an interesting question whether the chosen updating regime will have an impact in the properties of attractors and state transition graphs. Let us start with the steady state (fixed point) type of attractor. In a steady state the future state, *i.e.* the outcome of the

Boolean regulatory function, equals the current state for each node. This requirement is time independent, therefore steady states are independent of updating regimes. Indeed, in Fig. 1.4 the steady states of the Boolean model under the two updating regimes are the same. However, based on the example above, one can readily see that the state transition graph is different for the two different updating regimes. In synchronous updating, since all nodes are updated simultaneously, each state can only have one out-going edge. Due to this, the complex attractor in synchronous updating regime is also called a limit cycle as the set of recurring states repeats in a fixed order. [9] Also, the basin of attraction for each attractor will be separated as one state can only follow a unique path in STG. One can see that the STG under the synchronous updating shown in Fig 1.4(a) follow the descriptions above. While in asynchronous updating, each node can have multiple out-going edges due to the different updating order as illustrated in Fig 1.4(b). Thus states in the complex attractor can appear in an aperiodic manner.

Some limit cycles can only be observed under synchronous updating and any perturbation of the updating timescales will eliminate the attractor. [51] One can readily see this in the example shown in Fig. 1.5: the limit cycle between the states 001 and 010 is not observed under general asynchronous update, where two successive states can only differ in one node′s state. The figure also exemplifies that the basin of attraction of the attractors may overlap due to the randomness in the updating regime. The state transition graph can be seen as a graphical representation of a corresponding Markov Chain model, where each node is a state in the Markov chain and each edge corresponds to a transition with non-zero probability between states. If complete randomness is guaranteed, the system is taking a random walk on the state transition graph, which specifies a unique Markov Chain model. [9]

At the end, one needs to compare the model′s results with established experimental results. If there are discrepancies, one needs to revise the Boolean network or the Boolean regulatory function. [9,52,53] Boolean network should qualitatively reproduce properties demonstrated in biological systems including homeostasis or multi-stability. [7,38] In the next subsection, we use two biological examples to illustrate this point.

Dynamical models can also be used to make novel predictions, such as predictions about the effect of perturbations and about network control strategies. [6,9,54,55] These predictions can provide insight about the biological system and guide future experiments. In perturbation analysis, we determine the change in the attractors induced by external or internal perturbations, including knockout or constitutive expression/activity of a

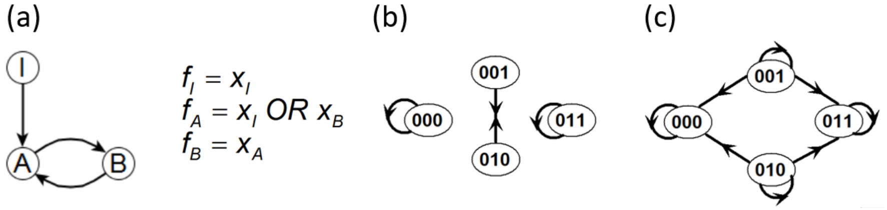


Figure 1.5: A simple three nodes network. (a) The network representation and corresponding Boolean rules. Node A and B form a positive feedback loop. Node B and source node I can independently activate node A. (b) The network′s (partial) state transition graph under synchronous update when the signal is set as OFF ($\sigma_I = 0$). The states are specified in the node order I, A, B. (c) The state transition graph under general asynchronous update when the signal is set as OFF ($\sigma_I = 0$). The figure is adapted from [4].

node. Node knockout can be modeled as fixing the corresponding node in the OFF state, while constitutive expression/activity can be modeled as fixing the node in the ON state. Transient perturbations can be modelled as temporary changes in the node′s state and letting the system evolve as before. [55] The perturbation analysis can predict changes in the attractors and their basin induced by each possible perturbation. Thus those perturbations that lead to a dramatic cascading effect will be identified, which helps us identify components key to maintaining a phenotype in a biological system. In a signal transduction network involved in a disease, the identified key components could be targets of therapeutic interventions. [3, 5, 47]

Several software tools are available for Boolean dynamic modeling of biological systems. The CoLoMoTo (Consortium for Logical Models and Tools) is a platform providing resources in logical modelling, including software tools and biological models. [56] Among them, SBML qual is an open-source model library, promoting a standard format to analyze and exchange qualitative models. [56] GINsim is a free Java software application for logical modelling of regulatory and signaling networks. [57, 58] It allows users to define a model or import models in various formats. It also supports simulations of logical models and generates state transition graphs under various updating regimes. The R package BoolNet provides attractor search and robustness analysis methods for synchronous, asynchronous and probabilistic Boolean models. [59] In addition, BooleanNet is a python package that can be used to simulate synchronous and random order asynchronous models and to determine their state transition graph. [60] There are other existing simulation and analysis software tools for logical models, including

ADAM [61], the Cell Collective [62], CellNetAnalyzer [63], CellNOpt [64], ChemChains [65], Odefy [66], SimBoolNet [67] and SQUAD [68].

1.3.3 Two biological network examples

It has been shown that Boolean models can capture characteristic dynamic behavior, such as excitation-adaptation behavior and multi-stability, as continuous models do. [37] For example positive feedback loops support multi-stability, coherent feed-forward loops support the filtering of noisy input signals, and incoherent feed-forward loops support excitation-adaptation behaviors. [7, 31, 37] The reader interested in the details of these examples can refer to [37].

Here we illustrate the capacity of Boolean network models to capture dynamic behavior using two biological network examples. The first one is the T cell Large Granular Lymphocyte Leukemia (T-LGL) network, which is mentioned in Sec. 1.2.1 and shown in Fig. 1.2. T-LGL leukemia is a rare blood cancer. While normal T cells undergo activation induced cell death (apoptosis) after successfully fighting a virus, leukemic T-LGL cells survive. Through an extensive literature search, Zhang *et al.* constructed a Boolean network model of T-LGL leukemia, which can reproduce the abnormal survival of T-LGL cells and other known experimental results of the system. [2] The details of the T-LGL model, including the Boolean regulatory functions, can be found in [2]. Zhang *et al.* chose a stochastic asynchronous updating regime. The model has two steady states under the relevant source node initial condition (Stimuli, IL15 and PDGF are ON and Stimuli2, CD45, and TAX are OFF.) [2] The two steady states respectively correspond to the apoptosis of T cells and survival of the abnormal T cells as seen in T-LGL leukemia. This is an example that the Boolean model successfully reproduces the qualitative experimental result and captures the multi-stability of a real system. Full analysis of the state space was not possible due to the large size of the network, thus follow-up work employed network simplifications to reduce the network size and the state space. Two kinds of network reductions were applied, both of which have been shown to preserve the attractor repertoire of the system . [69] First, one can determine and eliminate the nodes whose state stabilizes due to their regulation by sustained signals. Second, one can iteratively collapse nodes with one incoming and one outgoing edge, for example, node MCL1 could be removed in the Fig. 1.6(a). One can obtain a reduced network with 18 nodes after applying the first kind of reduction and a reduced network with 6 nodes after both reductions. Now itâĂŹs much easier to visualize the state space of the T-LGL leukemia network, which is shown in Fig. 1.7. Perturbation analysis of the Boolean model in Fig. 1.6(b) reveals that permanently reversing the node state of S1P, Ceramide or DISC in the T-LGL leukemia steady state can eliminate the T-LGL steady state and lead to apoptosis. [3] Nodes such as S1P, Ceramide or DISC can be called key mediators of the T-LGL state. These key mediators are candidate therapeutic targets, which is supported by experiments (one of which was performed to test this prediction). Similar analysis of the original 60-node Boolean model in Fig. 1.2 identifies 15 key mediators in the original network, which are also candidate therapeutic targets. [3]

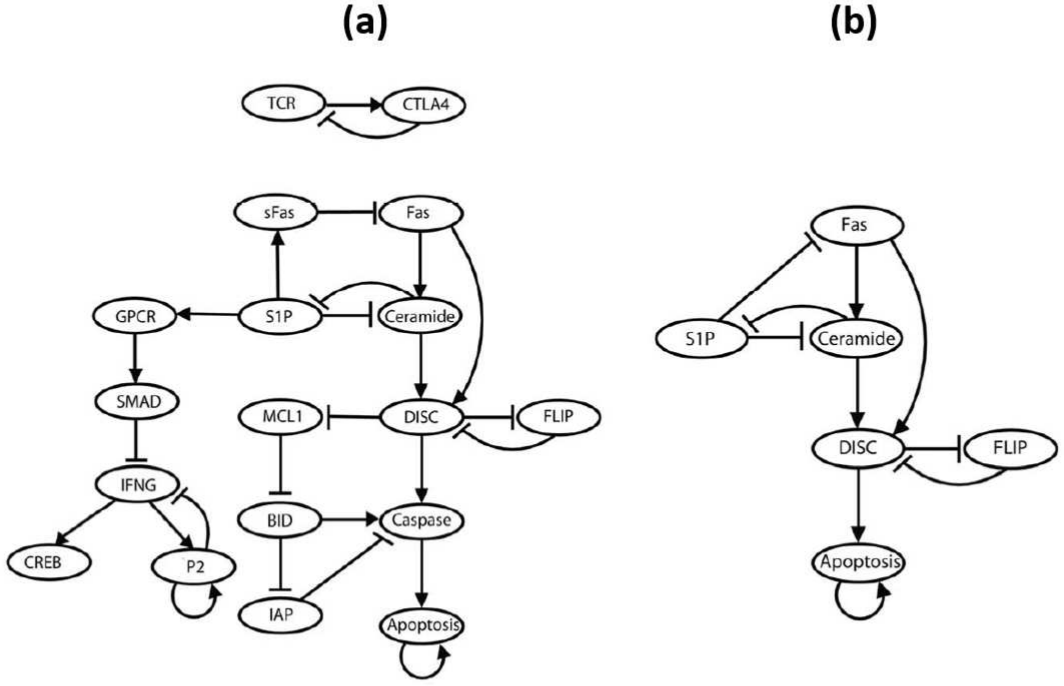


Figure 1.6: Reduced T-LGL leukemia signaling network. An arrow-head indicates a positive edge, and a blunt segment indicates a negative edge. (a) The 18-node network obtained by removing stabilized nodes due to the sustained state of source nodes. (b) The 6-node sub-network obtained by merging mediator nodes from the bottom subgraph in part A. This figure is reproduced from [3].

The second example is the Epithelial-to-Mesenchymal transition (EMT) network. EMT is a cell fate change, during which epithelial cell lose their original adhesive property, leave their primary site, invade neighboring tissue, and migrate to distant sites as mesenchymal cells. [5] EMT plays important roles in pathological processes, including the invasion process in hepatocellular carcinoma (HCC), thus it is important

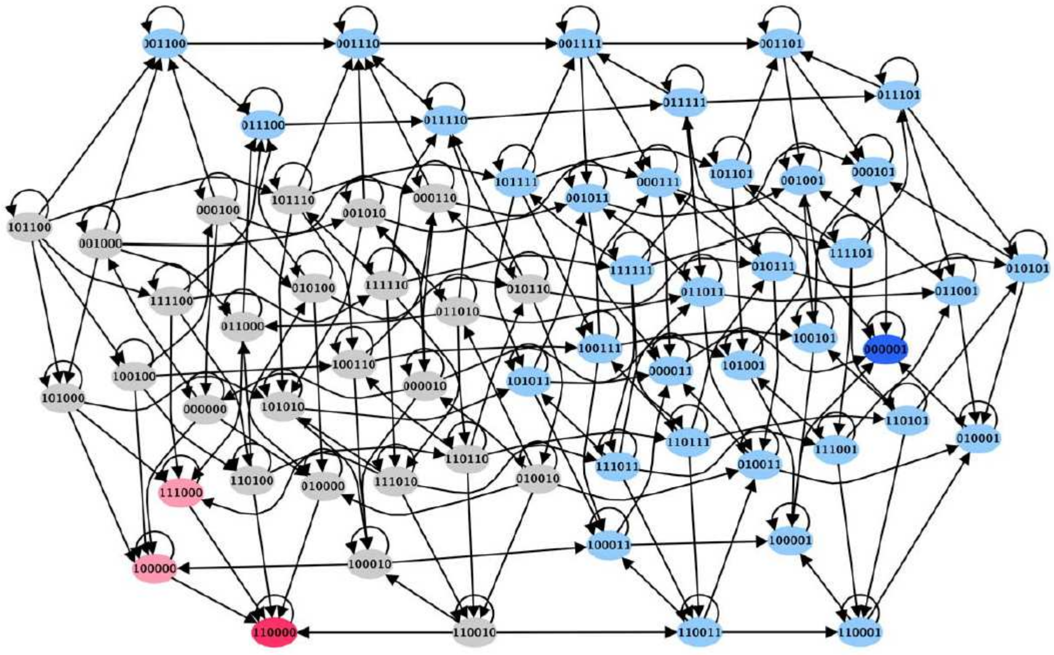


Figure 1.7: The state transition graph of the reduced 6-node subnetwork of T-LGL leukemia network shown in Fig. 1.6(b). There are 64 possible states in the state space. The dark blue node represents the normal steady state (Apoptosis of T cell) and the red node represents the T-LGL leukemia steady state. The light blue states are transient states that will evolve into the normal steady state (dark blue) and the pink states are transient states that will evolve into the leukemia steady state (red). Gray states are transient states that can evolve into either steady state. This figure is reproduced from [3].

to understand this signaling process and design strategy to suppress it. A hallmark of EMT is the loss of E-cadherin, a cell adhesion protein and EMT can be induced by transforming growth factor-β (TGFβ), growth factors and other external signals. [5] Through extensive literature search, Steinway *et al.* built a Boolean network model of EMT in the context of HCC invasion. [5] This network contains 69 nodes and 134 edges. In this network, E-cadherin is the sole negative regulator of the sink node, which is a conceptual node to represent the occurrence of EMT. The model is updated in a ranked asynchronous updating regime to account for the fact that the relevant signal transduction events occur substantially faster than the involved transcriptional events. [5] Simulations of the Boolean model can reproduce the EMT driven by TGFβ: starting from an epithelial state (which is an attractor of the signal-free system), and activating the TGFβ signal, the system will evolve and finally stabilize into a mesenchymal state. [5] With TGFβ fixed to be ON, the model can be reduced to a network with 19 nodes and

70 edges after applying similar network reduction techniques as in the T-LGL leukemia network. The mesenchymal state is the only steady state of the reduced network, which is confirmed by exploration of the state space of the reduced network. This suggests that the system will ultimately end in a mesenchymal state with the signal of TGFβ. A systematic search revealed that there are seven nodes, whose individual knockout can prevent TGFβ driven EMT. (*i.e.* inhibit the transition from the epithelial state from transitioning into the mesenchymal state in response to TGFβ) These seven nodes are all transcription factors that directly regulate E-cadherin. The effectiveness of the knockout of these transcription factors was already established experimentally, but unfortunately currently it is not possible to target these transcription factors by drugs. There are also six two-node knockouts (not involving any of the previous seven nodes) that can suppress TGFβ driven EMT. All these six knockout pairs require the inhibition of the SMAD complex. If constitutive activation is also considered, one new single-node intervention target, miR200, and one new two-node combination are identified. [5]

1.4 Connecting the structure and dynamics of molecular networks

There is increasing evidence that the dynamics of certain systems is not sensitive to the details of the interactions and to the kinetic parameters, which inspired researchers to explore the effect of the underlying network topology on the network dynamics. [7,37,38] Multiple lines of research have been devoted to shed light upon this subject; we will introduce several tools developed to analyze this relationship in this section. [6,54,70,71]

We first discuss network structural features that influence the attractor repertoire of Boolean models. As hypothesized and later verified by researchers, feedback loops play an important role in determining the network attractors. [31] Rene Thomas conjectured that necessary condition for a system to have multi-stability is the existence of a positive feedback loop and a necessary condition for a system to have sustained oscillations is the existence of a negative feedback loop. [31] This indicates that the sign of the cycles in the network determines the dynamical behavior of a system. An additional important feature, which is not explicitly represented by the interaction network, is the possible dependence or combinatorial effect of multiple incoming edges to a node. This motivates us to integrate a network representation with the Boolean regulatory functions of each

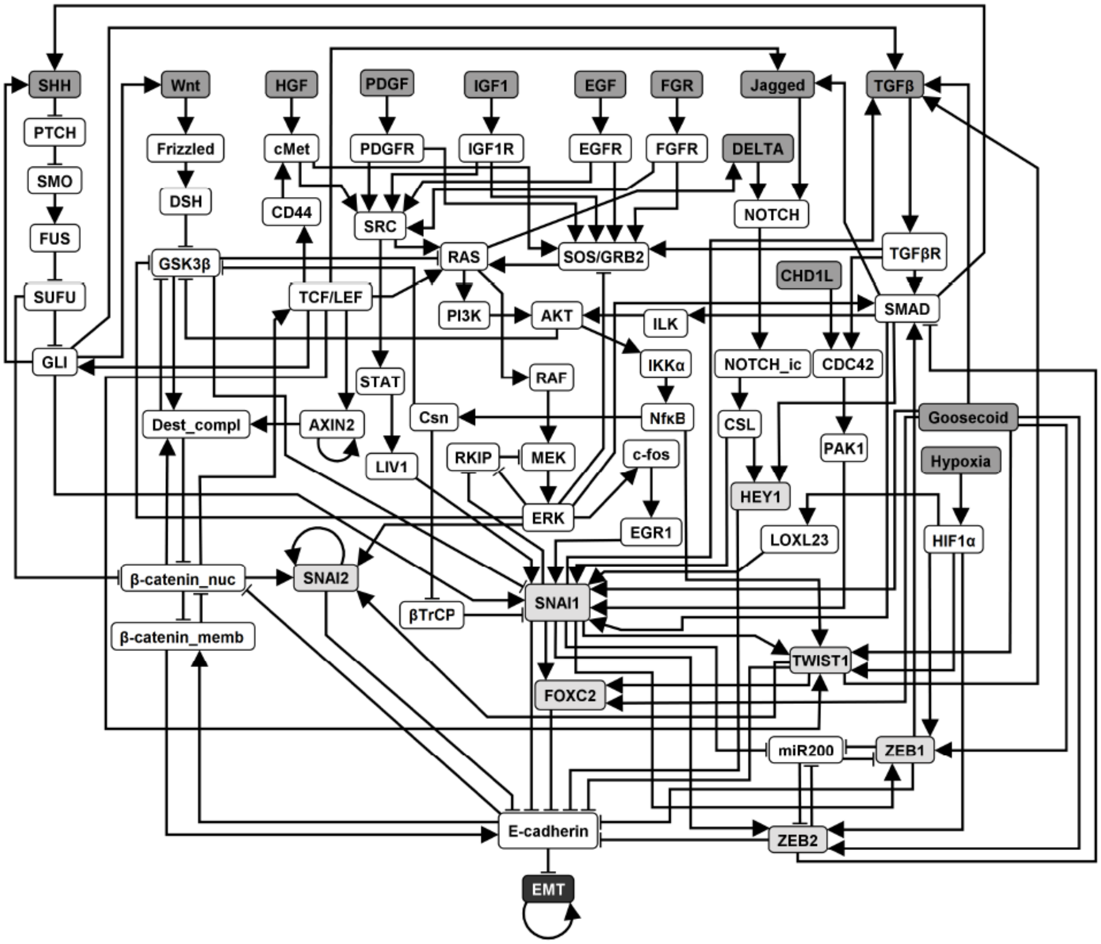


Figure 1.8: The EMT signaling network in HCC, which consists of 69 nodes and 134 edges. Signals are in dark gray fill, transcriptional regulators of E-cadherin are in light gray fill. The output node EMT is marked with black background. Positive edges are drawn with arrow-heads and negative edges terminate in blunt segment. The figure is reproduced from [5].

node into a so-called expanded network. [70]

1.4.1 Expanded network

We introduce the concept of expanded network with the example in Fig. 1.9, which consists of five nodes, the input I, and the regulated nodes O, A, B and C with the regulatory functions $f_A = \sigma_I, f_B = \sigma_A$ AND (NOT σ_C)$, f_C =$ NOT $\sigma_B, f_O = \sigma_A$ OR σ_B. First, we introduce a complementary node for each original node in the system to represent the negation (deactivation) of the original node, denoted by the real node′s name preceded with $\sim$. In all the Boolean regulatory functions, all the NOT functions are replaced by the negation state of the respective node (*i.e.* its complementary node) since the NOT function is a unary operators. The edges in the expanded network are redistributed according to the updated rules. For example, $f_C =$ NOT $\sigma_B = \sigma_{\sim B}$ and thus a corresponding edge is drawn from $\sim B$ to C in the expanded network. The Boolean regulatory function of a complementary (negated) node is the logical negation of the regulatory function of the original node. For example, $f_{\sim C} =$ NOT (NOT $\sigma_B) = \sigma_B$ and thus a corresponding edge is drawn from C to $\sim B$ in the expanded network. Thus the Boolean rules for all the complementary nodes in Fig. 1.9 are

$f_{\sim A} =$ NOT $\sigma_I = \sigma_{\sim I}$,

$f_{\sim B} =$ (NOT σ_A) OR $\sigma_C = \sigma_{\sim A} OR \sigma_C$,

$f_{\sim C} = \sigma_B$,

$f_{\sim O} =$ (NOT σ_A) AND (NOT σ_B) $= \sigma_{\sim A}$ AND $\sigma_{\sim B}$

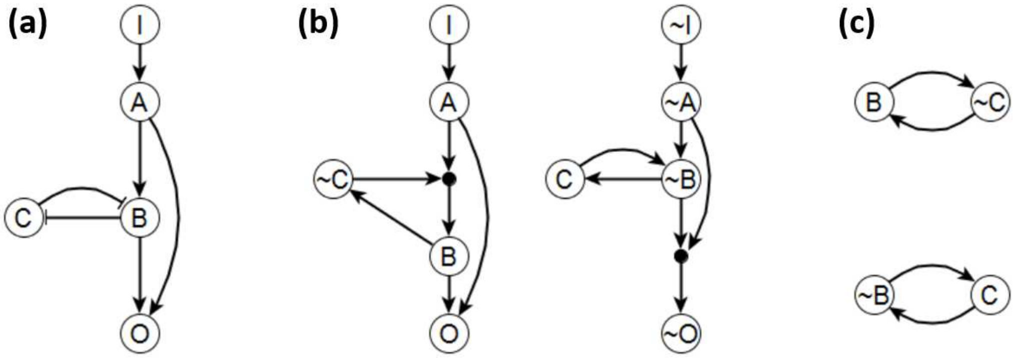


Figure 1.9: Illustration of the expanded network of a simple network. (a) A hypothetical signal transduction network similar to the reduced 6-node T-LGL leukemia network in Fig. 6b. (b) The expanded network of the given network in (a). The composite node is denoted by a solid circle. (c) The stable motif of the given network under a sustained signal input $x_I = 1$. The figure is adapted from [4].

Second, to differentiate AND rules from OR rules when considering the relationship of edges pointing toward the same target node, we introduce a composite node for each set of edges that are linked by an AND function. In order to uniquely determine the edges in the expanded network, the regulatory functions need to be specified in disjunctive normal format, that is, a disjunction of conjunctive clauses or grouped AND clauses separated by OR clauses. For example, (A AND B) OR (A AND C) is in a disjunctive normal form, while A AND (B OR C) is not. Algorithmically, the desired disjunctive normal form can be formed by a disjunction of all conditions that give output 1 in the Boolean table and then simplified to the disjunction of prime implicants (Blake canonical form) by the Quine-McCluskey algorithm. [72] Now we add a composite node for each AND clause in the Boolean regulatory function, denoted by a solid black node in Fig. 1.9. For example, the composite node in the left part of Fig. 1.9(b) represents the expression σ_A AND (NOT σ_C), which activates node B; the composite node in the right part of Fig. 1.9(b) represents the expression (NOT σ_A) AND (NOT σ_B), which "activates" the complementary node $\sim O$. Notice that one can read all the regulatory functions from the topology of the expanded network. The AND rule is indicated by a composite node with multiple regulators, while all the other edges represent independent activation (parts of an OR function).

1.4.2 Stable Motif

As the expanded network contains the essential information that determines the network dynamics, the expanded network serves as a basis for network reduction and attractor analysis, *i.e.* the dynamical information. One approach is through analyzing the stable motifs of the expanded network. [54] A stable motif is defined as the smallest strongly connected component (SCC) satisfying the following two properties: 1) The SCC cannot contain both a node and its complementary node and 2) If the SCC contains a composite node, it must also contain all of its input nodes. [54] The first requirement guarantees that the SCC does not contain any conflict in node states and the second requirement guarantees that all the conditional dependence is satisfied and the SCC is self-sufficient in activating each node state inside the stable motif. Thus the stable motif represents a group of nodes that can sustain their states irrespective of other outside nodes$'$ states. The corresponding node states implied by the stable motif can be directly read out: the original node represents the ON (1) state and the complementary node represents the OFF (0) state. [54] For example, in the top stable motif of Fig 1.9(c), the stable motif represents that B is ON and C is OFF.

Once we find a stable motif, we can plug in these node states into the Boolean regulatory functions and obtain a simplified network corresponding to this stable motif. We can identify all the stable motifs of this simplified network and repeat the process.

The results of this iterative process can be represented as a stable motif succession diagram. [6] For example, the stable motif succession diagram of the T-LGL network is shown in Fig. 1.10. [6] After iterative identification of stable motifs and network reduction, we will obtain one of two final outcomes: all the nodes will be in a fixed state (either in a stable motif or fixed during network reduction) or some nodes are not in a fixed state and will be expected to have oscillatory behavior. In the first scenario, we obtained a steady state. In the second scenario, we obtained a quasi-attractor, which tells us the fixed node states and potential oscillatory nodes among all states of a complex attractor. [6] Thus stable motif analysis can be used as a preliminary analysis or substitute for attractor analysis depending on the level of detail we care about. For example, in the T-LGL network, successive stabilization of stable motif shown in Fig. 1.10 will ultimately drive the system to one of the two steady states, the Apoptosis steady state or the T-LGL leukemia steady state. [6]

We are not only interested in building the molecular network to understand the underlying biological process, but also in designing interventions or therapeutic strategies to drive the system from an initial state to a desired state or attractor. The stable motif succession diagram readily implies a control strategy called stable motif control as the sequential stabilization of each stable motif in the stable motif succession diagram guarantees that the system will reach the desired attractor. We can control the system by controlling the corresponding stable motifs. [6] For example, sequential stabilization of the three motifs in the first line in Fig. 1.10 will drive the system to the normal steady state (Apoptosis). However, the control strategy does not need control of all the nodes involved as two types of reductions can be done. [6] First, not all stable motifs need to be controlled. If there is a branch-free line of stable motifs after a particular stable motif, or if all the branches lead to the same steady state in the succession diagram, then the stable motifs after this particular stable motif do not need to be controlled. For example, in the first sequence of stable motif in Fig. 1.10, one only need to control the first, cyan-colored stable motif. Second, not all the nodes in the stable motif need to be controlled in order to stabilize the stable motif. For example, forcing S1P in the OFF state is enough to stabilize the cyan stable motif in Fig. 1.10. Thus after these two levels of reduction in the control strategy, one would get a smaller set of nodes to drive the system to desired state, however, the intervention does not guarantee to be minimum in size. Readers interested in more mathematical or practical details can refer to [6, 54]. All these stable motif analysis and control strategies have also been applied to the EMT network, yielding

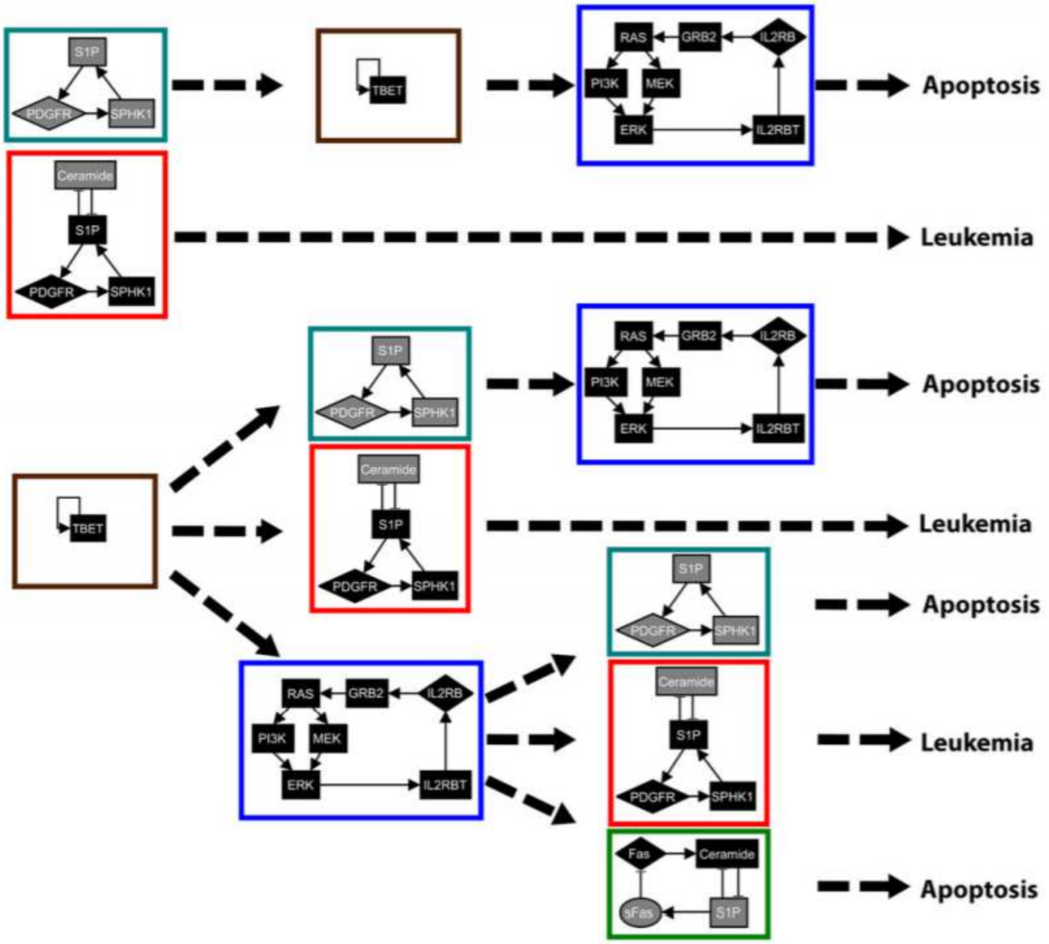


Figure 1.10: Stable motif succession diagram for the T-LGL leukemia network. Each colored rectangle represents a different stable motif. Inside the box, gray shaded nodes indicate nodes with ON state and black shaded nodes indicate nodes with OFF state. There are two possible steady-state attractors, the normal state of cell death (Apoptosis) and the diseased state (T-LGL leukemia). The attractor to which the sequence of stable motifs leads is marked at the rightmost. A dashed line pointing from a stable motif to a second stable motif means that the second stable motif can be found in the reduced network due to stabilization of the first stable motif. A dashed line pointing from a stable motif to an attractor means that applying network reduction with the fixed stable motif will lead to the attractor.The Figure is reproduced from [6]

strategies to prevent the system′s convergence to the mesenchymal state and to return the system to the epithelial state. [6, 54]

1.4.3 Elementary Signaling Mode

Another aspect of dynamical information about the network is to determine the contribution of each node into the system′s outcomes. Consider a network with a single source node (signal) and a single sink node that reflects the network′s output. The expanded network serves as an important tool to characterize the importance of intermediary (non-signal, non-output) nodes. One useful concept is called elementary signaling mode (ESM), which is defined as the minimal set of components able to perform signal transduction (*i.e.* able to functionally connect the signal to the output node) regardless of the rest of the network. [70] The elementary signaling mode will be a path or a subgraph connecting from the signal to the output node, which can be identified in the expanded network. For example, there are two ESMs between node I and node O: the path I, A, O and the subgraph consisting of I, A, the composite node, $\sim C$, B and O. One can show that they are both minimal as taking a node from the ESM will obstruct the signal from propagating. The elementary signaling modes can be used to rank the importance of the nodes in mediating the signal through studying the reduction in the number of ESMs due to the loss of the node (and of any other nodes that are lost as a consequence). [70] For example, node A appears in both ESMs found in Fig. 1.9, and its loss eliminated both, however node B only appears in one of the ESMs and its loss does not affect the other ESM. This suggests that node A is essential in the signal transduction process from node I to node O, while node B is not. In several examples of biological networks it was shown that the ESM-based analysis can identify essential nodes as effectively as a full dynamical analysis of the corresponding perturbed system. [70] ESMs, or more specifically, the number of node-independent ESMs, can also be used to quantify the system′s functional redundancy. [71]

1.4.4 Summary of Boolean Network

The improvements in experimental technology and the large amounts of generated data have brought us into an era where different types of dynamical models are needed to provide system-wide insights in biological systems. Although Boolean models are based on a series of assumptions and are limited in describing the quantitative features of dynamic systems, we have shown that they can capture emergent characteristics of real biological systems, demonstrate considerable dynamic richness and can predict successful intervention strategies in biological systems. Boolean network models do not require detailed knowledge of the kinetic parameters (as continuous models do), striking a balance between scale and realism. Their parsimonious nature makes them a preferred choice for systems where detailed quantitative experimental data is not available. Qualitative dynamical models, including Boolean network models, exist as a complement to quantitative dynamical models and will be often needed as we gradually develop our understanding of biological systems. The success of Boolean networks also indicate that in certain systems the behavior of the system is largely determined by the organization of the network structure rather than the kinetic details of individual interactions, which highlight the theoretical value of Boolean network models. In summary, Boolean networks serve as a useful foundation for modeling molecular systems; they can identify the network features (*e.g.* stable motifs) that are key determinants of the dynamics and whose detailed modeling would be most fruitful.

1.5 Network Control Problem

We have made a lot of progress in understanding and explaining the behavior of a complex system through building dynamical models as a first step. It is our ultimate goal to control the dynamical behavior of a complex system and learn the design principle of a complex system satisfying a specific application. [73, 74] Indeed, only after we are able to accomplish this goal, we can claim that we fully understand the behavior of a complex system.

Network control problems naturally arise in many different fields and are directly motivated by desired applications. [73, 74] For examples, in complex diseases such as cancer, we want to drive the biological system from a disease (cancerous) state into a healthy (normal) state. [6, 73] Also, as in infectious disease control, we want to design the most cost-efficient strategy to prevent the epidemic event from happening on a social network. [11] Many other examples can be given in technological network including electric circuit network, internet and power grid, ecological network and biological networks. [73]

The term of network control has a broad meaning due to its various applications as shown above. Control theory asks how to influence the system so that its dynamical output follows a desired goal (trajectory or final state). [73] However, different network control problems can have various setting in terms of control goals, intervention methods and the involved dynamical systems. These different settings corresponding to different realistic needs often lead to dramatic different solutions to the problem. [73] As to network control goals, we have full control [75, 76], attractor control [6, 77] and target control [78] in the decreasing order of difficulty. In full control, we want to drive the system from any initial condition to any final state. In attractor control, we want to drive the system from any initial condition to any natural existing attractor. In target control, we want to drive the system from any initial condition to a state with a subset of nodes in desired state (equivalently a specific basin in the state space). As to intervention strategy, different approaches include changing the parameters of the dynamical model, manipulating state variables through providing continuous or discrete signals [24, 77, 79], and modifying the network topology by introducing new interactions [55, 80] etc. Last, whether the dynamical model is continuous or discrete and linear or non-linear makes the problem having dramatic different level of difficulties. [73]

Fruitful progress has been achieved for control problems in linear ODE system since 1960s. Fundamental questions of controllability and observability are widely discussed before designing concrete strategy to control specific system. [73] It is possible to achieve full control for a linear time-invariant system and a well known criterion of controllability and accessibility was given by Kalman. [75] However, the goal of obtaining similar criteria of controllability test for non-linear system turned out to be too ambitious. Criteria for local accessibility and sufficient conditions for local controllability were established through the tool of Lie algebra for well described non-linear system (known dynamics and fixed parameter). [81, 82]

In spite of these rich results, we still lack of good understanding of control principles of complex networked system. The underlying reason is that we lack the complete accurate network descriptions of the system, the specific function forms of the dynamics and the precise parameters involved. [73] One approach is to start with a qualitative model well validated by existing experimental results such as Boolean networks. [9] Previous researches provide strategy of attractor control in Boolean network model such as stable-motif control. [6] In chapter 2, we design compensatory interactions to mitigate network deregulations in biological networks as preventive method or immediate treatment. In chapter 3, we design algorithms to solve the target control problem in Boolean network, which can be adapted to design mitigation strategy of stabilized deregulations. Another approach to make progress in spite of incomplete information is through structural control theory developed in 1970s [76], i.e. predicting controllability just based on the underlying network of the dynamical systems. Liu et. al. later use efficient algorithms on networks to identify drive nodes for full control of linear systems. [24] Dynamics involved in biological systems are mostly non-linear. Though linearizion of non-linear system around its equilibrium point (or nominal trajectory) generally leads to a linear time-invariant (or time-varying) system. However the linearized system only provide local information of the non-linear system of the equilibrium point. More importantly, the original system could be controllable while the linearized system is not controllable. [73] Mochizuki et. al. developed another structural-based algorithm for attractor control in non-linear system. [77] We applied both methodology to real biological systems to compare their differences as in chapter 4. We discuss possible future works in chapter 5.

Chapter 2 | Compensatory interactions to stabilize multiple steady states or mitigate the effects of multiple deregulations in biological networks

This Chapter is primarily based on a published work [80] , where I am the first author. Parts of reference are reproduced in this chapter with permission from Gang Yang, Colin Campbell and Réka Albert, Copyright 2014, American Physical Society.

2.1 Introduction

Complex networks are increasingly used to understand and simulate the behavior of biological systems such as cellular signaling networks [5, 8, 21, 36, 83, 84]. The network-based dynamic modelling approach aims to capture the biological function and behavior of these systems as an emergent property that arises from the totality of interactions among the components [9]. Several researchers have successfully used network-based approaches such as Boolean and logical models to study specific biological processes [5, 83, 85]. Complex diseases including diabetes and cancers can be modeled as network damage due to temporary or permanent node perturbation (e.g. constitutive activation of a protein arising from a genetic mutation) [5, 86]. Thus the topics of network repair and network control have drawn significant attention in the scientific community [24,55, 77,87]. Most approaches aim to influence network dynamics by controlling the states of certain nodes of the network [24,77]. Recently, another approach to the network control problem, namely modifying the interactions in the network, was proposed [55]. Using this approach, compensatory interventions can be found to stabilize an attractor (e.g. steady state) of the network after damage to a single node [55]. These interventions can be implemented as preventive measures or applied immediately after the onset of damage. The effect of the intervention is that the perturbation does not propagate to the rest of the network, and a close-to-normal behavior is restored [55]. Ultimately, a combination of node-based and edge-based approaches will provide researchers more potential therapeutic strategies.

Recent research suggests that complex diseases such as cancer often involve multiple gene mutations and the "one disease, one target, one drug" approach may not be effective to battle these diseases [86–88]. Thus it is worthwhile to use the network paradigm to explore the combinatorial effect of multiple gene mutations, and to design control measures to prevent these effects. Moreover, many biological systems were shown to have several possible steady states (e.g. several possible cell types), each reachable for alternative histories (time courses) [5,9,83]. Repair interventions should be cognizant of these alternative states and maintain (or eliminate) them as necessary or desired in the specific context. Here we generalize the method of Campbell *et al.* [55] to a multiple node damage setting, and to systems that have multiple steady states, aiming to provide a theoretical platform to mitigate damage more realistically.

We briefly repeat key concepts and notations of Boolean network and more details can be found in Chapter 1. A network is a mathematical abstraction of a set of relationships between various elements. The network consists of nodes that represent the different elements and edges that specify the pairwise relationships between them [11,22]. Biological networks were found to exhibit interesting topological properties such as a heterogeneous degree distribution [17,18]. However, in order to understand the biological function of a system, the network$'$s topological information alone is not enough and dynamical information should be incorporated. More specifically, in a Boolean dynamical model, each node i is characterized by a binary state variable σ_i, which can be 1(ON) or 0(OFF), and the vector $(\sigma_1,\cdots,\sigma_n)$ represents the state of the system [9]. The state of the system is followed at discrete time intervals. The activity of each node σ_i is described by a regulatory rule specificed by truth table $\sigma_i(t+\tau_i) = f(\sigma_{i_1}(t),\cdots,\sigma_{i_k}(t))$, where $i_1,\cdots,i_k$ are the regulating nodes of i and τ_i is a discrete time delay. The time trajectory of the system is simulated deterministically or stochastically depending on the updating scheme. For a finite system, the system will evolve from a given initial condition into an attractor, which can be a steady state (fixed point), several states that repeat regularly (limit cycle) or irregular repetition of a set of states (complex attractors). Steady states can be interpreted as cell types and limit cycles correspond to a cell cycle or circadian rhythms [9]. Fixed points (steady states) do not depend on the updating scheme [51]. Abnormal behavior of a certain element can be modelled as a change in the node state, either a temporary perturbation or permanent damage [5, 86]. Forf example, a loss-of-function mutation or the knockout of a gene can be represented as a permanent OFF state of the corresponding node in the network.

In this chapter, we presents three key results in the following. First, we use analytical and computational methods to study how network structure and regulatory logic affect the resilience of the network$'$s steady states to single node perturbation. Second, we present an algorithm to design compensatory interventions to stabilize a steady state of the network after double node damage and evaluate it on random Boolean networks and two biological examples. Third, we apply the algorithm on stabilizing two steady states simultaneously after a single node damage and discuss the emerging situations and their corresponding frequencies. We apply the algorithm to the biological examples and also adapt it to the alternative goal of stabilizing a steady state and destabilizing another.

2.2 Results

2.2.1 The influence of single node damage on a steady state of a system

We consider a Boolean model of a biological system; this model will have one or several attractors. We start from a steady states s. Then we consider damage to a node i by permanent knockout (sustained OFF state) or constitutive expression or activity (sustained ON state). If the damaged state s^* is a new steady state (i.e. other nodes are not affected by the perturbations), we say that steady state s is stable against the damage. In the converse case, the state of one or more nodes will change, which then has a cascading effect in the biological system. We say the steady state s needs repair in order to prevent damage propagation. We define the sensitive node set S_i as the set of nodes that would change their state as a direct consequence of the damage to node i.

Previous research has studied the relationship between a network$'$s structure and its topological resilience to incremental node loss [89] and the relationship between average degree and the effect of single node damage [55]. It was shown that the larger the average node degree, the less stable a steady state is against single node damage. Another related result is that random Boolean network ensembles will go through a phase transition from a frozen phase to a chaotic phase as the average node degree increases. Two states that initially differ in a single node's state will diverge on average in the chaotic phase. The critical boundary is average degree $<K> = 2$ when considering unbiased Boolean logic (all Boolean functions) and using an annealed approximation (at every time step the input nodes and Boolean functions are randomized for each node) [26, 90–94]. We note that our setting of a steady state damaged by a single node knockout is different from what was considered in previous work on random Boolean network ensembles.

As biological networks have been observed to exhibit degree heterogeneity and long-tailed decreasing degree distributions [8, 18, 95, 96], we explore the effect of degree heterogeneity on the resilience of a steady state following single node knockout. To probe a variety of regulatory rules consistent with a given number of regulators, we first consider random Boolean rules, and then focus on more realistic nested canalizing Boolean functions.

2.2.1.1 Theoretical estimation of resilience probability in case of single node damage

We define the resilience probability (RP) of a steady state as the probability that the steady state of the network is stable against single node damage. It follows that the damage probability $DP = 1 - RP$. We define $\alpha(I_j)$ to be the probability that a node j with in-degree I_j is stable (does not change state) if one of its randomly chosen inputs, i, is knocked out. Knocking out a node i will directly affect the state of at most O_i nodes, where O_i is the out-degree of node i. If we denote the nodes regulated by i as $n_1, n_2, \cdots, n_{O_i}$, then the probability that the state of the system is a steady state after we knock out node i alone is $p(i) = \prod_{i=1}^{O_i} \alpha(I_{n_i})$ since every regulated node must be stable for the overall network to be stable. The average RP is given by $RP = \frac{1}{N}\sum_{i=1}^{N} p(i)$. Under the mean-field assumption that every node follows the same node in-degree distribution

$f(I_i)$ and out-degree distribution $g(O_i)$, the average *RP* can be estimated as[1]

$$RP = \sum_{O_i} g(O_i)(\sum_{I_j} f(I_j)\alpha(I_j))^{O_i} \tag{2.1}$$

In all cases $\alpha(I_j = 0) = 1$ as a source node cannot, by definition, be disrupted by any other node. If each possible Boolean function occurs with equal chance, $\alpha(I_j) = \frac{1}{2}$ for $I_j > 0$; this is due to the equal probability of having 0 or 1 values in each position of the function, which leads to a chance of one half that a change in value of an input variable does not lead to a change in value of the output.[2] However, to make sure that the regulatory logic correctly reflects the desired topology, we use effective Boolean rules wherein no input is redundant or spurious [98]. That is, for any input node i, $f(\cdots,\sigma_i = 1,\cdots) \neq f(\cdots,\sigma_i = 0,\cdots)$ for at least one pair of input configurations. We find by exhaustive enumeration that for effective rules, the probability $\alpha(I_j)$ changes with the in-degree of the affected node j: $\alpha(I_j = 1) = 0, \alpha(I_j = 2) = 0.4, \alpha(I_j = 3) \approx 0.477, \alpha(I_j = 4) \approx 0.498$. A Monte Carlo calculation shows that α approaches 0.5 as node in-degree increases. Thus one can readily see from the estimated average *RP* (formula 2.1) that $(\sum_{I_j} f(I_j)\alpha(I_j))^{O_i}$ decreases from 1 exponentially as O_i increases from 0. Thus sink nodes and nodes with smaller out-degree have a greater contribution to the resilience probability, as they affect no or few other nodes. Given an average node out-degree, heterogeneity in the out-degree distribution tends to make the steady state of the network more stable against single node damage because it leads to more low-degree nodes. However, since $\alpha(I_j)$ increases relatively slowly and saturates at 0.5 as I_j increases, it is less straightforward to see the dependence between in-degree heterogeneity and the resilience probability of a steady state. We note that our mean-field approximation takes out-degree distribution and effective Boolean rules into consideration compared with annealed approximation.

We also analyze the effect of restricting the Boolean rules to nested canalizing rules, as research shows that the regulation in biological networks is frequently described in this way [99]. A nested canalizing Boolean function with k inputs can be generated by determining two sequences, the input sequence $(I_1, I_2, \cdots, I_k)$ and the output sequence $(O_1, O_2, \cdots, O_k)$, where I_i or O_i is either 0 or 1. The output o as a function of input

[1] To be exact, the in-degree distribution $f(I_i)$ in formula 2.1 should be the conditional in-degree distribution conditioned on a node being knocked out. The conditional in-degree distribution can be obtained through the in-degree distribution reweighted by in-degree.

[2] A similar result was obtained in [97].

configuration $(i_1, \cdots, i_k)$ is thus determined through the hierarchy $o = O_1$ if $i_1 = I_1$; $o = O_2$ if $i_1 \neq I_1$ and $i_2 = I_2$; $\cdots$; $o = O_k$ if $i_1 \neq I_1, \cdots, i_{k-1} \neq I_{k-1}, i_k = I_k$; $o = NOT\ O_k$ if $i_1 \neq I_1, \cdots, i_{k-1} \neq I_{k-1}, i_k \neq I_k$. The last condition is used to guarantee that the rule is an effective rule [99]. All nested canalizing functions can be written in the above form up to a permutation of node order. We determine analytically, and verify by numerical simulations, that the probability that a node′s state will not change after knockout of one of its x regulators is $\alpha(x) = \frac{x-1}{x}$ for nested Boolean functions generated by the method above with no bias in I_i or O_i. This is because knocking out the first dominant canalizing variable i_1 (the probability of this is $1/x$), will change the input configuration; the output will be changed with probability 1/2, which is the probability that two outputs O_1 and $O_l (l \neq 1)$ of the nested Boolean function hierarchy are different. Knocking out the second dominant canalizing variable changes the output only if $i_1 \neq I_1$ (the probability of this is 1/2), the probability that output is changed is 1/2 as before under the condition $i_1 \neq I_1$, and so on. Also notice that the order of the last two inputs in the hierarchy of the canalizing function does not affect the resilience probability, thus the probability of needing repair is $\frac{1}{x}(\frac{1}{2} + (\frac{1}{2})^2 + \cdots + (\frac{1}{2})^{x-1} + (\frac{1}{2})^{x-1}) = \frac{1}{x}$, and therefore $\alpha(x)$ is $\frac{x-1}{x}$. Notice that two different sequences may give the same rule, for example, for a one-input rule, $(I_1 = 1, O_1 = 1)$ is actually the same as $(I_1 = 0, O_1 = 0)$. Also, nested canalizing function ensembles generated by the input and output sequence with no bias lead to a different degeneracy of the Boolean functions in a Boolean table representation, in which the output of the Boolean function is specified for each possible input configuration. Simulations show that nested canalizing Boolean functions randomly picked from the Boolean table representation with equal probability have a different $\alpha(x)$ function: $\alpha(x = 2) = 0.5, \alpha(x = 3) = 0.625, \alpha(x = 4) = 0.712, \alpha(x = 5) = 0.766$. Regardless of the representation, α is larger for nested canalizing functions compared with random Boolean functions or effective random Boolean functions. This indicates that steady states of networks with nested Boolean functions will have an increased resilience against single node damage [99–101]. Since α is smaller than 1, the conclusion that heterogeneity in the out-degree distribution tends to make the steady state of the network more resilient against single node damage holds for nested canalizing functions.

2.2.1.2 Damage probability in simulations of random network ensembles

To estimate the resilience probability, we consider five random Boolean network ensembles with different in-degree/out-degree distributions, namely (a) constant in-degree and scale-free out-degree distribution (SF_out), (b) constant in-degree and Poissonian out-degree (NK_out), (c) constant in-degree and constant out-degree (NKK), (d) Poissonian in-degree distribution and constant out-degree (NK_in), and (e) scale-free in-degree distribution and constant out-degree (SF_in). The algorithm we used in generating these networks will give scale-free (power-law) degree distribution or Poisson degree distribution in the limit of very large network size. Even for small network sizes, the heterogeneity of these two types of networks is significantly different, e.g. the standard deviation of the first′s is approximately twice the second′s. [3] For each ensemble, we generate 1000 networks with 20 nodes. To make sure that the generated ensemble has the desired topology and degree distribution, we only accept at least weakly connected networks and use effective rules when assigning a Boolean function to each node. We study ensembles with average degree $<K> = 1$,2 and 3, which would be in frozen phase for $<K> = 1$ and chaotic phase for $<K> = 2$ or 3 when considering the annealed approximation, the infinite network size limit and unbiased *effective* Boolean rules [90–94]. Note that knowing the phase is not enough to predict the damage probability. For each network, we find all the steady states. We individually knock out (keep in the OFF state) every node that has the ON state in the steady state. A similar procedure can be followed to consider the constitutive expression (sustained ON state) of nodes that are currently OFF in the attractor; we do not explicitly consider this latter case.

We estimate the resilience probability (RP) and damage probability $(DP = 1 - RP)$ for networks with given topological characteristics by considering all steady states and all possible node knockouts with equal probability in the corresponding network ensemble. Fig. 2.1 summarizes the simulation results for the estimated damage probability. In agreement with the theoretical result, for single node damage, given a fixed node in-degree, heterogeneity in out-degree leads to a smaller damage probability for the steady state (compare SF_out, NK_out and NKK results). In contrast, with node out-degree fixed, heterogeneity in in-degree distribution does not show a general trend and is connectedness dependent: the damage probabilities of the NKK, NK_in and SF_in ensembles are close for $<K> = 2$ or $<K> = 3$. Thus the theoretical analysis (see 2.2.1.1 2^{nd} paragraph) is consistent with the computational result. A quantitative comparison of damage probability estimation by simulations and mean-field theory is shown in Table 2.1 for selected ensembles.

[3] When the average degree equals 2, the standard deviation of the node out-degree of one sample ensemble is 1.336 for Poisson distribution, 2.675 for scale-free distribution. When the average degree equals 3, the standard deviation is 1.588 for Poisson distribution, 3.188 for scale-free distribution.

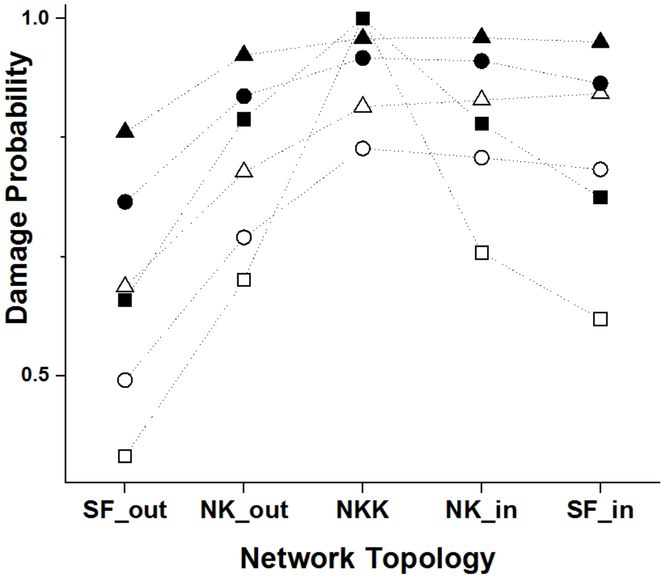


Figure 2.1: The estimated damage probability across five ensembles with different degree heterogeneity. Different symbol shapes of the series represent different average degree, $<K> = 1$ (squares), $<K> = 2$ (circles), and $<K> = 3$ (triangles). Single node knockout results are shown with empty symbols and double node knockout results are shown with solid symbols. The standard error of the average damage probability is estimated to be in the order of 0.001, which is negligible compared with the size of the symbol.

Table 2.1: A quantitative comparison of damage probability estimation

Average degree/Method	SF_out	NK_out	NKK	NK_in	SF_in
$<K> = 2$ Simulation	0.494	0.694	0.818	0.805	0.789
$<K> = 2$ Mean-field	0.496	0.711	0.84	0.842	0.824
$<K> = 3$ Simulation	0.624	0.785	0.877	0.886	0.895
$<K> = 3$ Mean-field	0.609	0.805	0.891	0.905	0.906

Single node damage probability estimation by simulations (1^{st} and 3^{rd} row) and mean-field theory (2^{nd} and 4^{th} row). Different columns correspond to different ensembles. Mean-field calculations employ degree distributions of the generated ensemble.

2.2.2 Double node damage

2.2.2.1 Classification of the resilience scenarios of double node damage

In this section, we investigate the properties of interventions that prevent the cascading effect of knocking out two nodes in a network. Our motivation is not only to generalize the single node damage repair algorithm proposed in Ref. [55], but also to identify the potential combinatorial effects of simultaneous damage to two nodes. This may be related to the observation of genetic interactions in biological systems; specifically, cases where combined knockout of two genes has a stronger or weaker effect than the sum of the effects of the individual knockouts [86]. For example, synthetic lethality and synthetic viability have been studied experimentally [102, 103] and theoretically [104, 105].

When repair is necessary, for each sensitive node, we define candidate nodes as nodes that are neither its pre-existing regulators nor the sensitive node itself, and we add a suitable interaction starting from a candidate node to prevent the state change. (We avoid using pre-existing regulators since it is less biologically feasible [55].) This way, we preserve the steady state aside from the immediate impact on the damaged node and block the cascading effect as soon as possible. Specifically, say node i is regulated by nodes that belong to set A, $x_i = f(x_{j_1}, \cdots, x_{j_k})$, where $j_1, \cdots, j_k \in A$. If one wants to repair node i so that it remains ON ($x_i = 1$), one needs to find a candidate node l (i.e. $l \notin A$ and $l \neq i$) and modify the rule such that $x_i = f(x_{j_1}, \cdots, x_{j_k})$ OR x_l if $x_l = 1$ or such that $x_i = f(x_{j_1}, \cdots, x_{j_k})$ OR $(NOT\ x_l)$ if $x_l = 0$. Here AND, OR and NOT are Boolean functions. Similarly, if one wants to repair node i so that it remains OFF ($x_i = 0$), one can modify the rule to be $x_i = f(x_{j_1}, \cdots, x_{j_k})$ AND $(NOT\ x_l)$ if $x_l = 1$; or $x_i = f(x_{j_1}, \cdots, x_{j_k})$ AND x_l if x_l=0 [55]. Assuming that a candidate node with the appropriate x_l value exists, which is generally the case in realistic networks, regulation of this sort is always possible in principle [55]. We say that a repair solution exists if each sensitive node can be repaired. In the algorithm for double node damage, the sensitive node set is determined after knockout of both nodes; then for each sensitive node, a candidate node set is identified with the additional restriction of excluding both damaged nodes. Then, similarly to single node knockout, an interaction is added from an appropriate candidate node to each sensitive node.

When considering the damage of node A, damage of a different node B, and damage of both nodes, six outcomes are possible, which are summarized in the first six rows of Table 2.2. In order to compare the repair solutions, we denote the sensitive node sets after damage to node A, B, and both A and B as S_A, S_B, and S_{AB}, respectively. If no node is a child node of both node A and node B, $S_{AB} = S_A \cup S_B$. However, if a node is a child node of both nodes A and B, different situations can emerge as indicated in the last four rows of Table 2.2. For completeness, for each class (i.e., situation) we list the possible subclasses in the first column of the table. Classes 1, 2, 3, and 5 admit a single subclass

Table 2.2: Classifications of different situations comparing single node damage and double node damage.

Class/ subclass	Status of SS after single node damage	Status of SS after double node damage
1 (b)	stable for both	stable
2 (c)	stable for both	Needs repair
3 (a)	stable in one case, needs repair in the other case	stable
4 (a,b,c,d)	stable in one case, needs repair in the other case	Needs repair
5 (a)	Both need repair	stable
6 (a,b,c,d)	Both need repair	Needs repair
a	$S_A \cup S_B \supsetneq S_{AB}$	
b	$S_A \cup S_B = S_{AB}$	
c	$S_A \cup S_B \subsetneq S_{AB}$	
d	$(S_A \cup S_B)\backslash S_{AB} \neq \emptyset \quad and \quad S_{AB}\backslash(S_A \cup S_B) \neq \emptyset$	

The first column is the class and its possible subclasses, whose definitions are given in the last four rows. The subclasses are defined based on the relationships between S_{AB} and $S_A \cup S_B$. $S_A \cup S_B \supsetneq S_{AB}$ means S_{AB} is a true subset of $S_A \cup S_B$. $(S_A \cup S_B)\backslash S_{AB} \neq \emptyset$ and $S_{AB}\backslash(S_A \cup S_B) \neq \emptyset$ means that S_{AB} is not a subset of $S_A \cup S_B$ and $S_A \cup S_B$ is also not a subset of S_{AB}, where \ means relative complement. SS means steady state.

only, while classes 4 and 6 can have any of the four subclasses.

In order to gain insight into how different networks lead to the different outcomes of Table 2.2, we consider a simple network motif, in which two nodes (A, B) regulate a third node C. We determine which class and subclass each two-variable Boolean function belongs (Table 2.3). We start from state (1, 1) for the two nodes (in the order A, B). If the output of state (0,1) is different from that of state (1,1), the state needs repair after damage to node A; similar conclusions apply to all the cases. The symmetrical AND rule and its negation belong to class 6, the symmetrical OR rule and its negation belong to class 2, four cases of unsymmetrical two-variable regulation belong to class 3, and the XOR/XNOR functions belong to class 5. The three node motif holds the same properties when embedded within a larger network. However, we emphasize that a network containing a three-node motif and additional nodes does not necessarily fall into the same category as the three-node motif alone, since different parts of the network may be different for different damage situations.

Table 2.3: The ten two-input effective Boolean functions and their classification in double node damage

Function Name	(1,1)	(1,0)	(0,1)	(0,0)	Class
(NOT A) OR (NOT B)	0	1	1	1	6b
(NOT A) OR B	1	0	1	1	3a
A OR (NOT B)	1	1	0	1	3a
A OR B	1	1	1	0	2c
XNOR(A,B)	1	0	0	1	5a
XOR(A,B)	0	1	1	0	5a
A AND B	1	0	0	0	6b
A AND (NOT B)	0	1	0	0	3a
(NOT A) AND B	0	0	1	0	3a
(NOT A) AND (NOT B)	0	0	0	1	2c

The first column indicates the name of the function of A and B, where XOR(A,B)= (A AND (NOT B)) OR ((NOT A) AND B), XNOR(A,B)=NOT XOR(A,B). The second to fifth columns list the value of the respective function for input combinations (A=1, B=1), (A=1, B=0), (A=0, B=1), (A=0, B=0). The sixth column gives the classification (as in Table 2.2) of a network motif composed of three nodes, wherein A and B are source nodes and the regulation of C is described by the respective Boolean function of A and B. This classification assumes that the input nodes are $A = 1, B = 1$ initially. The letter in the last column gives the subclass based on the relationship between $S_A \cup S_B$ and S_{AB} as described in Table 2.2.

We find that in subclass b, that is, $S_A \cup S_B = S_{AB}$, the repair solution for the double node damage will always be a subset of the "direct product" of the single node damage repair solution. More rigorously, let $S_A = \{A_1, \cdots, A_m\}$ and $S_B = \{B_1, \cdots, B_n\}$. A repair solution after knocking out node A has the form $(r_{A_1}, \cdots, r_{A_m})$, where r_{A_1} represents a way to stabilize node A_1. Let R_{A_1} be the set containing all r_{A_1} that appears in all possible repair solutions. The set of all possible solutions after knocking out node A will be denoted as $G_A = \{(r_{A_1}, \cdots, r_{A_m}) : r_{A_i} \in R_{A_i}\}$. For the direct product of single node damage repair solution, $S_D = \{D_1, \cdots, D_p\}$, $p = |S_A \cup S_B|$; $R_{D_i} = R_{A_i} \cup R_{B_i}$ if $D_i \in S_A \cap S_B$ and $D_i = A_i = B_i$; $R_{D_i} = R_{A_i}$ if $D_i \in S_A \cap S_B^c$ and $D_i = A_i$; $R_{D_i} = R_{B_i}$ if $D_i \in S_A^c \cup S_B$ and $D_i = B_i$. Then the direct product of single node damage repair solution is given by $G_D = \{(r_{D_1}, \cdots, r_{D_m}) : r_{D_i} \in R_{D_i}\}$. This can be explained in the following way: since a node only has two states in a Boolean network, it will either be stable or will need repair. When the node needs repairing, damage to an additional node reduces the number

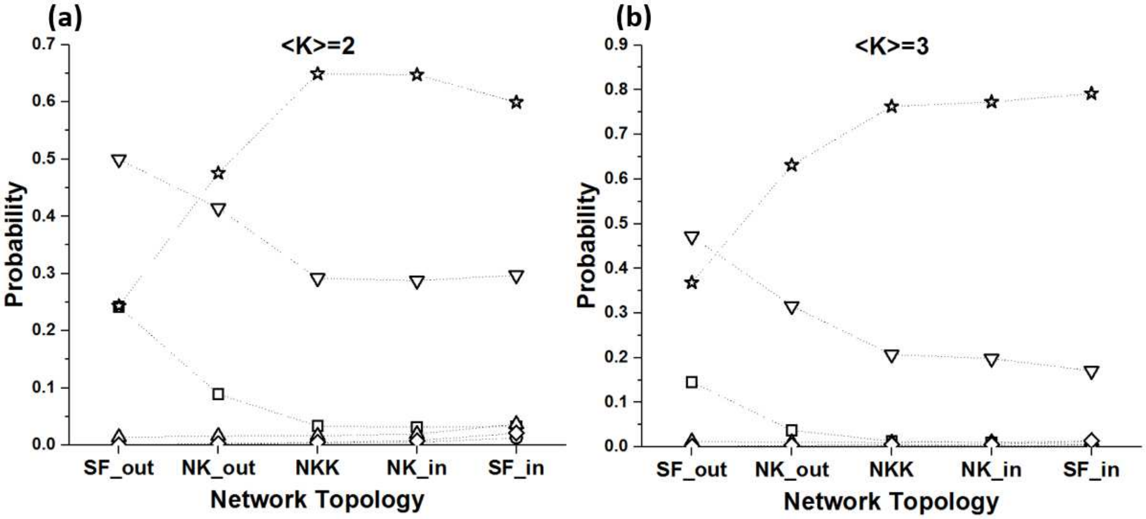


Figure 2.2: Probability of each class of double node knockout across the five ensembles with different degree distributions. The left and right graph shows the result for networks with average degree $<K> = 2$ and $<K> = 3$ respectively. Classes 1 to 6 are drawn in square, circle, up triangle, down triangle, diamond, and star symbols respectively.

of candidate nodes that can be used as starting points of the repair edges; nothing else should happen. Thus, the individual single node repair solutions are compatible with each other. Another observation is that $S_{AB} = S_A = S_B$ can only happen if A and B are regulating the same node(s). Otherwise, part of the damage, and thus also of the repair solutions, would be independent of each other.

2.2.2.2 Damage probability and class distribution in simulations of random network ensembles

Similar to Sec. 2.2.1.2, we study the effect of degree heterogeneity on the resilience probability in a double knockout setting. We also explore the distribution of the repair categories introduced in Sec. 2.2.2.1 using simulations of random Boolean networks. The computational details are similar as in Sec. 2.2.1.2 except we consider all possible pairs of knockouts to obtain the estimation for the damage probability and the classification of each category.

As shown in Fig. 2.1, the damage probability after double knockout is rather high regardless of the degree distribution and is higher than the damage probability after single knockout. As one can see, the double-knockout damage probability is higher in a network with higher average degree, which is consistent with the established conclusion that the complexity of the dynamics increases with larger average node in-degree [90–92]. The NKK model with K=1 is an exception; here the damage probability is 1 whether one or two nodes are damaged. This is because this network forms a single cycle. The only possible effective Boolean rules for K=1 are the identity (the output equals the input) and the negation. Thus knocking out any currently-ON node in the network will induce a change in its child node, which means the network will need to be repaired.

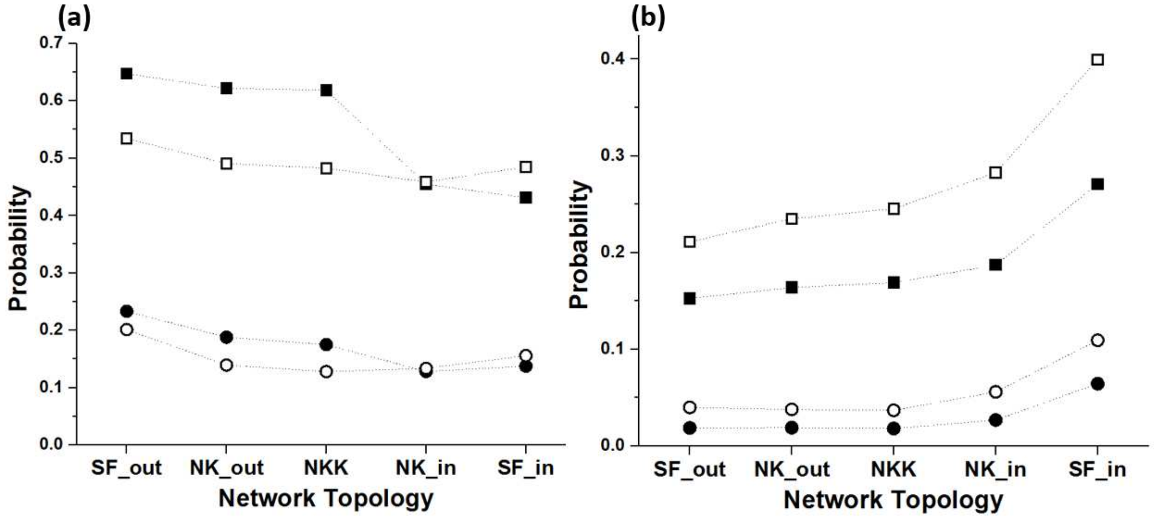


Figure 2.3: The probability that one needs to repair more or different (circles) or fewer (squares) nodes for simultaneous damage of two nodes compared to the union of the repairs needed for individual damage to each of the nodes. The probability that one needs to repair the same nodes is 1 minus the sum of the two shown probabilities. The same network ensembles as in Fig. 2.1 and 2.2 are used. Solid symbols represent $<K> = 2$ results and empty symbols represent $<K> = 3$. (a) Simultaneous damage of two nodes that share a downstream target. (b) The general case of simultaneous damage of two nodes.

Based on the simulations, the damage probabilities of ensembles with fixed out-degree (K=2 and K=3) for double node knockout are rather close to each other; in-degree heterogeneity does not significantly change the damage probability. However, when we compare the three ensembles with fixed in-degree, out-degree heterogeneity leads to a decrease in the damage probability; this is because of the abundance of sink nodes. These results are similar to the results of the single node knockout.

To illustrate the distribution of the double damage classes introduced in Table 2.2, in Fig. 2.2 we plot the probability of each class in the five ensembles. Based on the simulations, class 2 (both single damage cases are stable, repair is needed for double damage), class 3 (repair is needed for one case of single damage, stable after double damage) and class 5 (repair is needed for each single damage, stable after double damage) have very low probability of occurrence. The reason is that the occurrence of these situations requires that the nodes being knocked out are regulating a common target. In contrast, most randomly chosen node pairs are independent. Class 6 (see Table 2.2) tends to have the highest probability, followed by class 4 and class 1 ; the probability of these cases also varies more in the different ensembles. Comparing the three ensembles with a fixed out-degree (K=2 or K=3), the probability of each class is fairly close according to the simulation. Comparing the three ensembles with a fixed node in-degree, we can readily see that heterogeneity in node out-degree leads to a smaller probability for class 6 (stars) and larger probability for classes 4 (down triangles) and 1 (squares). This is related to the fact that heterogeneity in node out-degree leads to more sink nodes in the network.

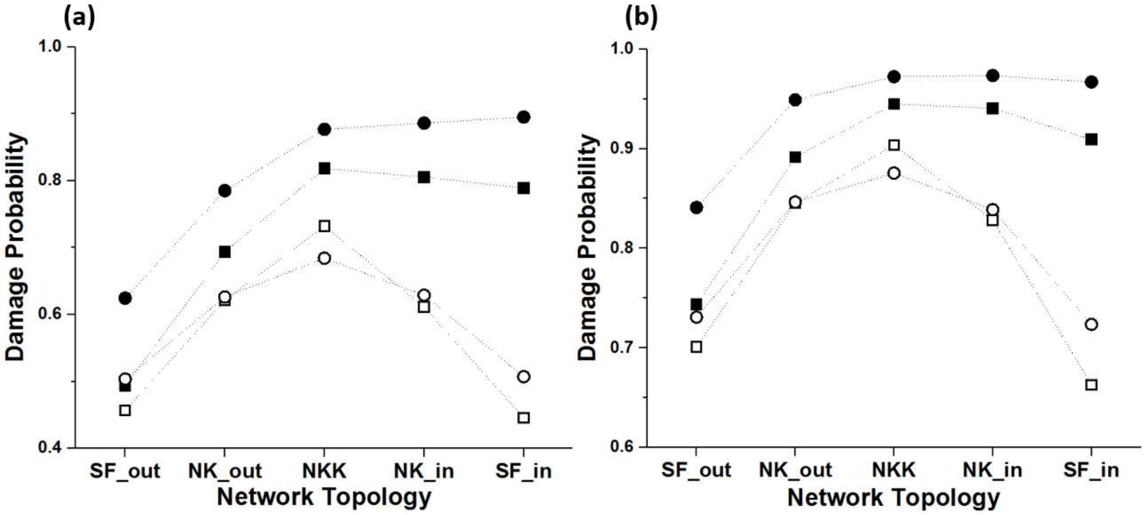


Figure 2.4: Damage probability in network ensembles using effective Boolean rules (solid symbols) or nested canalizing rules (empty symbols). Square symbols represent $<K> = 2$ and circular symbols represent $<K> = 3$. (a) Single node knockout; (b) double node knockout.

As we are interested in combinatorial effects of double node knockout, we marginalize all the (sub)classes into three categories based on whether we need to repair more or different nodes (class 2, 4c, 4d, 6c, and 6d), the exact same set of nodes (class 1, 4b and 6b), or less nodes (classes 3, 5, 4a, and 6a) in case of double damage compared to the union of the two single damage cases. We estimate the probability of each category by simulation using the five ensembles. If the two nodes being knocked out do not share a target, the two damage processes are independent and there will not be any combinatorial effect. It is therefore of particular interest to calculate the probability of each category in just the cases wherein the two nodes share a target (Fig. 2.3(a)), and compare with the general case (Fig. 2.3(b)). Since prior research shows that the average degree of biological networks is around 2, [97,106] we focus on $<K> = 2$ and $<K> = 3$. According to both Fig. 2.3(a) and 2.3(b), the probability that we need to repair fewer nodes (subclass a) in double knockouts is larger than the probability that we need to repair more or different nodes (subclasses c and d). This is consistent with the fact that in Table 2.3, there are more motifs corresponding to subclass a than to subclass c. For $<K> = 2$, compared with networks with constant in-degree (the left three ensembles in Fig. 2.3(a)), networks with constant out-degree and heterogeneous in-degree distribution (the right two ensembles in Fig. 2.3(a)) demonstrate a lower probability for cases wherein one needs to repair more or fewer nodes; for $<K> = 3$, the probabilities are close to each other. As the network topology changes from constant in-degree and scale-free out-degree distribution to constant in-degree and out-degree to constant out-degree and scale-free in-degree distribution (from left to right in Fig. 2.3(b)), the percentage of node pairs sharing a target node among all possible pairs increases. This change is more dramatic than the change in the probability of the three categories across different ensembles in Fig. 2.3(a). Thus as shown in Fig. 2.3(b), the probability of cases wherein one needs to repair more nodes (circles) or less nodes (squares) among all node pairs increases across the five ensembles.

As discussed in Sec. 2.2.1.1, using nested canalizing functions helps make a steady state more resilient to single node damage under the same network topology. This is confirmed by simulation results summarized in Fig. 2.4(a). The damage probability is smaller for networks with nested canalizing functions (empty symbols) for all five ensembles with $<K> = 2$ or $<K> = 3$. This conclusion holds for double node damage, as shown in Figure 2.4(b). As discussed in Sec. 2.2.1.1, out-degree heterogeneity leads to lower damage probability for both effective and canalizing functions (compare the first three ensembles). In the case of nested canalizing Boolean functions, in-degree heterogeneity also leads to a lower damage probability, in contrast with its minor effect in case of effective Boolean functions. This is because higher in-degree leads to more stability for nested canalizing functions, reflected in the fact that the stability probability $\alpha(I_j)$ of nested canalizing functions keeps increasing steadily and is much larger than that of effective Boolean functions for higher I_j (see Sec. 2.2.1.1).

2.2.2.3 Biological Example 1: T-LGL leukemia network

We apply our algorithm to the T cell large granular lymphocyte (T-LGL) leukemia network constructed by Zhang *et al.* [2]. T-LGL leukemia is a rare blood cancer. While normal T cells undergo activation induced cell death (apoptosis) after successfully fighting a virus, leukemic T-LGL cells survive. The network model includes the proteins involved in the activation of T cells, in activation induced cell death, as well as a number of proteins that were observed to be abnormally highly expressed or active in T-LGL cells. The model describes the regulation of each of these proteins with Boolean rules, and captures the normal (apoptosis) and leukemic (survival) states of the system [2]. The original network has 60 nodes, including three source nodes, and 142 regulatory edges. By fixing all the states of source (unregulated) nodes in the biologically relevant condition and iteratively replacing fixed node states in the Boolean rules, one can reduce the network to a smaller network, whose nodes$'$ states are not determined by the source nodes alone but rather by the specific dynamic trajectory of the system [3].

We perform additional network simplification as specified in Appendix A.1. The reduced network model (Fig. 5) has two steady states, a disease cell state and normal cell state. There are five nodes ON in the healthy steady state, and thus there are 10 double knockout cases (see Appendix A.3). Among them, four cases belong to class 4b (see Table 2.2). Six cases belong to class 6: four in 6b and one each in 6c and 6d . Thus in this example, there are no cases where less repair is needed, and there are two cases where combinatorial effect occurs. For the disease steady state, there are 7 nodes in the ON state and thus 21 double knockout cases. Among them one case belong to class 1, ten cases belong to class 4b and ten cases belong to class 6: three in 6a and seven in 6b. Although the T-LGL leukemia network does not belong to any of the five ensembles,[4] this distribution is similar to the consensus results of the ensembles (Fig. 2.2) in that class 6 and class 4 are the most well represented.

2.2.2.4 Biological Example 2: EMT network

Another example we employ to demonstrate our algorithm is the epithelial-to-mesenchymal transition (EMT) network. EMT is a cell fate change involved in embryonic development

[4] The average degree of the network is 1.43, the standard deviation is 0.65 for the in-degree and 0.94 for the out-degree.

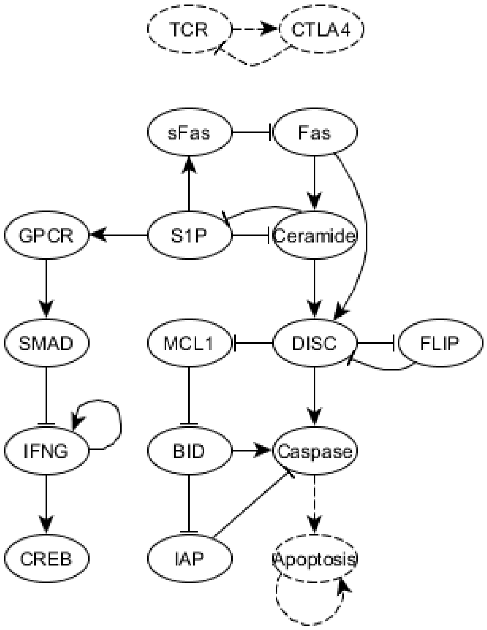


Figure 2.5: Reduced T-LGL leukemia signaling network. An arrowhead or flat bar end indicates a positive or negative regulation edge respectively. The nodes and edges drawn in dashed lines are ignored in the analysis (see text). The reduced network (without TCR, CTLA4 and Apoptosis) thus has two steady states, namely a disease state (0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1) and a healthy state (1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0), where the nodes are in the alphabetic order, BID, CREB, Caspase, Ceramide, DISC, FLIP, Fas, GPCR, IAP, IFNG, MCL1, S1P, SMAD, sFas. The first steady state corresponds to the disease (T-LGL) cell state and the second steady state corresponds to the normal T cell committed to the path to apoptosis.

which can be reactivated during cancer metastasis [107]. During EMT, epithelial cells lose their original adhesive property, leave their primary site, invade neighboring tissue, and migrate to distant site as mesenchymal cell. A Boolean network model of EMT in the context of hepatocellular carcinoma invasion has been established by Steinway *et al.* [5]. The EMT network has 70 nodes and 135 edges. Steinway *et al.* performed a network reduction to obtain a network with 19 nodes and 70 edges (Fig. 2.6). This type of network reduction has been shown to have no effect on the permitted dynamics and enables us to fully explore the state space [5]. In the reduced network, the adhesion factor E-cadherin is the sink node and its OFF state will indicate the transition to a mesenchymal state.

The reduced network has a healthy (epithelial) steady state and a disease (mesenchymal) steady state. For the healthy steady state, there are 6 nodes in the ON state and thus there are 15 double knockout cases. Among them, three cases belong to class 1, nine

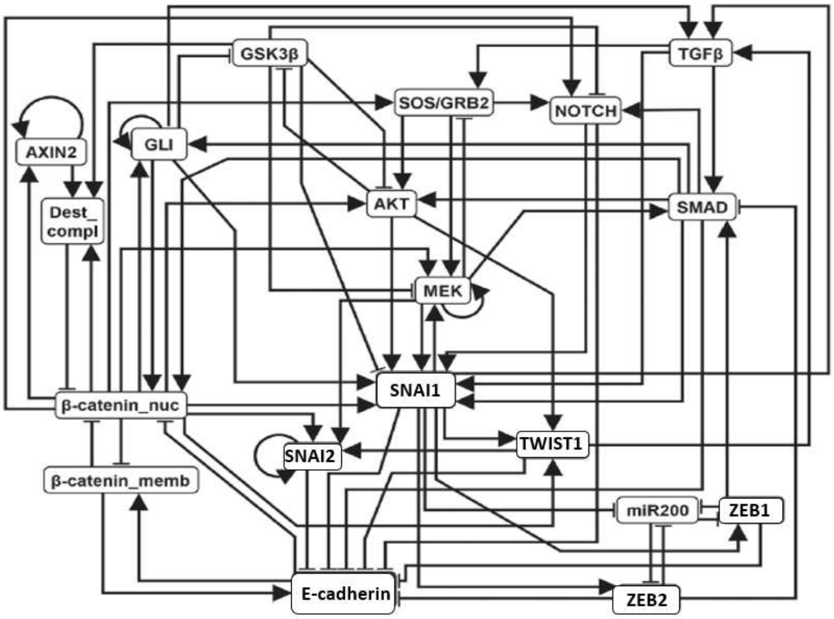


Figure 2.6: Reduced EMT Network. Nodes represent proteins and miRNAs involved. An arrowhead or flat bar end indicates positive or negative regulation, respectively. There are two steady states, epithelial state (0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1) and mesenchymal state (1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0), written in the order of AKT, AXIN2, β-catenin_memb, β-catenin_nuc, Dest_compl, E-cadherin, GLI, GSK3β, MEK, NOTCH, SMAD, SNAI1, SNAI2, SOS/GRB2, TGFβR, TWIST1, ZEB1, ZEB2, miR200.

cases belong to class 4b, two cases belong to class 6b and one case to class 6d. For the disease steady state there are 13 nodes with ON states and thus 78 node pairs. Among them, 44 cases belong to class 1, 1 case belongs to class 2, 2 cases belong to class 3, 28 cases belong to class 4 (one in 4a, twenty-six in 4b, one in 4c), 3 cases belong to class 6 (one in 6a and two in 6b). Thus there are four cases where less repair is needed and two cases where more repair is needed for double knockout compared to the union of two individual single knockouts. Compared with the ensemble average in Fig. 2, class 1 is more represented in the EMT network.[5]

[5] The average degree of the network is 3.68, the standard deviation is 1.80 for the in-degree and 2.38 for the out-degree. Thus the EMT network does not belong to any ensemble in Fig. 2.2.

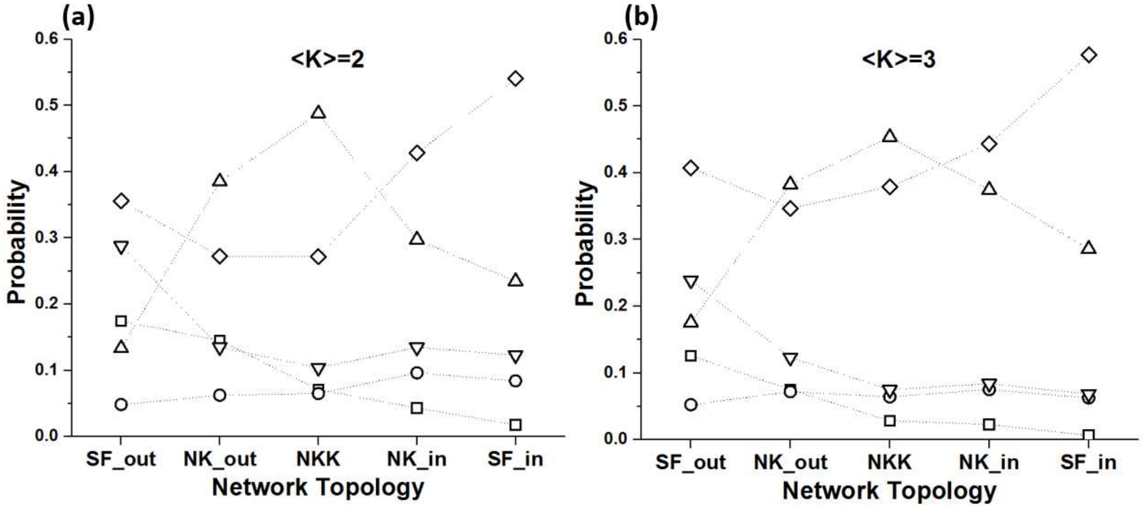


Figure 2.7: Probability of outcomes of node knockout on pairs of steady states across five network ensembles with different degree distributions. The left and right graph shows the result for networks with average degree $<K> = 2$ and $<K> = 3$ respectively. The classes are grouped as 1 (squares), the union of 2 and 3 (circles), the union of 4, 5, 6 (up triangles), 7 (down triangle), and union of 8 and 9 (diamonds).

2.2.3 Single node damage in networks with two steady states

2.2.3.1 General Discussion

Another follow-up direction is to explore the effect of single node damage on two different steady states of a network. The goal is to see whether a single solution can remedy the damage in multiple attractors (steady states here) at the same time. To classify all the situations of knockout damage to a single node, we observe that the damaged node may be normally (when undamaged) ON in both steady states, or ON in one steady state and OFF in the other. (We do not consider the situation that the node is OFF in both steady states, as the knockout damage will not change anything to either steady state). The categorization of the constitutive expression type damage will be analogous.

If the node is ON in both steady states, the steady states can be both stable, both in need of repair or one is stable and the other needs repair. If the node has different states in the two steady states, only one of them needs repair, as summarized in Table 2.4.

We explore the probability distribution of the classification shown in Table 2.4 in random Boolean networks. The computational details are similar to Sec. 2.2.1.2 except we consider all possible single node knockouts for every pair of steady states for a

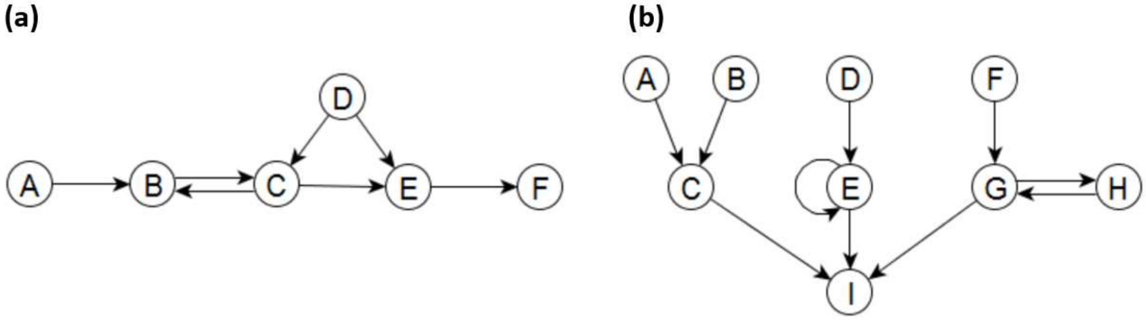


Figure 2.8: (a) A simple network to illustrate the situation when there are no valid repair candidates. All the edges are positive, the updating rules are B = A AND C, C = B OR D, E = C AND D. There are two steady states, (1, 1, 1, 1, 1, 1) and (1, 1, 1, 0, 0, 0). If node A is knocked out, we need to repair node B. All candidate nodes (namely D, E and F) are in different states in the two steady states, thus no common solutions exist. (b) An example network to illustrate incompatibility in stabilizing two steady states at the same time because the two steady states only differ in the state of the knocked-out node and of the sensitive node. C = A OR B, E = D OR E, G = F OR H, I = A OR E OR G. First, we consider the effect of knockout of source node A on the steady state pair (1, 0, 1, 1, 1, 1, 1, 1, 1) and (0, 0, 0, 1, 1, 1, 1, 1, 1). When knocking out node A, we need to repair node C to be ON for the first steady state. However, fixing C to be ON will eliminate the other steady state (where C is OFF) as all the candidate nodes (B, D, E, F, G, H or I) have the same state in the two steady states; thus no compatible repair solutions exist. The incompatibility mechanism is the same for knockout of node E in case of the steady state pair (0, 0, 0, 0, 1, 0, 0, 0, 1) and (0, 0, 0, 0, 0, 0, 0, 0, 0), and the knockout of node G in case of the state pair (1, 1, 1, 1, 1, 0, 1, 1, 1) and (1, 1, 1, 1, 1, 0, 0, 0, 1).

specific network. As there are 9 classes and each class may have a small probability, we marginalize class 2 and 3 (where the node is ON in both steady states before damage and one of the steady states needs repair after damage), class 4, 5 and 6 (where both need repair), class 8 and 9 (where node has different states before damage and one of the steady states needs repair after damage). As shown in Fig. 2.7, we found that the class in which both steady states need repair (up triangle) is less probable in heterogeneous networks. The class in which both steady states are stable (squares) is more probable in out-degree heterogeneous networks as sink nodes contribute to the resilience probability of steady state as in Sec. 2.2.1.

We are particularly interested in determining whether or not there are common repair solutions in the cases where both steady states need repair (classes 4, 5 and 6). We find from our simulations on network ensembles that not having common solutions is less

Table 2.4: Classification of the outcomes of single node knockout for two steady states of a network.The table lists nine situations that will happen after knocking out a single node in the network that has two steady states (SSs). The second column indicates the state of the knocked-out node before damage. The third column specifies whether the damaged steady state is stable against the damage or needs repair. We use the term "common solution" for when we need to repair exactly the same set of nodes for the two steady states after single node knockout, we say "compatible solution" for when there exists a solution that can be used to stabilize both steady states. Thus all common solutions are compatible solutions.

Situation Index	Node state before damage in the two SSs	Status of the two SSs after node damage
1	ON in both	Both stable
2	ON in both	One SS is stable, the other SS needs repair, compatible solutions exist
3	ON in both	One SS is stable, the other SS needs repair, no compatible solutions
4	ON in both	Both SSs need repair, common solution(s) exist
5	ON in both	Both need repair, compatible (but not common) solution(s) exist
6	ON in both	Both need repair, no compatible solutions
7	ON, OFF	Both stable
8	ON, OFF	SS with node ON needs repair, compatible solutions exist
9	ON, OFF	SS with node ON needs repair, no compatible solutions

probable (the fraction of classes 5 and 6 is between 5% and 7% for different ensembles with $<K> = 2$), thus we enumerate these situations. Similar to Sec. 2.2.2.1, we start by looking for three-node motifs that will lead to no common solutions. Reexamining the 10 motifs in Table 2.3, and considering pairs of possible steady states for these motifs, we find that only the XOR/XNOR motif will forbid a common solution for repairing the two steady states. The XOR/XNOR motif is rarely observed in biological networks, as they represent cases where each regulator can switch between being an activator or inhibitor depending on the state of the other regulator.

Another mechanism that will lead to no common solutions for repairing two steady states is that there is no valid candidate to use as a starting point of an additional edge. One such situation is that all the candidate nodes have different states in the two steady states, thus none of them can be used to realize the same function in the two steady states. This is exemplified in Fig. 2.8(a). Another situation is that the sensitive node is regulated by almost every node (other than the node itself) and there are no nodes left to be repair candidates since current regulators cannot be used. A combination of the two situations can also lead to no valid candidate for repair.

When the node is ON in one steady state and OFF in the other steady state, the damage will do nothing to the second steady state. However, the repair solution for the first steady state may or may not be compatible with the second steady state. Our simulations using random networks suggest that the incompatible situation is rarer. Incompatibility can arise in a lot of simple motifs of two or three nodes, including a single regulating edge (positive or negative), OR gate, AND gate, XOR gate, XNOR gate. The reason why this situation is rare in a real network is that if the network has nodes that have different states in the two steady states, any of these nodes can be used as starting points to an additional edge to node B. This additional edge will have an opposite effect in the two steady states and thus it can solve the incompatibility problem. It is rare, but still possible, that two steady states of a network have the same state for most of nodes and only differ in the state of the knocked-out node, the sensitive node, and possibly its current regulators. This can happen if the knocked-out node is part of a bistable motif connected with the rest of the network with a canalizing function such as an OR gate. Thus the bistable motif neither affects nor is affected by the rest of the network in a steady state. Examples of bistable motifs are a source node, a node with a self-loop and a two-node feedback loop (see Fig. 2.8(b)).

2.2.3.2 Biological Example 1:T-LGL leukemia network

We apply our algorithm of stabilizing two steady states simultaneously after a single node damage to the T-LGL leukemia network described in Sec. 2.2.2.3. The reduced network model has two steady states, a disease cell state and normal cell state (apoptosis) [2, 3]. While one generally wishes to eliminate rather than repair a disease state in a biological network, this network nonetheless provides a useful framework for applying our methodology; after considering joint repair, we turn our attention to removing the disease state. As shown in the caption of Fig. 2.5, most of the nodes have opposite states in the two steady states of the reduced network. This is not surprising since the two steady states correspond to two opposite biological outcomes and since the network reduction eliminates nodes that are fixed by source nodes and have the same state in both steady states [3]. The only two nodes having the same state in the reduced network are CREB and IFNG, which exist in a sink branch of the network and do not directly determine the cell state.

Thus when we consider the simultaneous repair of the two steady states, there will be 12 cases wherein the damaged node is ON in one state and OFF in the other. Among them, 4 cases (Caspase, FLIP, IAP, or SMAD knockout) fall into class 7 (see Table 2.4). Directly damaging the node Caspase may be not biologically interesting as we treat this node to be the sink node of the signaling network here. All the other 8 situations (BID, Ceramide, DISC, Fas, GPCR, MCL1, S1P, sFas) fall into class 8. For example, if Fas is knocked out, we need to repair Ceramide to be ON to avert cascading damage to the healthy steady state. The algorithm will give 9 repair solutions involving a new independent edge, shown in Appendix A.2. Two edge repair solutions ("Ceramide= $\cdots$ OR NOT IFNG" and "Ceramide= $\cdots$ OR NOT CREB", where $\cdots$ stands for the original rule for Ceramide) are not compatible with the second steady state since these two nodes have the same node state in the two steady states. This distribution bears similarity with the consensus result for the ensembles (Fig. 2.7) in that class 8 is well represented. However, classes 1-6 do not exist in the T-LGL leukemia network, because the two steady states are almost exactly opposite.

However, in this network, a more biologically meaningful intervention is to keep the normal steady state as intact as possible and destabilize the disease steady state. If the node to be repaired has opposite states in the two steady states, adding a new edge starting from a node that has the same state in the two steady states will destabilize the disease state. In the example of knocking out node Fas, either of the previously mentioned repair strategies: "Ceramide= $\cdots$ OR NOT IFNG" or "Ceramide= $\cdots$ OR NOT CREB", will make the disease state a transient state and the system will keep evolving. Whether the system evolves toward the healthy steady state depends on the node knocked out and the repair solution. In the example above, if we knock out Fas and fix Ceramide to be ON by adding an edge from CREB, this will make the system evolve towards another steady state wherein Caspase is ON, a state biologically similar to the healthy steady state.

2.2.3.3 Biological Example 2: EMT network

Similarly, we apply our algorithm to the EMT network described in Sec. 2.2.2.4. The reduced network has a healthy (epithelial) steady state and a disease (mesenchymal) steady state [5]. One can notice that most of the nodes have different states in the two steady states (shown in the caption of Fig. 2.6) since the two steady states have the opposite biological meaning. We applied our single node damage repair algorithm on this pair of steady states. There are 17 nodes whose knockout can be considered (Dest_compl and SOS/GRB2 are OFF in both steady states). Among them, two cases (AXIN2, SNAI2 knockout) belong to class 1 (see Table 2.4); nine cases (AKT, β-catenin_nuc, GLI, NOTCH, SMAD, TGFβR, TWIST1, ZEB2, miR2000) belong to class 7 ; six cases (β-catenin_memb, E-cadherin, GSK3β, MEK, SNAI1, ZEB1) belong to class 8. As in Sec. 2.2.3.2, this distribution bears similarity with the result for the ensemble average (Fig. 2.7) in that class 8 is well represented, but classes 2-6 do not exist in this network because the two steady states are almost exactly opposite.

As an example, let us consider permanently knocking out node GSK3β, which is ON in the healthy steady state and OFF in the disease steady state. One needs to repair node AKT, MEK, SNAI1, NOTCH and there are 14, 14, 10, 14 simple repair choices for each corresponding node (see Appendix A.4). As most of the nodes have opposite states in the two steady states, the majority of the repair solutions will be compatible with the other steady state. The algorithm then calculates that there will be 11, 11, 6, 11 repair choices for each corresponding node. The specific choices are listed in Appendix A.4.

As in the T-LGL leukemia network, the biologically useful scenario is to preserve the healthy steady state and destabilize the disease steady state after node knockout. Similarly to the T-LGL leukemia network, using nodes with the same node state in the two steady states preserves the healthy steady state and perturbs the disease steady state, however, in some cases the new attractor is not a healthy one (E-cadherin is not guaranteed to be ON).

2.3 Discussion and Conclusion

One promising approach to mitigating the effects of diseases is to proactively manipulate the interactions in the relevant biological network. For example, cancerous cells fail to undergo natural cell death; compensatory interactions in the cancer signaling network may in principle drive cancerous cells to undergo cell death. While a theoretical basis for such manipulation has been established in the case of deregulation of a single node (e.g. a single genetic mutation) [55], complex diseases are triggered by several co-existing gene mutations [86–88]. The algorithm presented here can be used to design preventive interventions for combinations of multiple dysfunctions of the network. Our identified repair strategy classes provide a framework to explore the short-term combinatorial effects of double knockouts and can be straightforwardly adapted to other types of multiple perturbations.

The network ensembles most considered here exist in the chaotic phase for very large networks according to the well-studied annealed approximation (due to the average in-degree of 2 or 3), where the topology and update functions are randomized after each time step [26, 90–92]. Thus, we expect the effects of network perturbations to propagate throughout the network. However, to gain detailed insight into the dynamic behavior of the network and to determine specific repair strategies, it is necessary to consider a fixed network topology and interaction rules. We therefore consider two specific biological case studies in this report.

As patients are often diagnosed with complex diseases after symptoms already developed, the cascading effect of the initial gene mutation or protein dysfunction is already in progress. Thus it is interesting to consider the long-term effects of damage when aiming to repair the effects of single or multiple dysregulations, which is addressed in Chapter 3. One can define a node′s region of influence as the nodes whose states will be changed due to the cascading effect of its perturbation. Similar to what we have done in the short-term setting, if the regions of influence of two nodes do not intersect and are not co-regulating another target, then the two damage processes are independent of each other, and one would expect to be able to mitigate their effects independently. If the regions of influence of two initially damaged nodes intersect or co-regulate a third node, combinatorial effects will appear and can be analyzed in a similar way as we did here.

In some cases two or more steady states with distinct biological meanings, such as natural cell death and cancerous persistence, may exist [86, 108]. As demonstrated in two biological case studies, our algorithm provides strategies to find compatible ways to stabilize two steady states or stabilize one and destabilize the other. The approach we take here is most useful in designing preventive interventions for disease, as the repair is assumed to be effective on a faster timescale than the propagation of damage. Model-based design of therapeutic methods for complex diseases entails an understanding of the disease state and the identification of manipulations that drive the system from the disease state back to a normal state [6]. As a first step, our method provides choices to destabilize the disease state and a framework to test the feasibility of simple edge modifications. A systemic study of the trajectories from a destabilized disease state into a normal state would be another interesting area for future work.

Chapter 3 | Target Control in Logical Models Using the Domain of Influence of Nodes

This Chapter is based on a work submitted to the special issue “Logical Modeling of Cellular Processes” of *Frontiers in Physiology* , where I am the first author and the corresponding author. This chapter was reproduced with permission from Gang Yang, Jorge G. T. Zañudo and Réka Albert.

3.1 Introduction

In cellular systems various molecular species, such as DNA, RNA, proteins and small molecules, interact in diverse ways. The totality of these interactions gives rise to cellular functions. The relationship between molecular interacting systems and cellular functions is studied in the new emerging field of systems biology [7, 8]. A promising systems biology methodology is to represent the molecular interacting system as a network, construct a dynamic model of the information propagation on this network, and identify the cellular functions with long-term behaviors of the dynamic model [8–11]. Various dynamical models of biological networks have been built to integrate related experimental results and to reveal the underlying mechanisms of complex diseases such as cancers. Among various types of dynamical models, logical models, such as Boolean network models, have the advantage of being scalable and not requiring detailed knowledge of kinetic parameters [9, 46]. An abundance of recent literature has shown that logical models can capture the emergent behaviors of real biological systems, they can generate predictions that are validated by follow-up experiments and they can predict successful intervention strategies [4, 109–112].

Network control has recently become a popular research topic as it reflects our interest to not only understand an interacting system, but also intervene in it and modify its outcomes [73, 74]. Network control is a broad subject; different underlying models, different control goals and different possible interventions can be considered [73]. Various control strategies have been designed for both continuous dynamical systems [24, 77, 87, 113–115] and discrete ones [6, 80, 116, 117]. In electric circuits modeled by a system of linear ordinary differential equations, it is possible to obtain full control of the system, that is, to drive the system to any state from any initial condition [24, 76]. For non-linear systems, attractor control, that is to drive the system to one of its natural attractors from any initial condition, has been achieved in several modeling frameworks, such as feedback vertex control for ordinary differential equation models [77] and stable motif control for logic (Boolean) models [6]. However, in biological systems it is not necessary and often not practical to control every component of the system. A more realistic problem is target control, where we assume that the state of the system is mostly determined through a subset of nodes and the control goal is to drive these nodes into desired states. The target control problem has been considered for continuous models by [78]; here we provide a framework to solve the target control problem in Boolean network models.

Despite recent progress in molecular biology, quantitatively manipulating the level of a chemical species is still a challenging problem for experimentalists. Thus any control strategy involving applying time-dependent, variable signals to a target is hard to implement in real systems. However, gene knockout, pharmacological inhibition of proteins and providing sustained external signals have been robustly implemented and demonstrated to be effective intervention strategies [118]. Thus we choose our intervention options to be maintaining a sustained state (either absence or abundant activity) in order to make the solution more practical. To describe the long-term effect of such a sustained state, we define a node property called domain of influence for each node. This is essentially asking which other nodes will adopt a fixed state due to the sustained intervention regardless of initial conditions. Then it follows that the solution to the target control problem will be the set(s) of nodes whose domain of influence can cover the desired target node state combinations.

In the following, we briefly repeat the Boolean modeling framework and relevant previously-developed concepts such as the expanded network and stable motif. Then we define the domain of influence (DOI) and logical domain of influence (LDOI) of a node or multiple nodes and present several useful properties. We then define our target control problem and describe our target control strategy based on DOI using greedy randomized adaptive search in state space. We finally illustrate the effectiveness of our target control strategy in random ensembles and real biological network models.

3.2 Materials and Methods

3.2.1 Boolean network models of biological systems

A dynamical model of a biological system starts with the construction of a network consisting of nodes that represent the system′s elements and edges that specify the pairwise relationships between nodes. In biological networks at the molecular level, nodes are molecular species such as small molecules, RNA, protein, and edges indicate interactions and regulatory relationships. In discrete models, each node i is characterized by a discrete state variable σ_i, and the vector $(\sigma_1, \cdots, \sigma_n)$ represents the state of the system [4]. The state of the system can be followed in continuous time or at discrete time intervals. In discrete time models, each node state is updated at discrete time intervals. Mathematically, the activity of each node σ_i is described by a regulatory rule $\sigma_i(t+\tau_i) = f_i(\sigma_{i_1}(t), \cdots, \sigma_{i_k}(t))$, where $i_1, \cdots, i_k$ are the regulating nodes of i and τ_i is a discrete time delay [4]. The regulatory functions f cannot be constant functions (i.e. cannot yield the same output regardless of the state of the regulators). In models describing signal transduction networks the external signals are represented with source nodes whose regulatory functions depend only on their own state, usually sustaining this state: $\sigma_i(t+\tau_i) = \sigma_i(t)$ [4].

Here we focus on discrete time Boolean network models, where node states are binary, 1(ON) or 0(OFF), and the regulatory function is specified by a truth table or using the Boolean operators AND, OR, NOT [4, 119, 120]. This is motivated by the fact that biological species are frequently observed to demonstrate switch-like behaviors and have highly nonlinear regulations; thus the node state 1 means the molecular species is above a threshold concentration or activity and thus it is able to regulate its targets and the node state 0 means it is below a threshold concentration or activity and is thus ineffective [9, 53]. Depending on the updating scheme, the time trajectory of the system is simulated deterministically or stochastically. A simple deterministic updating scheme is synchronous updating, where $\tau_i = 1$ for every node [9]. In this scheme, the system will deterministically evolve from a specific initial state into an attractor, which can be a steady state (fixed point) or a limit cycle, which consists of several states that repeat regularly. Steady states can be interpreted as cell types; limit cycles may correspond to a cell cycle or circadian rhythms [4]. In general asynchronous updating, a commonly used stochastic updating scheme, a random node is selected to be updated at each time step [121]. This type of update is motivated by the fact that different biological processes have various timescales, and often the timescales of specific processes are not known [122]. While limit cycles depend on the specific chosen updating regime, fixed points (steady states) do not depend on the updating scheme [51]. Stochastic update may lead to attractors that involve irregular repetition of a set of states, called complex attractors.

3.2.2 Previously established concepts in Boolean networks

The expanded network integrates the original network with the regulatory rules of each node [70]. We illustrate the expanded network with the example in Fig. 3.1 , which consists of five nodes, node 0, 1, 2, 3 and 4 with the regulatory functions $f_0 = \text{NOT } \sigma_3, f_1 = (\text{NOT } \sigma_0) \text{ OR } \sigma_3, f_2 = \text{NOT } \sigma_1, f_3 = (\text{NOT } \sigma_2) \text{ OR } (\text{NOT } \sigma_4), f_4 = \sigma_0 \text{ OR } \sigma_1$. First, we denote each original node i by n_i in the expanded network, and we introduce a complementary node for each original node in the system to represent the negation (deactivation) of the original node, denoted by $\sim n_i$ [70]. As the NOT function is a unary operator, all the NOT functions are replaced by the negated state of the respective node (*i.e.* its complementary node) in each Boolean regulatory function. The edges in the expanded network are revised according to the updated rules so that every edge represents a positive regulatory relationship in the expanded network. For example, $f_0 = \text{ NOT } \sigma_3$ implies the rule for the original node n_0 as $f_{n_0} = \text{ NOT } n_3 \ = \sim n_3$, and thus a corresponding edge is drawn from $\sim n_3$ to n_0 in the expanded network. The Boolean regulatory function for the complementary (negated) node is the logical negation of the regulatory function of the original node. In this example, $f_{\sim n_0} = \text{ NOT } (\text{NOT } n_3) = n_3$ and thus a corresponding edge is drawn from n_3 to $\sim n_0$ in the expanded network.

Second, to differentiate OR rules from AND rules when multiple edges point toward the same target node, we introduce a composite node for each set of edges that are linked by an AND function [70]. In order to uniquely determine the edges of the expanded

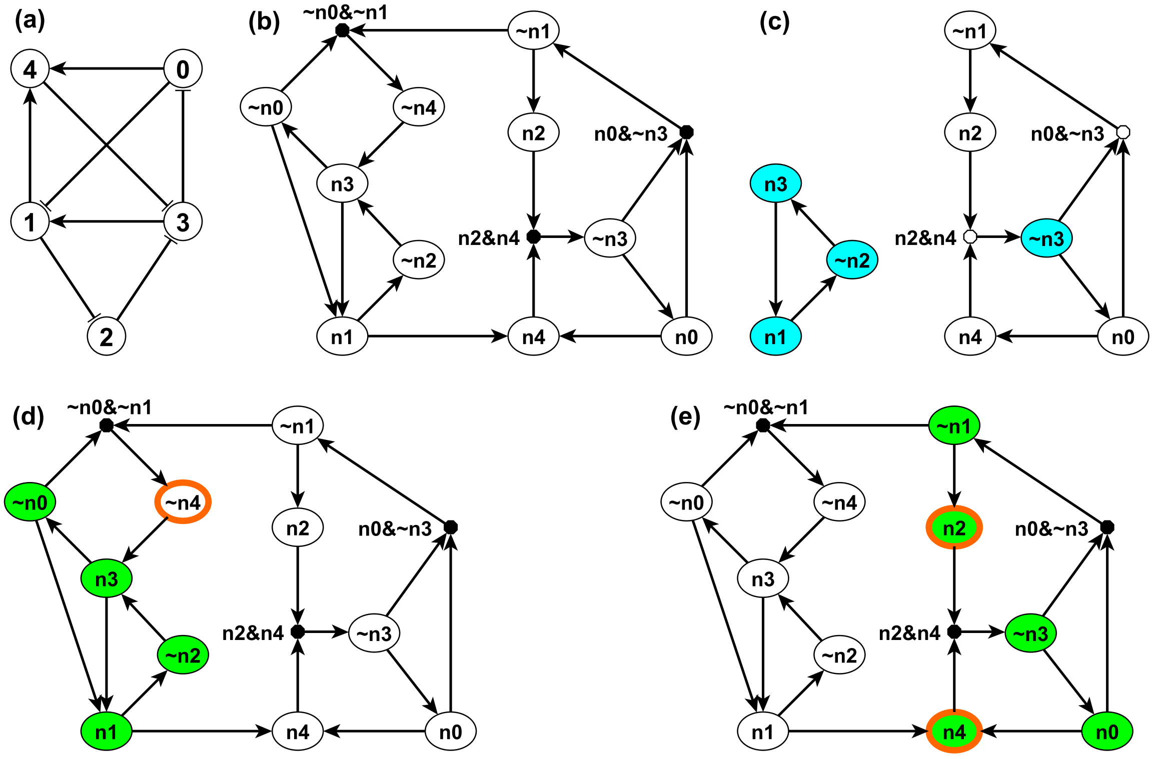


Figure 3.1: An example network, its corresponding expanded network and its stable motifs are shown in sub-figures (a), (b) and (c) respectively. The LDOI of $\{\sim n_4\}$ and $\{n_2, n_4\}$ illustrated on the expanded network are shown in sub-figures (d) and (e) respectively. In panel (a) each edge with an arrow represents activation and each edge with a flat bar represents inhibition. Each node i in panel (a) has a correspondent n_i and its complementary node $\sim n_i$ in panel (b). (Note that n_i is labeled as *ni* in panel (b) to be more visible). A composite node is drawn as a filled black circle and & represents the AND logic operator. In panel (c), each blue node is a single-node core of the corresponding stable motif. In panel (d) and (e), nodes with thick orange boundary are the sustained interventions and the green nodes are their LDOI.

network, the regulatory functions need to be specified in disjunctive normal form, that is, a disjunction of conjunctive clauses (in other words, grouped AND clauses separated by OR clauses). For example, (A AND B) OR (A AND C) is in a disjunctive normal form, while A AND (B OR C) is not. The desired disjunctive normal form can be formed by a disjunction of all conditions that give output 1 in the Boolean table and then simplified to the disjunction of prime implicants (Blake canonical form) by the Quine-McCluskey algorithm [72]. Now we add a composite node for each AND clause in the Boolean regulatory function, denoted by a filled black circle in Fig. 3.1 (b). For example, the composite node $\sim n_0 \& \sim n_1$ in the left upper part of Fig. 3.1 (b) represents the expression (NOT n_0) AND (NOT n_1), which implies the complementary node $\sim n_4$. Notice that one can read all the regulatory functions from the topology of the expanded network. The AND rule is indicated by a composite node with multiple regulators, while all the other edges represent independent activation (parts of an OR function).

As the expanded network encapsulates the regulatory logic that determines the network dynamics, it can serve as a basis for attractor analysis. One approach is through analyzing the stable motifs of the expanded network [54]. A stable motif is defined as the smallest strongly connected component (SCC) satisfying the following two properties: 1) The SCC cannot contain both a node and its complementary node and 2) If the SCC contains a composite node, it must also contain all of its input nodes [54]. The first requirement guarantees that the SCC does not contain any conflict in node states and the second requirement guarantees that all the conditional dependence is satisfied and the SCC is self-sufficient in maintaining each node state inside the stable motif. Thus the stable motif represents a group of nodes that can sustain their states irrespective of outside nodes$'$ states. The corresponding node states implied by the stable motif can be directly read out: an original node represents the ON (1) state and a complementary node represents the OFF (0) state [54]. For example, in the left part of Fig. 3.1 (c), node $n_1, \sim n_2$ and n_3 form a stable motif, representing that node 1 and node 3 are ON and node 2 is OFF. There is a strong correspondence between stable motifs and the attractors of the system. Specifically, there is a one-to-one correspondence between a sequence of stable motifs and a fixed point or a partial fixed point (a part of a complex attractor). A partial fixed point is defined as a true subset of all the nodes whose respective state remains unchanged after being updated regardless of the states of the nodes excluded from this subset [54].

3.2.3 The domain of influence of a sustained node state

We define the domain of influence (DOI) of an intervention that maintains a sustained node state as all the node states that will be stabilized in the long term under the influence of this intervention for all initial conditions in any updating regime. Mathematically, $\mathscr{D}(\sigma_i = \tilde{\sigma}_i) = \{\sigma_j = \tilde{\sigma}_j : \sigma_j(t) = \tilde{\sigma}_j \; as \, t \to \infty \; for \; any \; (\sigma_1(t=0), ..., \sigma_k(t=0)) \; when \, \sigma_i(t) = \tilde{\sigma}_i \; for \, any \, t > 0\}$, where $\sigma_i(t) = \tilde{\sigma}_i$ is the intervention, $\tilde{\sigma}_i = 0$ represents knockout and $\tilde{\sigma}_i = 1$ represents sustained activation, $\tilde{\sigma}_j$ represents a node state fixed by the intervention, and $(\sigma_1(t=0),...,\sigma_k(t=0))$ represents the initial condition of all the nodes of the system. We do not include the intervention node state $\sigma_i = \tilde{\sigma}_i$ in its own DOI, unless the node is sufficient to maintain the corresponding node state in the long term even in the absence of a sustained intervention. Notice that there is one-to-one correspondence between a node state $\sigma_i = \tilde{\sigma}_i$ and a non-composite node n^{ex} in the expanded network : $\sigma_i = 1$ corresponds to a normal node n_i in the expanded network and $\sigma_i = 0$ corresponds to a negation node $\sim n_i$. Thus we use the two notations interchangeably, that is, $\sigma_j = 1 \in \mathscr{D}(\sigma_i = 1)$ is equivalent to $n_j \in \mathscr{D}(n_i)$ and $\sigma_j = 0 \in \mathscr{D}(\sigma_i = 0)$ is equivalent to $\sim n_j \in \mathscr{D}(\sim n_i)$.

The DOI of a node is difficult to calculate because it entails determining the common part of all attractors of a dynamical system to identify the nodes whose states stabilize due to the considered intervention. As an alternative to this computationally hard problem, we define a related concept called the logic domain of influence (LDOI) of an intervention that maintains a sustained node state. The logic domain of influence consists of all the node states that, for any initial condition, are stabilized by the first update of the corresponding node in an updating regime that preserves the level order (breadth first search order) of the expanded network. An updating regime preserves the level order if all the nodes in the nth level are updated at least once before updating any node in the $(n+1)$th level (see details in Appendixl). We denote the LDOI of a node state σ_i as $\mathscr{L}\mathscr{D}(\sigma_i = \tilde{\sigma}_i)$. We define the LDOI of an empty set to be an empty set, $\mathscr{L}\mathscr{D}() = .$ This is consistent with the definition as an updating order preserving the level order starting from a null set can start from any node, and a node will not be stabilized to a fixed state upon its very first update for all initial conditions unless its regulatory function is a constant. Source nodes stabilize in their initial state, which nevertheless will be different for different initial conditions.

3.2.4 Determining the logical domain of influence of a sustained node state

We propose to find the LDOI of a node state by doing a modified breadth first search (BFS) on the expanded network (see the pseudocode in Appendix). In order to find the LDOI of $\sigma_i = \tilde{\sigma}_i$, we start the search from the corresponding node (or complementary node) on the expanded network. If we meet another non-composite node, we add this node to the LDOI; if we meet a composite node, we add this composite node only if all of its parent nodes (i.e. regulators) are already part of the LDOI. This is due to the fact that any edge from a node to a non-composite node represents a sufficient relationship and any edge from a node to a composite node represents a necessary relationship. We keep searching on the expanded network until no new nodes can be added to the LDOI. For example, in Fig. 3.1 (b), one can readily see that $\mathscr{LD}(\sigma_1 = 1) \equiv \mathscr{LD}(n_1) = \{n_4, \sim n_2, n_3, n_1, \sim n_0\}$ following the described search procedure. The first difference from a normal BFS to find a connected component starting from a node is that we put an extra rule for including a composite node. Another subtle difference is that we do not include the starting point unless we visit this starting point again in our search process.

During the search process, there is a possibility that we meet the negation of the starting point. This reflects the possibility that a node state can indirectly lead to the opposite state through a negative feedback loop. This outcome represents a conflict with the original intervention. We do not add this node to the LDOI because we assume that the intervention can sustain the original node state, thus the opposite state is not reachable. This truncation of the LDOI to avoid including the negation of the starting node state ensures that the LDOI will not contain a node which is the negation of an already visited node. Mathematically, if a non-composite node $n_i^{ex} \in \mathscr{LD}(n_j^{ex})$, then n_j^{ex} is sufficient to activate n_i^{ex} , i.e., the long-term logical rule for n_i^{ex} can be expressed in the form $n_i^{ex} = n_j^{ex}$ OR $\cdots$; this implies $\sim n_i^{ex} = \sim n_j^{ex}$ AND $\cdots$, *i.e.*, $\sim n_j^{ex}$ is necessary to activate $\sim n_i^{ex}$. Thus any conflict between n_i^{ex} and $\sim n_i^{ex}$ will occur after the conflict between n_j^{ex} and $\sim n_j^{ex}$ during the search process. This truncation of the LDOI is the third difference compared with a normal BFS.

For example, in the network of Fig. 3.1 (d), the LDOI of the complementary node $\sim n_4$ includes nodes $n_3, \sim n_0, n_1, \sim n_2$ following the search procedure. From n_1 one can also reach node n_4, which is the negation of the considered intervention. Thus we stopped this branch of searching based on our truncation rule. Since there are no more nodes that can be added, we conclude that $\mathscr{LD}(\sim n_4) = \{n_3, \sim n_0, n_1, \sim n_2\}$.

Our LDOI search procedure is equivalent to doing a simulation on the expanded network. If we update the system corresponding to the BFS order of the expanded network starting from the intervention node (i.e., we update node i if we visited n_i^{ex} on the expanded network), all the updated nodes are guaranteed to stabilize in the corresponding visited state on the expanded network, *i.e.* as in the logic domain of influence (LDOI) of that node. In the example of Fig. 3.1, as discussed above, $\mathscr{LD}(n_1) = \{n_4, \sim n_2, n_3, n_1, \sim n_0\}$. If we update the nodes in the order 4, 2, 3, 1, 0, each node will stabilize in the state as in $\mathscr{L}\mathscr{D}(n_1)$. We note that this does not put a restriction on the updating regime: if we update the system in an arbitrary order, each node in the LDOI of the given sustained intervention will stabilize in the first update after all of its regulators included in the LDOI have been updated once. For example, if we fixed the node 1 to be ON and we perform rounds of update of the nodes in the order 0, 1, 2, 3, 4, nodes 2, 3 and 4 will be stabilized in the first round of updating, while nodes 0 and 1 will be stabilized in the second round.

The difference between the LDOI and DOI is that LDOI requires the nodes to be stabilized when being updated for the first time, while DOI just requires the nodes to be stabilized in finite time. Thus one can see that the LDOI of a node will be a subset of the DOI of a node. In many cases the two concepts give the same result. Two exceptions are illustrated in Fig. 3.2. In both cases the DOI of an intervention contains more nodes than the LDOI of this intervention. This is because certain nodes may stabilize not because of the influence of the intervention but because of the collective effect of two inconsistent feedback loops or because of a stable motif stabilized by an oscillation. In the network of Fig. 3.2 (a), the three regulators of node B are independent and the network includes both a positive and a negative feedback loop. To analyze the LDOI of $A = 1$, taking the feedback effect of C and D on B into consideration, the regulatory function of B is simplified into $\sigma_B(t+\tau_B) = \sigma_B(t-\tau_C)$ OR NOT $(\sigma_B(t-\tau_D))$, which yields a constant state $\sigma_B = 1$. Thus $\mathscr{D}(A) = \{B, C, \sim D\}$, as the stabilization of B leads to the stabilization of C and D as well. However, $\mathscr{L}\mathscr{D}(A) =$ as the activation of the composite node requires nodes $A, \sim C, \sim D$ on the expanded network shown in Fig. 3.2 (b) and thus we cannot add the composite node to the LDOI of node A. In the example shown in Fig. 3.2 (c), the two regulators are independent for node B, $\mathscr{D}(C) = \{B\}$ as the negative feedback loop of node A will make A oscillate, but B will stabilize into the ON state after the first time that A visits the ON state and activates B, while $\mathscr{L}\mathscr{D}(C) =$ for the same reason as in the last example.

3.2.5 Properties of the logical domain of influence of a sustained node state

In order to further illustrate the concept of LDOI, we discuss a few of its properties and its relationship with established concepts in Boolean dynamics.

A natural question to ask is about the possible inclusion relationship between the

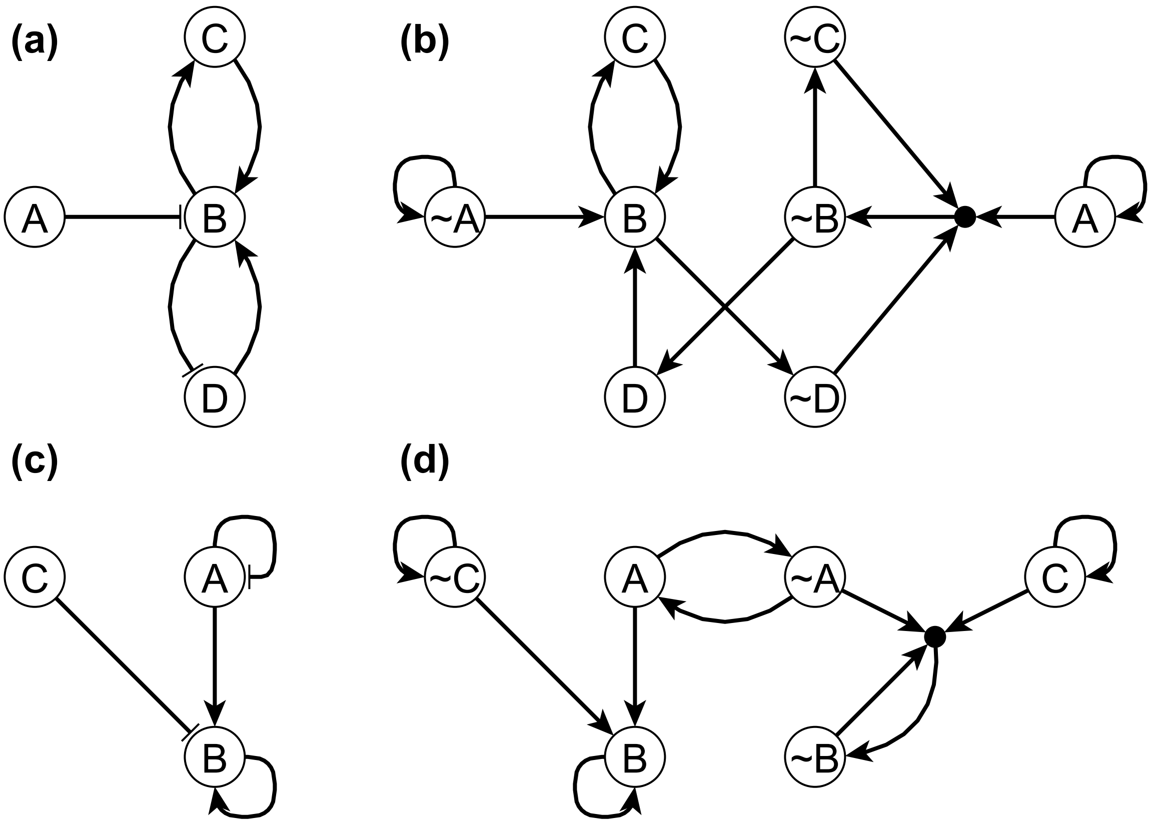


Figure 3.2: Two example networks (panel (a) and (c)) and their respective expanded networks (panel (b) and (d)) that illustrate the difference between DOI and LDOI. In both networks, an edge with an arrowhead represents activation while an edge with flat bar represents inhibition. Implicit positive self-loops for source nodes are not shown in panel (a) and (c). In panel (a) the regulatory functions are $f_A = A$, $f_B = (\text{NOT } A) \text{ OR } C \text{ OR } D$, $f_C = B$, $f_D = \text{NOT } B$. When A=0 the system has a single attractor, the fixed point is $\sigma_B = \sigma_C = 1, \sigma_D = 0$. In panel (c) the regulatory functions are $f_A = \text{NOT } A$, $f_B = A \text{ OR } B \text{ OR NOT } C$, $f_C = C$. When $\sigma_C = 1$ the system has a complex attractor in which A oscillates and $\sigma_B = 1$.

logic domains of influence of two node states $\sigma_i = \tilde{\sigma}_i$ and $\sigma_j = \tilde{\sigma}_j$ in the case when $\sigma_j = \tilde{\sigma}_j \in \mathscr{LD}(\sigma_i = \tilde{\sigma}_i)$ or $n_j^{ex} \in \mathscr{LD}(n_i^{ex})$ in the expanded network notation, where n_i^{ex} and n_j^{ex} represent any non-composite node in the expanded network. In a directed graph, if node n_j is a reachable from node n_i, all descendants of n_j will also be reachable from n_i; indeed one can easily prove this by contradiction. However, due to the special properties of the expanded network and the truncation of the LDOI, this inclusion relationship $\mathscr{LD}(n_j^{ex}) \subseteq \mathscr{LD}(n_i^{ex})$ is not generally true for the expanded network. It is possible that $n_j^{ex} \in \mathscr{L}\mathscr{D}(n_i^{ex})$, however, $\sim n_i^{ex} \in \mathscr{L}\mathscr{D}(n_j^{ex})$. In this case, by definition of the logic domain of influence, we won't allow the negation of a node state to be part of the logic domain of influence of a node state. For example, $n_1 \in \mathscr{L}\mathscr{D}(\sim n_4)$, however, $n_4 \in \mathscr{L}\mathscr{D}(n_1)$. Thus $\mathscr{L}\mathscr{D}(n_1) \not\subset \mathscr{L}\mathscr{D}(\sim n_4)$.

If we add an additional restriction on the two nodes, this inclusion relationship will hold the same way as for descendants in a directed graph. To be specific, the *first key property of the LDOI* is, if the node state $\sigma_i = \tilde{\sigma}_i$ and $\sigma_j = \tilde{\sigma}_j$, corresponding to the two non-composite node n_i^{ex} and n_j^{ex} on the expanded network, are both included in the same (partial) fixed point and $n_j^{ex} \in \mathscr{L}\mathscr{D}(n_i^{ex})$, the logic domain of influence of n_j^{ex} will be a subset of the logic domain of influence of n_i^{ex}, i.e. $\mathscr{L}\mathscr{D}(n_j^{ex}) \subseteq \mathscr{L}\mathscr{D}(n_i^{ex})$. (Recall that a partial fixed point is a subset of nodes whose respective state remains unchanged after being updated regardless of the states of the nodes excluded from this subset.) The reason why the inclusion relationship holds is that node states in a (partial) fixed point stabilize in the long term, thus they will not lead to a situation with opposing behavior $n_j^{ex} \in \mathscr{L}\mathscr{D}(n_i^{ex})$ and $\sim n_i^{ex} \in \mathscr{L}\mathscr{D}(n_j^{ex})$. This restriction can be weakened to only require that node state n_i^{ex} is in a (partial) fixed point. The reason is that if $n_j^{ex} \in \mathscr{L}\mathscr{D}(n_i^{ex})$ and n_i^{ex} is in a (partial) fixed point, then n_j^{ex} must also be in the same (partial) fixed point, or be a node whose state stabilizes due to the nodes in the partial fixed point. Also, as one or more stable motifs are part of a (partial) fixed point, the conclusion will be true if one replaces âĂIJ(partial) fixed pointâĂİ by âĂIJstable motifâĂİ in the above statement. For example, as nodes n_1, $\sim n_2$ and n_3 form a stable motif and its corresponding (partial) fixed point is $(\sigma_1, \sigma_2, \sigma_3) = (1,0,1)$, which also lead to the stabilization of the remaining two nodes as $\sigma_0 = 0$ and $\sigma_4 = 1$, thus $n_3 \in \mathscr{L}\mathscr{D}(n_1)$ implies that $\mathscr{L}\mathscr{D}(n_3) \subseteq \mathscr{L}\mathscr{D}(n_1)$. In fact, $\mathscr{L}\mathscr{D}(n_3) = \mathscr{L}\mathscr{D}(n_1) = \{n_4, \sim n_2, n_3, n_1, \sim n_0\}$. Also $n_4 \in \mathscr{L}\mathscr{D}(n_1)$ implies that $\mathscr{L}\mathscr{D}(n_4) \subseteq \mathscr{L}\mathscr{D}(n_1)$. Note that only n_1 is part of the stable motif or partial fixed point in the latter example, n_4 is not.

As stable motifs represent generalized positive feedback loops of the Boolean network [54], we explore the relationship between stable motifs and the logic domain of influence of a node state. The *second key property of LDOI* is, if the logic domain of influence of a node state contains this node state itself, the logic domain of influence contains a stable motif. As the LDOI of a node state only contains the node state itself if we meet this node during the search process on the expanded network, this indicates the existence of a positive feedback loop, which is the intuition why this proposition holds. (A sketch of proof from the dynamical standpoint is included in Appendix .) For example,

$n_1 \in \mathscr{L}\mathscr{D}(n_1)$ implies that there exists a stable motif contained in $\mathscr{L}\mathscr{D}(n_1)$, indeed, $SM_1 = \{n_1, {\sim}n_2, n_3\} \subseteq \mathscr{L}\mathscr{D}(n_1)$.

3.2.6 The domain of influence of a node state set

Now we generalize the concept of DOI of a single node state to DOI of a node state set (i.e., a set of nodes, each in a sustained state). We define the DOI of a node state set as all the node states that can be stabilized in the long term by the given set of node states under all initial conditions in any updating regime. Mathematically, $\mathscr{D}(\{\sigma_i = \tilde{\sigma}_i\}) = \{\{\sigma_j = \tilde{\sigma}_j\} : \sigma_j(t) = \tilde{\sigma}_j \, as \, t \to \infty \, for \, any \, (\sigma_1(t=0), ..., \sigma_k(t=0)) \, when \, \sigma_i(t) = \tilde{\sigma}_i \, for \, any \, t > 0\}$, where $\{\sigma_i(t) = \tilde{\sigma}_i\}$ represents the intervention consisting of a specific set of node states. Note that the following two notations are equivalent: $\mathscr{D}(\{\sigma_i = \tilde{\sigma}_i\}) \equiv \mathscr{D}(\{n_i^{ex}\})$. Similarly, we define the logic domain of influence of a node state set, $\mathscr{L}\mathscr{D}(\{\sigma_i = \tilde{\sigma}_i\})$, as all the nodes that can be stabilized by the first update in any BFS order-preserving (on the expanded network) update order starting from this given set of node states under all initial conditions. As in the single node state case, the LDOI of a node state set will be a subset of the DOI of the same node state set.

The LDOI of a node state set can be determined by a modified BFS on the expanded network, now using multiple starting points. This does not add complexity to the iterative implementation of BFS: we just need to initialize the queue with the set of given node states. Similar to the case of finding the LDOI of a single node state, we need to deal with the conflicts that may occur during the search process. To be precise, conflict means that during the search we visit a node state that is the negation of a node state included in the intervention. We call such intervention set an incompatible set. The incompatible situation can arise in the following two scenarios. First, a node state in the given set may have a LDOI that was truncated to avoid containing its own negation. The second scenario is when the LDOI of two node states n_i^{ex} and n_j^{ex} have the property ${\sim}n_i^{ex} \in \mathscr{L}\mathscr{D}(n_j^{ex})$ or ${\sim}n_j^{ex} \in \mathscr{L}\mathscr{D}(n_i^{ex})$, or both. Similar to the truncation we did to find the LDOI of a single node state, we do not include any node state that is the negation of any node state given in the intervention set and we stop searching that branch. We note that this truncation strategy avoids any following conflict. For example, if $n_C \in \mathscr{L}\mathscr{D}(n_A)$ and ${\sim}n_C \in \mathscr{L}\mathscr{D}(n_B)$, then one may expect that the LDOI of the set $\{n_A, n_B\}$ will have a conflict between n_C and ${\sim}n_C$. However, $n_C \in \mathscr{L}\mathscr{D}(n_A)$ implies that ${\sim}n_C$ requires ${\sim}n_A$, this means that meeting the conflict between n_C and ${\sim}n_C$, must be after meeting the conflict between n_A and $\sim n_A$, which is avoided by our truncation strategy.

For a compatible set $\{n_i^{ex}\} \equiv \cup_i n_i^{ex}$, it is guaranteed that $\cup_i \mathscr{LD}(n_i^{ex}) \subseteq \mathscr{LD}(\cup_i n_i^{ex})$. For example, as shown in Fig. 3.1 (e), the node set $\{n_2, n_4\}$ is a compatible node set as $\mathscr{LD}(n_2) =$, $\mathscr{LD}(n_4) =$ and $\mathscr{LD}(\{n_2, n_4\}) = \{\sim n_3, n_0, n_4, \sim n_1, n_2\}$. Note $\mathscr{LD}(n_2) \cup \mathscr{LD}(n_4) \subseteq \mathscr{LD}(\{n_2, n_4\})$. However, for an incompatible set, we just know that the situation $\cup_i \mathscr{LD}(n_i^{ex}) \subsetneq \mathscr{LD}(\cup_i n_i^{ex})$ cannot happen and all the remaining situations are possible. In the network of Fig. 3.1, node set $\{n_2, \sim n_4\}$ is an incompatible node set as $\mathscr{LD}(n_2) =$, $\mathscr{LD}(\sim n_4) = \{n_3, \sim n_0, n_1, \sim n_2\}$ and $\mathscr{LD}(\{n_2, \sim n_4\}) = \{n_3, \sim n_0, n_1\}$. Note that neither n_4 nor $\sim n_2$ are included in $\mathscr{LD}(\{n_2, \sim n_4\})$ due to the truncation rule and $\mathscr{LD}(\{n_2, \sim n_4\}) \subsetneq \mathscr{LD}(n_2) \cup \mathscr{LD}(\sim n_4)$. Node set $\{\sim n_1, n_3\}$ is another incompatible set as $\mathscr{LD}(\sim n_1) = \{n_2\}$, $\mathscr{LD}(n_3) = \{\sim n_0, n_1, \sim n_2, n_4, n_3\}$ and $\mathscr{LD}(\{\sim n_1, n_3\}) = \{n_2, \sim n_0, \sim n_4, n_3\}$. Note that $\mathscr{LD}(\{\sim n_1, n_3\}) \not\subset \mathscr{LD}(\sim n_1) \cup \mathscr{LD}(n_3)$ and $\mathscr{LD}(\sim n_1) \cup \mathscr{LD}(n_3) \not\subset \mathscr{LD}(\{\sim n_1, n_3\})$.

The properties of the LDOI of a single node can also be generalized to the LDOI of a given node set. For the first key property, let $S_j = \{\sigma_j = \tilde{\sigma}_j\}$ and $S_i = \{\sigma_i = \tilde{\sigma}_i\}$ be two sets of node states, if S_i is a subset of any (partial) fixed point and $S_j \subseteq \mathscr{LD}(S_i)$, then $\mathscr{LD}(S_j) \subseteq \mathscr{LD}(S_i)$. The intuition is similar, the requirement restricting us to consider those nodes which can be stabilized in the long term, that is, we rule out the possibility of S_i being an incompatible node set. For example in Fig. 3.1 consider $S_i = \{\sim n_3\}$ and $S_j = \{n_2, n_4\}$. As $\sim n_3$ is part of the stable motif $SM_2 = \{n_0, \sim n_1, n_2, \sim n_3, n_4\}$, corresponding to the fixed point $(\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4) = (1, 0, 1, 0, 1)$, $S_j \subset \mathscr{LD}(S_i)$ implies $\mathscr{LD}(S_j) \subseteq \mathscr{LD}(S_i)$. In fact, $\mathscr{LD}(S_j) = \mathscr{LD}(S_i)$.

The second key property also generalizes: if the logic domain of influence of a given node state set contains the set itself, then the logic domain of influence of the set contains at least one stable motif. The intuition and proof is similar to the case of a single node state. Taking the same example, consider $S_i = \{\sim n_3\}$ and $S_j = \{n_2, n_4\}$, note that both $S_i \subset \mathscr{LD}(S_i)$ and $S_j \subset \mathscr{LD}(S_j)$, this implies that both $\mathscr{LD}(S_i)$ and $\mathscr{LD}(S_j)$ contain a stable motif, which is SM_2 in this case.

Following these examples, we define the core of a stable motif to be a minimal subset of the stable motif whose logic domain of influence contains the stable motif. Here by minimal we mean that no true subset of the core of the stable motif will contain the entire stable motif. The core of a stable motif can be a single node or more than one node. For example, as shown in Fig. 3.1 (c) $\sim n_3$ is a single-node core of the stable motif $SM_2 = \{n_0, \sim n_1, n_2, \sim n_3, n_4\}$. $\{n_2, n_4\}$ is another core of the same stable motif as

$SM_2 \not\subseteq \mathscr{L}\mathscr{D}(n_2)$, $SM_2 \not\subseteq \mathscr{L}\mathscr{D}(n_4)$ and $SM_2 \subseteq \mathscr{L}\mathscr{D}(\{n_2, n_4\})$.

We also define a driver node (set) of the stable motif to be a node (set) whose domain of influence contains the entire stable motif. The driver node (set) can be inside the stable motif, in which case it is the core of the stable motif; it can also be an upstream node that is sufficient to activate (the core of) the stable motif. We note that stabilization of a stable motif does not require the sustained state of a driver node, that is, oscillations can also lead to the stabilization of a stable motif. An example of this behavior was shown in Fig. 3.2 (b): node B, which constitutes a self-sustaining stable motif, can stabilize by a single instance of A=1, regardless of the fact that the negative self-regulation of A makes it oscillate.

3.2.7 Target control algorithm

Now that we have equipped ourselves with the tool of LDOI to find the long term effect of a sustained intervention, we can formulate the target control problem as the identification of a node set S^* whose logic domain of influence contains the target node state set, *i.e.* $\mathscr{L}\mathscr{D}(S^*) \supseteq Target$. This problem can be framed as a planning search problem [123]. We start with a null set whose LDOI is also null. We repeatedly add a new node to the set until the LDOI of this set contains the target node state set. We use LDOI instead of DOI for this purpose because identification of the DOI is a computationally more difficult problem. Our current solution using LDOI sets a tight upper bound for the optimal solution for the target control problem as $\mathscr{D}(S^*) \supseteq \mathscr{L}\mathscr{D}(S^*) \supseteq Target$.

In order to avoid a full state space search in this combinatorial search problem, we apply a random heuristic algorithm called the greedy randomized adaptive search procedure (GRASP) [124, 125]. The pseudocode is described in Algorithm Table 1 and 2. The algorithm consists of two main phases. The first phase is the construction of a greedy randomized solution and the second phase is a local search to remove any redundancy of the solution.

In the first phase, we first generate an initial candidate list (line 4 in Algorithm 2). In the simplest case, the initial candidate list is all the non-composite nodes of the expanded network except the nodes in the target set and their negation, both of which are ineligible for control. One can also be more selective to adapt to the specific needs of controlling biological systems. For example, we can forbid the use of certain nodes or node states when constructing the initial candidate list, to incorporate the fact that certain chemical

Algorithm 1 GRASP algorithm for Target Control Problem

```
 1: procedure GRASP($G\_expanded$,$Target$,$max\_itr$)
 2:     $solutions \leftarrow List()$
 3:     for $index \leftarrow 1, max\_itr$ do
 4:         $solution \leftarrow$ConstructGreedyRandomizedSolution($G\_expanded, Target$)
 5:         $solution \leftarrow$LocalSearch($G\_expanded, Target, solution$)
 6:         if $solution$ then
 7:             $Solutions.append(solution)$
 8:         end if
 9:     end for
10:     return $solutions$
11: end procedure
```

Algorithm 2 Algorithm for constructing a greedy randomized solution

```
 1: procedure CONSTRUCTGREEDYRANDOMIZEDSOLUTION($G\_expanded, Target$)
 2:     $solution \leftarrow Set()$
 3:     $\alpha \leftarrow random(0, 1)$
 4:     $candidates \leftarrow$Construct_Initial_Candidates($G\_expanded, Target$)
 5:     while $candidates$ do
 6:         $RCL \leftarrow$MakeRCL($candidates, alpha$)
 7:         $s \leftarrow$Select_Candidate($RCL$)
 8:         $solution \leftarrow solution \cup \{s\}$
 9:         if $Target \subset LDOI(solution)$ then
10:             return $solution$
11:         end if
12:         Update_Candidates($candidates$)
13:     end while
14:     return $Set()$
15: end procedure
```

species are harder or even unrealistic to control. Thus these nodes/chemical species will never appear in the final solution since they are not in the initial candidate list.

Then, we begin the procedure of iteratively adding nodes to the trial solution set (which is initially empty) and evaluating whether the LDOI of the trial solution set covers the target set. We form a restricted candidate list (RCL) based on a greedy measure $G(v)$ defined for each candidate node v in the candidate list (line 6 in Algorithm 2). A greedy function is a heuristic score to estimate whether this node should be included in the solution. We discuss several choices of $G(v)$ below. We determine the minimum score $G_{min} = min_{v \in V}\ G(v)$ and maximum score $G_{max} = max_{v \in V}\ G(v)$ among the heuristic scores of all the nodes. Then we use a previously generated random number α from a uniform distribution between 0 and 1 to set a passing score for the RCL as $G_{pass} = G_{min} + \alpha \cdot (G_{max} - G_{min})$. Then the RCL consist of nodes whose greedy function is no less than the passing score, i.e., $RCL = \{v \in V | G(v) \geq G_{pass}\}$.

Next we randomly pick a node from the RCL and add it to the current trial solution (line 7 and 8 in Algorithm 2). If after this addition the LDOI of the solution covers the target set, we end the first phase and start the second phase (local search procedure) with this candidate solution (line 9 and 10 in Algorithm 2). Otherwise, we update the candidate node set and start the next iteration toward adding another node from the RCL to the trial solution set. We update the candidate node set by removing the previously added node, its negation and any node in the LDOI of the current trial solution (line 12 in Algorithm 2). We do this latter exclusion because these nodes will stabilize because of the current trial solution, and it is useless to add any stabilized state to the trial solution. We repeat the whole procedure including selecting a node randomly from the candidate set as long as there are still candidate nodes (line 5 in Algorithm 2). We return an empty set if we do not find a solution (line 14 in Algorithm 2).

In the second phase (see the pseudocode in Appendix), we start with a candidate solution that covers the target set. We randomize the order of nodes in the candidate solution and then iteratively attempt to remove each node. If after removing this node the LDOI of the modified solution still covers the target set, then we replace the candidate solution with the modified solution. Thus after one iteration of traversing all the nodes, we obtain a final solution. At worst, no node is removed from the set and the final solution is the same as the candidate solution. The randomness in the removal order provides a possibility for obtaining different minimal solutions from the same candidate solution.

In this random heuristic algorithm, we introduce two aspects of randomness in the construction phase, one is the randomness of the passing score by a different α for each iteration of solution generation process (line 3 in Algorithm 1) and another is the random selection of a node each time from the RCL inside each solution generation process (line 7 in Algorithm 2). These techniques help strike a balance between the bias of a greedy function and exploring the whole state space [124, 125]. An efficient greedy function/ heuristic score is important to guide the search procedure towards the subspace with the optimal solution. However, a universally efficient greedy function may not exist; rather, the efficiency of a greedy function may depend on the specific network structure and target set. We have implemented five choices of greedy functions

$G(v)$ for a given node state (equivalently, non-composite node of the expanded network): score 1 is the size of the LDOI of that node state (denoted as $|LDOI|$); score 2 is the size of the set of composite nodes which are nearest neighbors of the LDOI of that node state (denoted as $|Comp\_LDOI|$); score 3 is a linear combination of the previous two measures with equal weight (denoted as $Scores\_1{+}2$), and score 4 and 5 as the size of the LDOI of that node state with penalty if the LDOI contains a node that is the negation of a node in the target set (denoted as $|LDOI|\_Pen1$ and $|LDOI|\_Pen2$). The penalty can be implemented by multiplying this score by -1 (score 4) or by decreasing this score by the size of the largest LDOI among all node states (score 5); both of these implementations ensure that this score becomes non-positive. All relevant code is available at https://github.com/yanggangthu/BooleanDOI .

3.2.8 Computational complexity of the target control algorithm

The time complexity of calculating the LDOI of any set is bounded by $O(N_{ex}+E_{ex})$, where N_{ex} is the number of nodes and E_{ex} is the number of edges of the expanded network. For each non-composite node in the network, we initially calculate its LDOI and the value of its greedy function, with time complexity $O(N(N_{ex}+E_{ex}))$, where N is the number of nodes in the original network. We then cache these results to improve the performance of the GRASP algorithm. In the first phase of the GRASP algorithm, we run at most N iterations and we need to calculate the LDOI of the trial solution in each iteration, thus the time complexity is bounded by $O(N(N_{ex}+E_{ex}))$. In the second phase, the time complexity is also bounded by $O(N(N_{ex}+E_{ex}))$ as we need to go through each node, bounded by $O(N)$ as a crude estimate, delete the node from the solution and check the modified solution$'$s LDOI, which is $O(N_{ex}+E_{ex})$. The Boolean regulatory functions of biological network models are often nested canalizing rules [99, 100], thus for each node with k regulators there are at most k newly generated composite nodes in the expanded network, as well as two corresponding non-composite nodes; each of these nodes have at most k regulators. Thus N_{ex} is bounded by $O(\bar{k}N)$, and E_{ex} is bounded by $O(\overline{k^2}N)$. Biological networks are sparse, with an average node in-degree $1<\bar{k}<3$ [11]. Thus the complexity of the target control algorithm applied to biological network models is $O(\overline{k^2}N^2)\sim O(N^2)$ for a well-behaved degree distribution in the sparse limit and bounded by $O(N^3)$ for an extremely skewed degree distribution in the sparse limit. Different iterations of the solution generation process (line 3 in Algorithm 1) can be easily parallelized as each iteration is independent.

3.2.9 Damage mitigation as target control

We can generalize the target control algorithm to solve a damage mitigation problem. Consider a Boolean network that has two steady states, one corresponding to the normal state of the system and the other corresponding to a disease state. The system is currently in the normal steady state, but damage to a node, which causes it to stabilize in the opposite state, will lead the system to the disease steady state without any intervention. Under such conditions, previous research has proposed modifying the network topology (as soon as possible, or preventatively) to block the propagation of damage [80]. Here we are interested in designing a damage mitigation strategy to bring the system back to an attractor similar to the normal steady state in the sense that a subset of nodes are in the same state as their states in the normal steady state. This problem is almost the same as the target control problem except that we need to take the permanent damage into consideration. There are two ways of incorporating this. First, we treat this permanent damage as an initial condition and apply network reduction to the system. However, this risks reducing a significant fraction of the nodes in the network, including the target nodes we are interested in. Second, we can apply our GRASP algorithm as above while initializing the solution with the damaged node state(s) and forbidding the damaged node state to be removed in the local search phase in GRASP algorithm. This means that we include the damage as part of the treatment/intervention. When the LDOI of the node state set containing the damage effect covers the target set, the target nodes will stabilize in their desired states after a finite number of time steps under all initial conditions of the subspace of the damaged network. We note that we only need to do this when the damage is a permanent one; when the damage is temporary (i.e. when the node is allowed to go back to its original state), this can be treated as a different initial condition for the target control problem and we can still apply our GRASP algorithm to solve it as DOI/LDOI is robust to any initial condition by definition.

3.3 Results

3.3.1 Application to ensembles of random Boolean networks

We tested the two proposed properties of the LDOI and the target control algorithm on different random Boolean network ensembles. Specifically, we generated an ensemble of 1000 random networks, with size ranging from 15 to 50 nodes and average in-degree ranging from 1 to 2 . The Boolean regulatory functions of the random ensemble are required to be effective (irreducible) Boolean functions [126] to be consistent with the generated topology, or nested canalizing functions to simulate biological systems. We have successfully tested and validated the two properties for the LDOI of each node in the generated networks. We also tested and validated the properties of the LDOI of node sets of size up to 3∼7 depending on the specific network (as the complexity of testing the property grows faster than N^k for $k << N$, where N is the network size and k is the node set size).

With respect to testing the target control algorithm, we generate 50 random target sets with size 2 or 3 for each random network. It may not always be possible to find a solution for a specific target set for a network, especially when the Boolean network model does not have a (partial) fixed point type of attractor (i.e. if all nodes oscillate in the attractor) or when the desired target state set consists of node states that are part of different attractors, which conflict with each other. In the simulations of the two ensembles mentioned above, we verified that we are able to find a solution for more than 99.5% of the target sets when the target set satisfies two criteria: (i) it is a subset of a (partial) fixed point and (ii) the targets in this set are accessible from nodes outside of this set in the original network (that is, the targets do not consist of source nodes only and do not form a motif without any incoming edges). Note that there can be counter-examples where satisfying these criteria is not sufficient to find a solution. For example, in Fig. 3.2 (a) and (b), there are no solutions for the target set $\{\sim B, \sim C\}$ as the remaining nodes are not enough to activate the composite node in Fig. 3.2 (b). However, the probability of such situations is small in both random ensembles with moderate size and real biological network. Moreover, the fact that one cannot find a solution through our GRASP algorithm for the target control problem often indicates that the target set is not a reasonable target. It is likely that one would not be able to find a solution in such situation even with a whole state space search.

We also test the performance of different heuristic functions for the target control problem. We calculate the average number of generated solutions for each pair formed by a target set and a network. As shown in Table 3.1, greedy functions with a penalty for containing the negation of a node state included in the target set (score index 4 and 5) consistently perform better than the greedy functions directly using the size of the LDOI (score index 1 and 3). The intuition behind this is clear, the binary essence of the node state is important and it is thus more efficient to choose from those nodes whose domain of influence does not contain the undesired node state. The second greedy function ($|Comp\_LDOI|$) also performs quite well.

Table 3.1: Mean number of solutions found for each target set and random network pair for 50 target sets and 1000 networks. Half of 50 target sets have size two and the other half is of size three; none of them contain source nodes. The 2^{nd} to 6^{th} columns correspond to different custom score (greedy function) indexes and notations, which are described in the last paragraph in Sec. 3.2.7. The second and third row corresponds to the random network ensemble with nested canalizing rules and effective Boolean rules respectively.

Custom Score Index and Notation	1 $\|LDOI\|$	2 $\|Comp\_LDOI\|$	3 $Scores\_1{+}2$	4 $\|LDOI\|\_Pen1$	5 $\|LDOI\|\_Pen2$
Nested Canalizing Rules	12.29	31.66	12.30	31.21	40.64
Effective Boolean Rules	26.21	61.94	26.22	57.08	66.91

3.3.2 Biological Examples

We applied our methodology on four Boolean models of signal transduction networks. In the following we demonstrate our algorithm on two of these, the epithelial-to-mesenchymal transition (EMT) network and the PI3K mutant ER+ breast cancer network. The results on the ABA induced stomatal closure network and the T-LGL leukemia network are shown in Appendix and.

3.3.2.1 EMT network

EMT is a cell fate change involved in embryonic development, which can be reactivated during cancer metastasis [5]. During EMT, epithelial cells lose their original adhesive property, and become mesenchymal cells which leave their primary site, invade neighboring tissue, and migrate to distant sites. A Boolean network model of EMT in the context

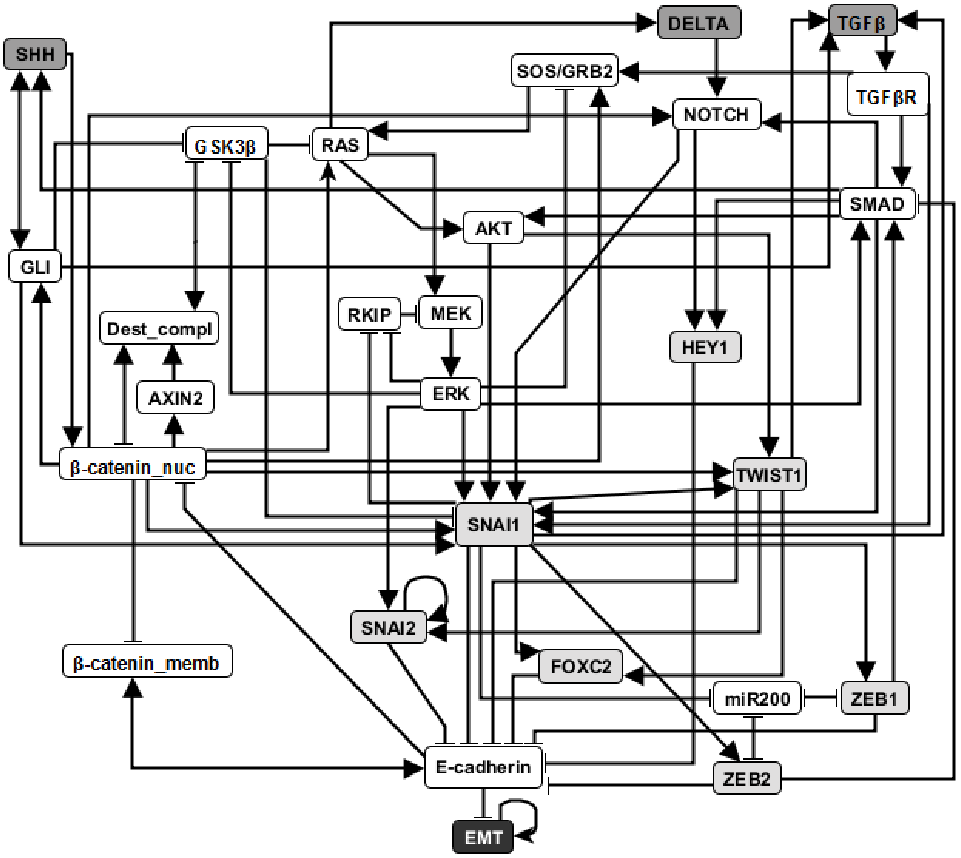


Figure 3.3: An illustration of the EMT network. Attractor-preserving network reduction was applied to better focus attention on the most relevant nodes. Nodes with light gray background are direct regulators of E-cadherin and nodes with dark gray background represent external signaling molecules. Edges ending with an arrow represent positive regulation while edges end with a flat bar represent negative regulation. See more details in Appendix .

of hepatocellular carcinoma invasion has been established by Steinway *et al.* [5]. Several predictions of this model were validated experimentally [5, 109]. The EMT network has 70 nodes and 135 edges. The adhesion factor E-cadherin is the sink node; its OFF state indicates the transition to a mesenchymal state. The network has a normal (epithelial) steady state and an abnormal (mesenchymal) steady state. (See details in Appendix). In Fig. 3.3 we show a simplified version of the EMT network; our analyses were done on the full network.

Previous research on this network has indicated that sustained activation of TGFβ signal can trigger EMT through the activation of eight stable motifs [109]. In addition, stabilization of any of these stable motifs can drive EMT. Our target control algorithm shows that any of 60 node states (out of 138 node states for the 69 nodes) can lead to EMT, including the previously established EMT drivers. As we are more interested in designing therapeutic strategies to convert the abnormal steady state into a normal steady state, the negation of EMT is a more relevant target. Previous analysis indicated that when considering an initial epithelial state and turning on the TGFβ signal, the knockout of any of the transcription factors that downregulate E-cadherin (i.e. knockout of SNAI1, SNAI2, FOXC2, TWIST1, ZEB1, ZEB2, HEY1) or multiple double node knockout combinations (knockout of SMAD and one of RAS, CSL, DELTA, NOTCH, NOTCH_ic, SOS/GRB2) are effective in blocking EMT (i.e. leading to E-cadherin=ON). The effectiveness of transcription factor knockout had been established in the literature; unfortunately these transcription factors cannot be targeted with existing drugs. Several double knockout combinations were validated experimentally in [109] and are more amenable to drug targeting.

For EMT as target, our target control algorithm gives 7 two-node solutions (activation of β-catenin_memb and knockout of any of SNAI1, SNAI2, FOXC2, TWIST1, ZEB1, ZEB2, HEY1) and 5 three-node solutions (activation of β-catenin_memb, knockout of SMAD and knockout of any of RAS, CSL, DELTA, NOTCH and NOTCH_ic). The main difference between the target control solution and the previously found EMT-blocking single and double knockout interventions is that our target control solution includes the additional control of β-catenin_memb. To understand this difference, we note that EMT is in the LDOI of TFGβ, however, EMT is not in the LDOI of the set consisting of TGFβ together with any of the previously found EMT-blocking knockout interventions. This indicates that the knockout intervention is effective in the sense that it can block the process of reaching EMT. However, $\sim$EMT is also not in the LDOI of the set of TGFβ together with any knockout intervention. The knockout intervention is effective when the initial condition is the epithelial steady state, however the knockout intervention does not block EMT for all initial conditions. The target control algorithm, which can block EMT for all initial conditions, requires one more node (β-catenin_memb) in the target control solution. In fact, treating this problem as a damage mitigation problem, where the damage is sustained activation of TGFβ, we verify that EMT is in the LDOI of TGFβ together with any of the target control solutions.

As established in previous results, the single node EMT-blocking knockouts do not lead back to an epithelial state but rather to hybrid epithelial or mesenchymal steady states [109]. The hybrid epithelial steady state has certain epithelial features, e.g. E-cadherin and β-catenin_memb are activated, and also some mesenchymal features, e.g. MEK, ERK and SNAI1 are activated. The hybrid mesenchymal steady state demonstrates the opposite features compared to the epithelial steady state. A good target set to avoid reaching such a hybrid state (which is likely pathological and may even be a worse outcome as the mesenchymal state) would be $\{\sim$EMT, $\sim$MEK$\}$ [109]. The minimum solution found involves controlling three nodes: activation of β-catenin_memb, inhibition of SNAI1, inhibition of RAS or RAF. We also find a four-node intervention that does not involve ERK and SNAI1: activation of β-catenin_memb, miR2000 and RKIP, and also inhibition of RAS. If the target set is $\{\sim$EMT, $\sim$MEK, $\sim$SNAI1$\}$, the minimum solution size is found to be six.

Stable motif control indicates that control of at least five nodes is needed to drive any initial state (including the mesenchymal state) to the epithelial state (see Supplemental Table 3 of [109]) Although the control goal is different, one can still see the connection between our target control solution for the target $\sim$EMT and the stable motif control solution (to drive the system to the epithelial state). Specifically, they both require activation of β-catenin_memb. Knockout of SNAI1 , knockout of TWIST1 or knockout of SMAD and RAS, as one of the target control solutions, also appear as a part of stable motif control solution that does not require control of TGFβ or TGFβR.

These results demonstrate both the accuracy and effectiveness of our target control algorithm, as the solutions found through 1000 iterations are comprehensive (comparable to the solution found through a systematic search of knockout pairs) and indifferent to the distance to the target nodes.

3.3.2.2 Breast cancer network

Zañudo *et.al*. established a discrete dynamical model of the signal transduction processes involved in the PI3K mutant, estrogen receptor positive (ER+) breast cancer, as shown in Fig. 3.4 [110]. The model includes 58 nodes, which correspond to proteins, transcripts, drugs, and two cellular outcomes, apoptosis (programmed cell death) and proliferation (cell cycle progression). A fraction of the nodes (16), including the outcome nodes, are characterized by multiple levels, which is implemented by additional virtual nodes, e.g. apoptosis2 corresponds to level 2 of apoptosis, which has a more stringent regulatory function than apoptosis1 (level 1 of apoptosis). This network as implemented is essentially a Boolean network because all the regulatory functions are Boolean [110]. The

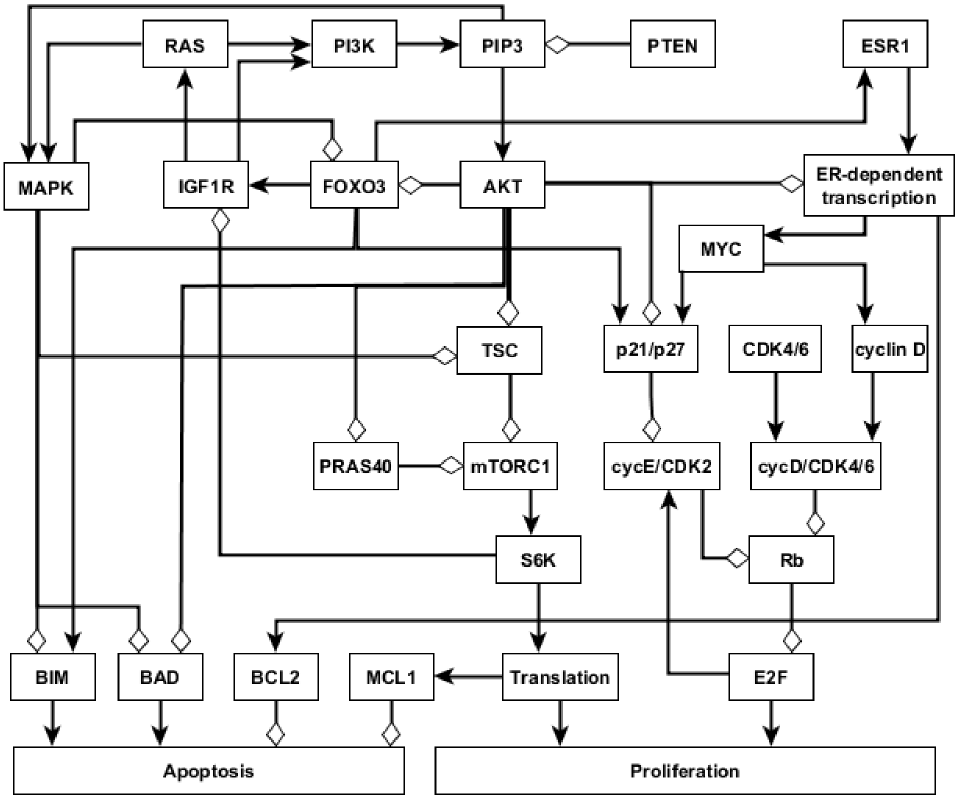


Figure 3.4: An illustration of the PI3K mutant, ER+ breast cancer network. Attractor-preserving network reduction was applied to focus on the nodes most relevant to our analysis. Nodes are colored according to the signaling pathway that they participate in. Edges ending with an arrow represent positive regulation while edges ending with a hollow diamond represent negative regulation. See more details in Appendix

network model successfully captures the key role of the PI3K/AKT/mTOR signaling pathway in determining the pathological proliferation and survival of cancer cells. In untreated simulated cancers cells, PI3K, MAPK, AKT, mTORC1 and ER signaling are active, leading to high level of proliferation and lack of apoptosis. The network model successfully captures the effectiveness of PI3K inhibiting drugs in leading to low level of proliferation and high level of apoptosis [110]. Through extensive simulations, the network model confirms known drug resistance mechanisms, i.e. additional mutations or other dysregulations that lead to the loss of effectiveness of PI3K-inhibiting drugs. It also predicts new possible resistance mechanisms and the degree of survivability under different resistance mechanisms. [110].

Similar insights can be drawn by applying the target control algorithm to the discrete dynamical network model without doing dynamical simulations, which demonstrate the rich information contained in the network topology and logic and the effectiveness of our control methodology. We obtained a (relatively large) reduced network by considering the system under the relevant initial condition of PI3K mutant, ER+ cancerous state, while keeping the seven drugs as source nodes (see details in Appendix .) For example, if we set the target to be high level of apoptosis (Apoptosis = 2), the algorithmâĂŹs output is inhibition of PI3K or PIP3. As the target control solution works for any initial condition of the reduced network, this result confirms the key role of PI3K in avoiding apoptosis. If we set the target to be high level of apoptosis and no proliferation, i.e., Target = $\{$Apoptosis2, $\sim$Proliferation$\}$, the algorithm gives multiple two-node interventions as minimal interventions, these consists of either of $\{\sim$PI3K, $\sim$PIP3$\}$ and inhibition of any node in the MYC-CDK4/6 axis of cell-cycle regulation, *i.e.*, $\{\sim$ESR1, $\sim$ER_transcription, $\sim$MYC, $\sim$CDK46, $\sim$cyclinD, $\sim$cycD_CDK46, $\sim$Rb, $\sim$E2F$\}$. There are several drugs that can target these nodes. For example, Alpelisib is a PI3K inhibitor, Fulvestrant is a ESR1 inhibitor and Palbociclib is a CDK4/6 inhibitor. This result is consistent with the results found in the [110]: inhibition of PI3K leads to an increase in ER transcriptional regulatory activity, leading to a decrease in proliferation, and simultaneous PI3K and ER inhibition has a synergistic effect in completely blocking proliferation and maintaining a high level of apoptotic activity. If PI3K inhibitor or PIP3 inhibitor is not allowed to be used, the algorithm finds three node solutions involving an AKT inhibitor (e.g. Ipatasertib), MAPK inhibitor (e.g. Trametinib) and inhibition of any node from the MYC-CDK4/6 axis of cell-cycle regulation. In other words, inhibition of AKT together with MAPK provides a similar functionality with inhibition of PI3K. One can also use the LDOI to identify possible drug resistance mechanisms, i.e. perturbations that make PI3K inhibition less effective. As $\{$Apoptosis2,$\sim$Proliferation4$\} \subset \mathscr{LD}(\sim$PI3K), we simply go through all possible two-node interventions containing PI3K inhibitor and screen out those interventions whose LDOI either does not contain Apoptosis2 or contain Proliferation3 or higher level (Proliferation4). We reproduce most of the potential drug resistance mechanism to PI3K inhibitors indicated in Table 3 of [110].

3.4 Discussion

In summary, we have developed the new measures DOI and LDOI to describe the long-term effect of a sustained intervention. We have applied these measures to find solutions to the target control problem in logical network models. This work takes a step forward towards practical control of real biological systems, as illustrated by the applications presented here. The target control solutions we find recover previous predicted interventions obtained by other methods (dynamic simulations and stable motif analysis). As several of these previous predictions are validated experimentally, this agreement also serves as validation of our target control solutions. Notably, by generating a large number of valid target control solutions, we are going significantly beyond previous results. The multitude of predicted target control interventions allows their filtering according to biological or technological considerations.

Here we assumed the existence of a discrete dynamical model. As there are significant uncertainties in the existing models due to the scarcity of experimental information, we estimate the sensitivity of the LDOI measure to the incompleteness of the dynamical model. As the primary way of obtaining causal information that can be used in a logical model is to perform knockout experiments, the predominant causal information indicates a node as being necessary for the activation of another node. For example, if the knockout of either of two regulators A or B leads to a decrease in the activity of target C, we would infer that the logical rule for C is $C = A$ AND B. Suppose that there is a so far undetected regulator of C, which we denote by X. This X will likely also be necessary, which would maintain agreement with the previous observations, *i.e.* $C = A$ AND B AND X is the true rule. Consider the rule for the complementary node $\sim C = \sim A$ OR $\sim B$ in the case of the incomplete system versus the true rule $\sim C = \sim A$ OR $\sim B$ OR $\sim X$. We can see that the LDOI of any of $\sim A, \sim B, A, B$ will be robust to the addition of X. The LDOI of node X and $\sim X$ need to be established in the true system. The LDOI of node state set $\{A, B\}$ will be affected by this change. (However, LDOI of $\sim A$ and $\sim B$ will not change.) Thus the size of the solution of the target control problem may increase due to this incomplete information. Due to the binary essence of the Boolean rule, missing a sufficient regulator (an extra OR rule) will give similar results.

The DOI and LDOI can be related to prior research on logical networks. The concept of elementary signaling mode (ESM), originally defined as a minimal subgraph that can propagate a signal from a source node to an output node, [70, 71] can be generalized to start from any node of a directed network and end in any node reachable from it. An ESM on the expanded network is the generalization of a path on a usual directed network. Similarly, the LDOI of a node on the expanded network is analogous to a connected component reachable from a node on a usual directed network. In the same way a connected component reachable from node i consists of nodes that have a path starting from node i, the LDOI of a node consists of all the nodes included in any ESM that starts from that node. Recent work by [127] developed a logic framework to identify causal relationships that are sufficient or necessary. This framework allows an alternative definition of the LDOI. The LDOI of the ON state of a node ($\tilde{\sigma}_i = 1$) includes all the nodes for which the node is a sufficient activator (these nodes will have $\tilde{\sigma}_j = 1$) or sufficient inhibitor (these nodes will have $\tilde{\sigma}_k = 0$). Similarly, the LDOI of the OFF state of a node includes all the nodes for which the node is a necessary activator (these nodes will have $\tilde{\sigma}_j = 0$) or necessary inhibitor (these nodes will have $\tilde{\sigma}_k = 1$).

An algorithm to construct ESMs through a backward search from an output node was presented in [128]; this algorithm can be adapted to find solutions of the target control problem of a single output. If we treat the output node as the root of a backward search, the set of nodes found in the ESM in each search depth (distance from the output node) can serve as a control solution. A truncation technique similar to ours needs to be applied to deal with inconsistent feed-forward or feed-back loops. This algorithm can be generalized to solve the target control problem of a target set by simultaneous search from each target node. We chose to transform the target control problem into a planning search problem; and it has been established that such a planning search problem can be solved in both a forward propagation and a backward propagation approach, or even a mixed approach [123]. It will be an interesting future work if such techniques can improve the efficiency of the algorithm.

This work points out interesting questions as future research directions. First, though evaluating DOI of a node (set) is computationally hard, a better estimation of DOI than LDOI is desirable and can be used to reduce the size of the solution given by our current target control algorithm. Second, the requirement that the solution works for all initial conditions in the setup of the target control problem gives robust solutions, however it may still be conservative for biological systems in certain applications, especially if one is certain about the relevant initial condition subspace. A semi-structural approach (without doing dynamical simulations) to solve the target control problem starting from a subspace of initial conditions are also desirable.

Chapter 4 | Structure-based control of complex networks with nonlinear dynamics

This chapter is based on published work [115], where I am the second author. This chapter was reproduced with permission from Jorge G.T. Zañudo, Gang Yang and Réka Albert, PNAS 2017 114 (28) 7234-7239, Copyright 2017, the National Academy of Science of the USA.

4.1 Introduction

Controlling the internal state of complex systems is of fundamental interest and enables applications in biological, technological and social contexts. An informative abstraction of these systems is to represent the system's elements as nodes and their interactions as edges of a network. Often asked questions related to control of a networked system are how difficult to control it is, which network elements need to be controlled, and through which control actions, to drive the system toward a desired control objective [6, 24, 77, 113, 114, 129–134]. As discussed in Sec. 1.5, we face the challenge of incomplete information including unknown dynamics mechnism and price parameter for these non-linear control problems in biological system. Among control frameworks, structure-based methods distinguish themselves due to their ability to draw dynamical conclusions based solely on network structure and a general assumption about the type of allowed dynamics. For example, structural controllability, which assumes unspecified linear dynamics or linearized nonlinear dynamics, allows the identification of the minimal number of nodes whose receiving an external signal $u(t)$ drives the system into a state of interest [76, 135].

Despite its success and wide-spread application [73, 136–139], structural controllability may give an incomplete view of the network control properties of a system. In case of systems with nonlinear dynamics it provides sufficient conditions to control the system in the neighborhood of a trajectory or a steady state ([24, 73], SI Appendix), and its definition of control (full control; from any initial to any final state) does not always match the meaning of control in biological, technological, and social systems, in which control tends to involve only naturally occurring system states [140]. In addition to the approaches provided by nonlinear control theory [73, 132–134], new methods of network control have been proposed to incorporate the inherent nonlinear dynamics of real systems and relax the definition of full control [6, 73, 113, 117, 134]. Only one of these methods, namely feedback vertex set control (FC), can be reliably applied to large complex networks in which only the structure is well known and the functional form of the governing equations is not specified. This method, introduced by Fiedler, Mochizuki et al in [77, 141], incorporates the nonlinearity of the dynamics and considers only the naturally occurring end states of the system (e.g. steady states and limit cycles) as desirable final states.

In this work, we use feedback vertex set control on biological, technological, and social networks to predict the nodes whose override (by external control) can steer a network's dynamics towards any of its natural long term dynamic behaviors (its dynamical attractors). We identify the topological characteristics underlying the predicted node overrides, compare the obtained results with those of control theory's structural controllability [24, 76, 135] and identify the model-dependent and model-independent overrides it provides for network models with parameterized dynamics.

4.2 Structural controllability

In structural controllability (SC) we consider a system with an underlying network structure whose autonomous dynamics are governed by linear time-invariant ordinary differential equations

$$\frac{d\mathbf{x}}{dt} = A\mathbf{x}(t), \tag{4.1}$$

where $\mathbf{x}(t) = (x_1(t), x_2(t), \ldots, x_N(t))$ denotes the state of the system, and A is a $N \times N$ matrix that encodes the network structure and is such that a_{ik} is nonzero only if there is a directed edge from k to i. Given this system, SC's aim is to identify external driver node signals $\mathbf{u}(t) = (u_1(t), \ldots, u_M(t))$ that can steer the system from any initial state to any final state in finite time (i.e., full control, 4.1a), and that are coupled to Eq. 4.1 in the following way

$$\frac{d\mathbf{x}}{dt} = A\mathbf{x}(t) + B\mathbf{u}(t), \tag{4.2}$$

where B is a $N \times M$ matrix that describes which nodes are driven by the external signals $\mathbf{u}(t)$.

Lin, Shields, Pearson, and others have showed that if such a system can be controlled in the specified way by a given pair (A, B), this will also be true for almost all pairs (A, B) (except for a set of measure zero) [76, 132, 135]. A well-known controllability test is given by Kalman′s rank condition, namely, the $N \times NM$ matrix $(B, AB, A^2B, \ldots, A^{N-1}B)$ has full rank, i.e., $\text{rank}(C) = N$ [75]. In other words, SC is necessary and sufficient for control of almost all linear time-invariant systems consistent with the network structure in A. The applicability of SC also extends to nonlinear systems; SC of the linearized nonlinear system around a steady state or system trajectory of interest is a sufficient condition for local controllability of the system around said steady state or trajectory in a sufficiently small time [73, 82, 132]. Furthermore, SC of the linearized nonlinear system is also a sufficient condition for some nonlinear notions of controllability such as accessibility [73, 82, 132].

The mathematical intuition behind SC is that a node can fully manipulate only one of its successor elements at a time and that a directed cycle is inherently self-regulatory. A consequence of this is that the driver nodes are such that every network node is either part of a set of non-intersecting linear chains of nodes that begin at the driver nodes or is part of a set of directed cycles that do not intersect each other or the set of linear chains and which are reachable from the driver nodes (4.1). As Ruths & Ruths showed [130], this implies that there are three types of network nodes that must be directly manipulated by a unique driver node, and which we call SC nodes: (i) every source node, and every successor node of a dilation (when a node has more than one successor node) that is not part of the set of linear chains or of the cycles, namely (ii) the surplus of sink nodes with respect to source nodes or (iii) internal dilation nodes.

4.3 Structure-based network control with nonlinear dynamics

Most real systems are driven by nonlinear dynamics in which a decay term prevents the system's variables from increasing without bounds. The state of the system's N nodes at time t, characterized by source node variables $S_j(t)$ (for nodes with no incoming edges) and internal node variables $X_i(t)$, obeys the equations

$$dX_i/dt = F_i(X_i, X_{I_i}, t), \tag{4.3}$$

$$dS_j/dt = G_j(t), \tag{4.4}$$

where $i = 1, \dots, N - N_s$, $j = N - N_s + 1, \dots, N$, and N_s is the number of source nodes. The dynamics of each source node j is independent of the internal node variables X_i (by definition), is fully determined by $G_j(t)$, and does not include a decay term. In the simplest case $G_j = 0$ and S_j will remain in its specified initial value. The dynamics of each internal node i is governed by $F_i(X_i, X_{I_i}, t)$, which captures the nonlinear response of node i to its predecessor nodes I_i (which can be source or internal nodes), and which includes decay in the dependence of F_i on X_i ($\partial_1 F_i(X_i, X_{I_i}, t) < 0$, where ∂_1 indicates the partial derivative with respect to the X_i argument but not the X_{I_i} argument). Additionally, F_i and its first derivatives are assumed to be continuous functions and are assumed to be such that $\mathbf{X}(t)$ is bounded ($|\mathbf{X}(t)| < C$ for some constant C) for any finite initial condition $\mathbf{X}(t_0)$ and for all $t \geq t_0$, including the limit $t \to \infty$.

Functions used to describe the dynamics of birth-death processes [142, 143], epidemic processes [142, 144, 145], biochemical dynamics [7, 146], and gene regulation [1, 7, 19, 146], usually follow the form $F_i = f_i(X_{I_i}) - \alpha_i(X_{I_i})X_i$, which satisfy the above conditions. As an example, $X_i(t)$ can denote the concentration of proteins involved in a signal transduction pathway, and $S_j(t)$ the concentration of extracellular signals (molecules). In this case f_i can take the form of a Hill function (e.g. $f_i = \beta_i X_k^2/(X_k^2 + \theta^2)$ if k is the only node in I_i) or of a mass-action term (e.g. $f_i = \beta_i X_k X_l$ if k and l are the only nodes in I_i). As an alternative example, $X_i(t)$ can denote the probability that an individual is infected in a contagion network and $S_j(t)$ the influence of vaccination or prevention measures on certain individuals, and F_i can take the form of a susceptible-infected-susceptible model term (e.g. $F_i = \beta_i X_k(1 - X_i) - \alpha_i X_i$ if k is the only node in I_i).

The dynamics described by Eqs. 4.3 are such that they possess some naturally occurring end states, or dynamical attractors. Dynamical attractors in biological, social, and technological systems represented by networks have been found to be identifiable with the stable patterns of activity of the system. E.g., in gene regulatory networks dynamical attractors correspond to cell fates [1, 7, 19]; in opinion spreading dynamics on social networks they correspond to opinion consensus states of groups of individuals [145]; and in disease or computer virus spreading they correspond to the long-term (endemic) patterns of infected elements [144].

In many systems there is adequate knowledge of the underlying wiring diagram but not of the specific functional forms and parameter values required to fully specify F_i and G_j. Analyzing such systems requires the use of structure-based control methods such as feedback vertex set control (FC). FC, developed by Fiedler, Mochizuki et al. [77, 141], is a mathematical formalization of the following idea: in order to drive the state of a network to any one of its naturally occurring end states (dynamical attractors) one needs to manipulate a set of nodes that intersects every feedback loop in the network - the feedback vertex set (FVS). This requirement encodes the importance of feedback loops in determining the dynamical attractors of the network, a fact that was recognized early on in the study of the dynamics of biological networks [120, 147]. Fiedler, Mochizuki et al. mathematically proved that for a network governed by the nonlinear dynamics of Eq. 4.3, the control action of forcing (overriding) the state variables of the FVS into the trajectory specified by a given dynamical attractor of Eq. 4.3 ensures that the network will asymptotically approach the desired dynamical attractor, regardless of the specific form of the functions F_i. Mathematically, consider a differential equation system governed by Eq. 4.3 with dissipative functions F_i, and the associated directed graph G obtained from the I_i. We also assume F_i and its derivatives to be continuous. Moreover, G can contain a self-loop only if F_i does not satisfy the decay condition $\partial F_i / \partial X_i < 0$. Then a possibly empty subset $J \subseteq \{1, 2, \ldots, N\}$ of vertices of G, and any two solutions $\mathbf{X}$ and $\widetilde{\mathbf{X}}$ of Eq. 4.3 satisfy

$$\lim_{t\to\infty}\left(X_J(t) - \widetilde{X}_J(t)\right) \to 0 \qquad\qquad \text{implies}$$
$$\lim_{t\to\infty}\left(\mathbf{X}(t) - \widetilde{\mathbf{X}}(t)\right) \to \mathbf{0}$$

for all choices of nonlinearities F_i if and only if J is a feedback vertex set (FVS) of the graph G. Note that FC does not utilize a controller or driver signal, and instead considers node state override as its control action. [1]

This type of intervention is often used in biological systems, with examples such as genome editing or pharmacological treatment [140, 148], and in epidemic spreading networks, where vaccination is a node state override that prevents a node from being infected. When using node state overrides as the control action, controlling the FVS is sufficient to drive the system to any of its attractors for each form of F_i and necessary if this must hold for every F_i ([77, 141] and . The problem of exactly identifying the minimal FVS is NP-hard, but a variety of fast algorithms exist to find close-to-minimal solutions. For example, to solve the FVS problem, Pardolos *et. al.* adapt a heuristic algorithm known as the greedy randomized adaptive search procedure (GRASP) [125, 149], which is commonly used for combinatorial optimization problems. In addition, Galinier *et. al.* established another efficient heuristic algorithm to solve the minimal FVS problem, a simulated annealing algorithm with a novel local search procedure [150]. We implemented the later algorithm and the code is available at https://github.com/yanggangthu/FVS_python.

In the structural theory of Mochizuki et al., every element is governed by Eq. 4.3. It is assumed that the source nodes converge to a unique state (or trajectory) and do not need independent control; thus they are iteratively removed from the network prior to applying FVS control. However, source nodes can denote external stimuli or boundary conditions the system is subject to; a different set of attractors may be available for each state of a source node. E.g., in the parameterized biological models we consider, source nodes provide positional information for the cells and affect the patterning behaviors cells are capable of.

Here we adapt the structural theory of Fiedler, Mochizuki et al. to networks in which source nodes are governed by second line in Eq. 4.3 (Fig. 4.1b). Since the source nodes are unaffected by other nodes, one additionally needs to lock the source nodes of the network in the trajectory specified by the attractor. We emphasize that the treatment of source nodes is not merely cosmetic, since the state of a source node can affect the dynamical attractors available to the system. E.g., steady states can merge, appear, or disappear depending on the presence or absence of an external stimulus represented by a source node [146, 151]. In summary, control of the source nodes and of the FVS of a network guarantees that we can guide it from any initial state to any of its dynamical

[1] The general task of designing a controller with an attractor as the target state in a nonlinear system is a difficult and unsolved problem that depends strongly on the functions F_i although several numerical algorithms for specific types of controllers have been proposed ([73, 113, 114]).

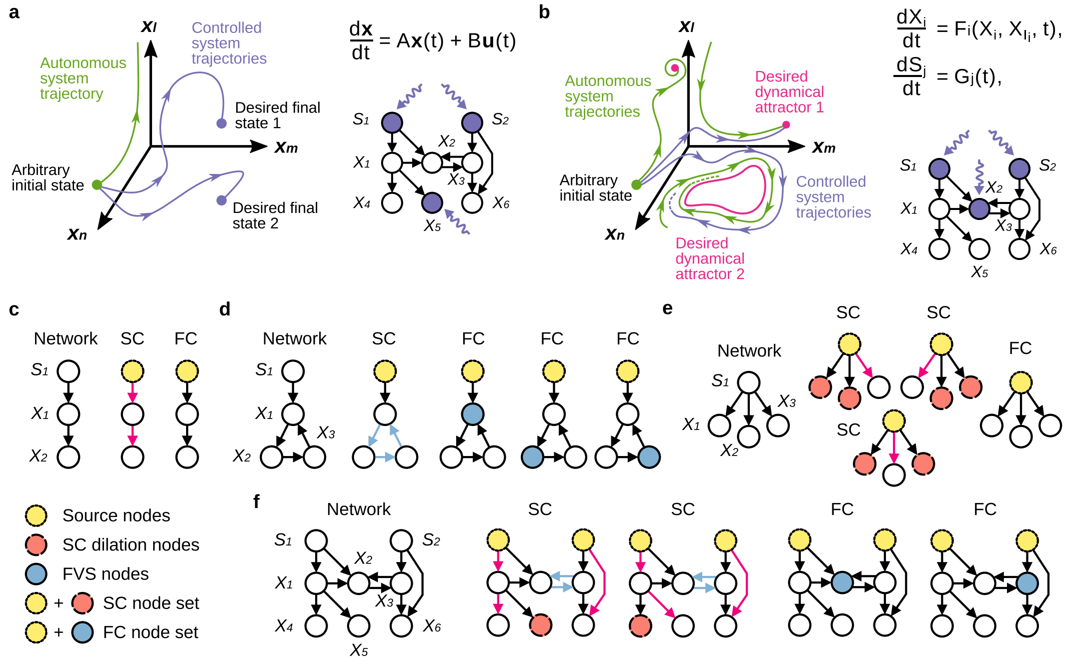

a
x_l
x_m
x_n
Autonomous system trajectory
Controlled system trajectories
Desired final state 1
Desired final state 2
Arbitrary initial state
$\frac{d\mathbf{x}}{dt} = A\mathbf{x}(t) + B\mathbf{u}(t)$
S_1
S_2
X_1
X_2
X_3
X_4
X_5
X_6
b
X_l
X_m
X_n
Autonomous system trajectories
Desired dynamical attractor 1
Arbitrary initial state
Controlled system trajectories
Desired dynamical attractor 2
$\frac{dX_i}{dt} = F_i(X_i, X_{I_i}, t),$
$\frac{dS_j}{dt} = G_j(t),$
c
Network
SC
FC
d
Network
SC
FC
FC
FC
e
Network
SC
SC
SC
FC
f
Network
SC
SC
FC
FC
Source nodes
SC dilation nodes
FVS nodes
+ SC node set
+ FC node set

Figure 4.1: Structure-based control methods. Structure-based control methods make conclusions about the dynamics of a system using solely the network structure. This figure repeats some panels from Fig. 1. (a) In structural controllability (SC) the objective is to drive the network from an arbitrary initial state to any desired final state by acting on the network with an external signal $\mathbf{u}(\mathbf{t})$. The dynamics are considered to be well-approximated by linear dynamics. (b) In feedback vertex set control (FC) the objective is to drive the network from an arbitrary initial state to any desired dynamical attractor (e.g. steady state) by overriding the state of certain nodes. (c-f) Structure-based control in simple networks. Control of the source nodes (yellow nodes with dotted outlines) is shared by SC and FC. SC additionally requires controlling certain dilation nodes (red nodes with dashed outlines) but requires no independent control of cycles. FC requires controlling all cycles by control of the feedback vertex set (FVS, blue nodes with solid outlines). The edges of the non-intersecting linear chains of nodes of SC are colored purple and the edges involved in a directed cycle are colored blue.

attractors (i.e., its natural long term dynamic behaviors) regardless of the specific form of the functions. In the following we refer to this attractor-based control method as feedback vertex set control (FC) (Fig. 4.1b), and to the group of nodes that need be manipulated FC as a FC node set.

To illustrate how the nodes that need to be manipulated in SC and FC can differ from each other, consider the example networks in 4.1. In a linear chain of nodes (4.1c, left) the only node that needs to be controlled in both frameworks is the source node S_1. For

4.1d, which consists of a source node connected to a cycle, SC requires controlling only the source node S_1 since the cycle is considered self-regulating (4.1d, middle), while FC additionally requires controlling any node X_i in the cycle, the feedback vertex set in this network (4.1d, right). 4.1e consists of a source node with three successor nodes; SC requires controlling two of the three successor nodes because of the dilation at the source node S_1, while for FC controlling S_1 is sufficient. In 4.1f we show a more complicated network with a cycle and several source and sink nodes, and two minimal node sets for SC and FC. These examples illustrate that the control of the source nodes is shared by full control in SC and attractor control in FC, and that their main difference is in the treatment of cycles, which require to be controlled in FC and do not require independent control in SC.

4.4 Feedback vertex set control and dynamic models of real systems

Validated dynamic models can be an excellent testing ground to assess control methods [6, 113, 131]. We compare the results of the two control methods for the gene regulatory network of the Drosophila segment polarity genes, for which several dynamic models exist [1, 19, 152]. The segment polarity genes, especially wingless (*wg*) and engrailed (*en*), are important determinants of embryonic pattern formation and contributors to embryonic development [19]. The wingless mRNA and protein are expressed in the cell that is anterior to the cell that expresses the engrailed and hedgehog (*hh*) mRNA and protein. All models consider a group of four subsequent cells as a repeating unit, and include intra-cellular and inter-cellular interactions.

Here we use two models for the gene regulatory network underlying the segmentation of the fruit fly (*Drosophila melanogaster*) during embryonic development: a differential equation (ODE) model by von Dassow et al. [19] (Fig. 4.2a) and a discrete (Boolean) model by Albert and Othmer [1] (Fig. 4.2b). Both models consider a group of four subsequent cells as a repeating unit, include intracellular and intercellular interactions among proteins and mRNAs, and both recapitulate the observed (wild type) stable pattern of gene expression (Fig. 4.2a-c).

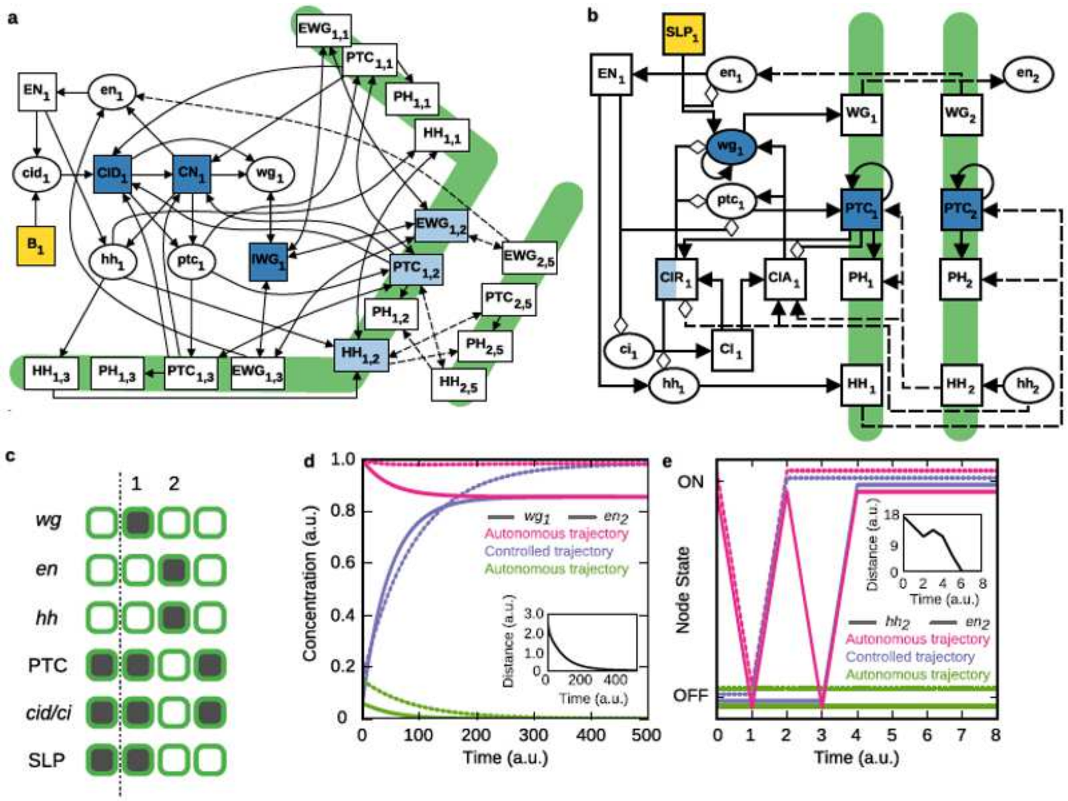


Figure 4.2: Control of the Drosophila segment polarity network models. (a, b) Networks corresponding to the differential equation model (panel a) and the discrete model (panel b). Each figure shows one cell of the four-cell parasegment together with the cell boundaries (thick green lines); the complete networks contain four cells in a symmetric completion of each figure. Elliptical nodes denote mRNAs and rectangular nodes denote proteins, which can be localized inside the cell or in the membrane (subscripts refer to the cell number and surface index). Intracellular interactions are drawn with solid lines and intercellular interactions are dashed. In panel b, positive edges are drawn with black arrowheads and negative edges with white diamonds. Yellow nodes are source nodes, blue nodes are FC nodes in every cell, and half white/half blue nodes are FC nodes in alternating cells. Dark blue nodes are sufficient for attractor control in the considered dynamic models. (c) Wild type segment polarity gene product expression pattern in a Drosphila parasegment. The parasegment boundary (dotted line) is between the *wg*-expressing cells (cell 1) and *en*-expressing cells (cell 2). (d, e) The dynamics of *wg* in the first cell (panel d, solid lines) and *hh* in the second cell (panel e, solid lines), and *en* in the second cell (dotted lines) in the models. Pink lines and green lines represent autonomous trajectories that start from different initial conditions and converge to different steady states (the wild type state and the unpatterned state, respectively). Blues lines represent the case when the system starts from the initial condition that autonomously evolves to the unpatterned state, but when applying FC, evolves into the wild type steady state. Insets: evolution of the norm of the difference between the desired attractor and the controlled state trajectory using FC.

4.4.1 Structure-based control of the von Dassow et al. differential equation model

The continuous model of von Dassow et al. represents each cell as a hexagon with six relevant cell-to-cell boundaries. It includes 136 nodes that represent mRNAs and proteins, among them 4 source nodes and 24 sink nodes, and 488 edges that represent transcriptional regulation, translation, and protein-protein interactions. Fig. 4.2a shows the network corresponding to the *wg*-expressing cell (cell 1) and three of its boundaries with the *en*-expressing cell 2. Additional nodes in the network include, *ptc* (patched), *ci* (cubitus interruptus), its proteins *CID* and *CN* (repressor fragment of *CID*), *IWG* (intracellular *WG* protein), *EWG* (extracellular *WG* protein), *PH* (complex of patched and hedgehog proteins), and *B*, a constitutive activator of *ci*. For each gene, the mRNA is written in lower case and the protein(s) are written in upper case. The nodes are characterized by continuous concentrations, whose rate of change is described by ordinary differential equations (ODE) involving Hill functions for gene regulation and mass action kinetics for protein-level processes, and using 48 kinetic parameters [153, 154]. von Dassow et al. have shown that the model can reproduce the essential feature of the wild type steady state: *wg*/*WG* are expressed anterior to the parasegment boundary (cell 1) and *en*/*EN*/*hh*/*HH* are expressed posterior to the parasegment boundary (cell 2) as shown in Fig. 4.2(c). The initial condition that yields this steady state for the most parameter sets is the so-called " crisp" initial condition. The differential equation system is solved using a custom code in Python and the odeint function with default parameter setting. We used the differential equations given in the appendix of [154]. *Ingeneue* can be found at http://rusty.fhl.washington.edu/ingeneue/papers/ papers.html. Relevant initial conditions and their corresponding steady states are shown in Appendix C.1.

The FC method predicts that one needs to control $N_{FC} = 52$ nodes (4 source nodes and 48 additional nodes) to lead any initial condition to converge to any original attractor of the model. There are multiple control sets with $N_{FC} = 52$; one of them consists of *B* (source node), *CI*, *CN*, *IWG*, *EWG* on every other side, *HH* on every other side, *PTC* on every other side in all four cells (shown in 4.3a). We perform simulations using two benchmark parameter sets to test this prediction. We use the second parameter set provided by the Ingeneue program to test the system's convergence to a steady state [153, 155]. The ODE system has at least two steady states with this parameter set. A nearly null initial condition leads to the unpatterned state (illustrated by the green lines

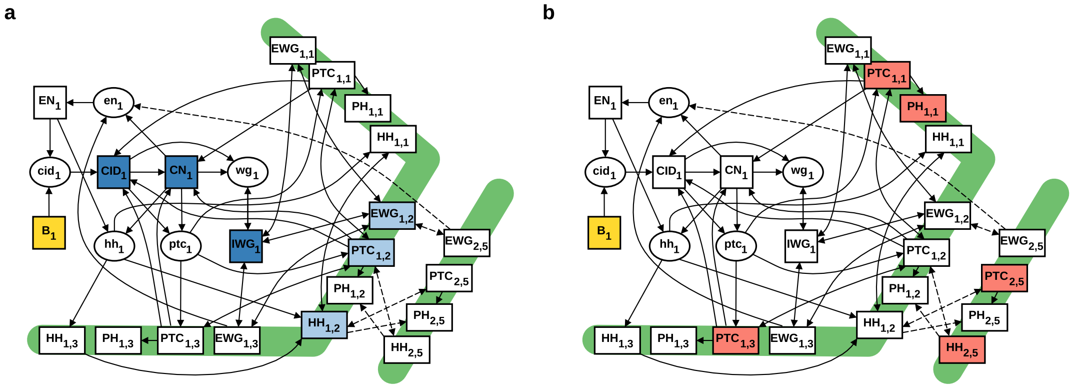


Figure 4.3: Control of the von Dassow et al. model of the Drosophila segment polarity network. The figure shows a cell of the four-cell parasegment together with three of its six boundaries (green lines). The complete network contains four cells in a symmetric completion of the figure. Elliptical nodes represent mRNAs and rectangular nodes are proteins. Intracellular interactions are drawn as solid lines and intercellular interactions are dashed. Yellow nodes are source nodes. (a) Blue nodes are FC nodes in every cell. Dark blue nodes are sufficient for attractor control in the considered dynamic models. (b) Red nodes are SC nodes in every cell.

in Fig. 4.2d in the main text). The crisp initial condition leads to the wild type pattern (see pink lines in Fig. 4.2d), which we choose as the desired steady state. If we start from the nearly null initial condition and maintain the concentrations of the nodes in the FC node set in the values they would have in the desired steady state, the system evolves into the desired steady state (see blue lines and inset of Fig. 4.2d). We obtained the same success of FC control when starting from 100 different random initial conditions (shown in 4.4a). We also obtained the same success using a reduced FC set (blue lines in 4.4b), which consists of *B*, *CID*, *CN*, *IWG* in every cell. In contrast, in the absence of control none of the trajectories converge to the wild type steady state (red lines in 4.4b).

We also numerically verified, using a different benchmark parameter set, namely the first parameter set provided by the Ingenue program, that FC control can also successfully drive any state to a limit cycle attractor (see 4.5a). This limit cycle attractor has the same expression pattern of *en*, *wg* and *hh* as the wild type steady state, thus we refer to it as the wild type limit cycle (illustrated in 4.5c). We also obtained the same success of driving any state to a limit cycle attractor using the same reduced Feedback vertex control shown in 4.5b.

SC control indicates multiple control sets with $N_{SC} = 24$ nodes. One possible combination is B_*, $PTC_{*,1}$, $PTC_{*,3}$, $PTC_{*,5}$, $HH_{*,5}$, $PH_{*,1}$, where $*$ represents all cells (shown

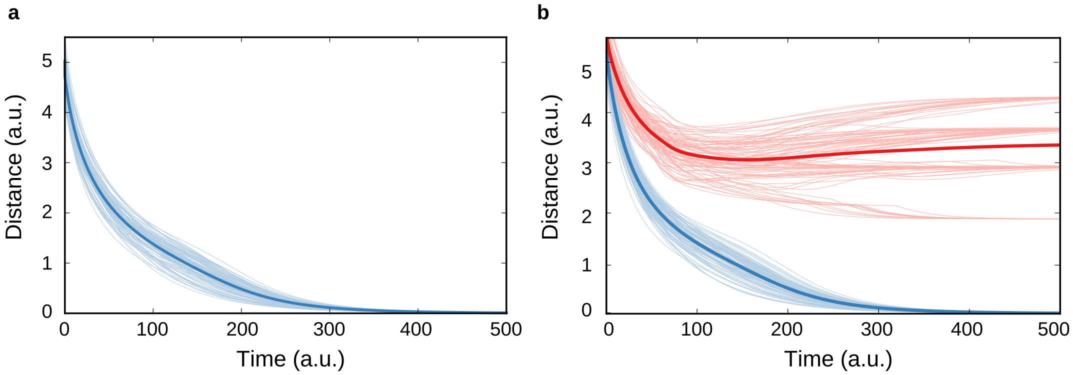


Figure 4.4: Effectiveness of the control of the Drosophila segment polarity differential equation model. (a) The thin light blue lines indicate the evolution of the norm of the difference between the desired wild type steady state and the controlled state trajectory using FC (blue symbols on 4.3a) for 100 randomly chosen initial conditions. (b) The thin light blue lines are the evolution of the norm of the difference between the wild type steady state and the controlled state trajectory using reduced FC (dark blue symbols on 4.3a) for 100 randomly chosen initial conditions. The thin red lines indicate the norm of the difference between the uncontrolled trajectory and the wild type steady state for 100 randomly chosen initial conditions. In all initial conditions the concentration of each quantity is chosen uniformly from the interval $[0,1]$. The thick blue (red) lines indicate the average of the relevant 100 realizations.

in 4.3b. Though SC predicts that less nodes need to be controlled, applying it requires a potentially complicated time-varying driver signal, which would need to be determined for each initial condition using, for example, minimum-energy control or optimal control [73, 156].

4.4.2 Structure-based control of the Albert & Othmer Boolean model

The Boolean model implements a few modifications in the network topology compared with the ODE network model, and considers only two cell-to-cell boundaries instead of six. There are 56 nodes and 144 edges in the network as shown in Fig. 4.2b. One difference compared with the von Dassow et al. model is the existence of three cubitus interruptus proteins: the main protein *CI*, and two derivatives with opposite function: *CIA*, which is a transcriptional activator, and *CIR*, a transcriptional repressor. There are four source nodes, representing the sloppy paired protein (*SLP*), which is known to have a sustained expression in two adjacent cells (cells 0 and 1 if the *wg*-expressing

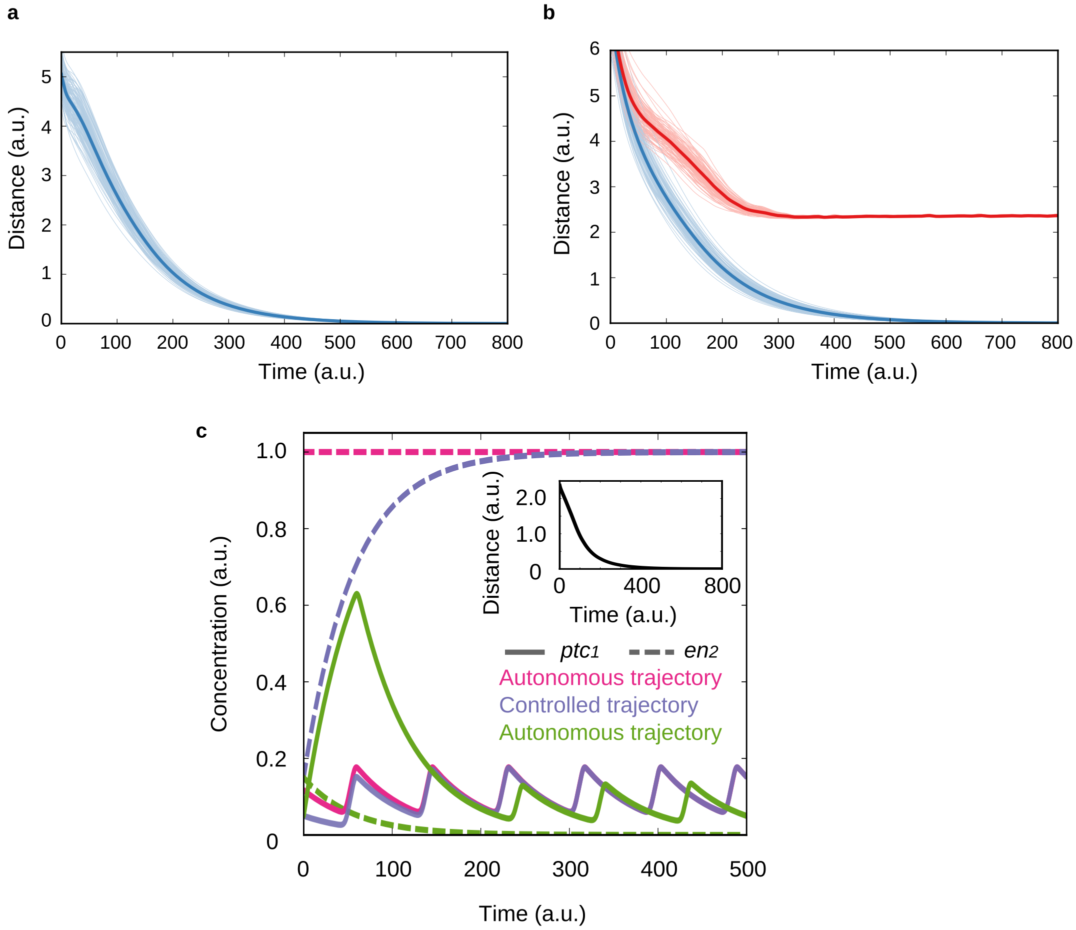


Figure 4.5: Control of the Drosophila segment polarity gene differential equation model for a different parameter set than that used to generate Fig. 4.2. (a) The thin light blue lines show the evolution of the norm of the difference between the wild type attractor and the controlled state trajectory using FC for 100 randomly chosen initial conditions. (b) The thin light blues lines are the evolution of the norm of the difference between the wild type attractor and the controlled state trajectory using reduced feedback FC for 100 randomly chosen initial conditions. The thin red lines are the evolution of the norm of the difference between the wild type attractor and uncontrolled trajectory using reduced FC for 100 randomly chosen initial conditions. In all initial conditions the concentration of each quantity is chosen uniformly from the interval [0,1]. The thick blue(red) line is the average of the 100 realizations. (c) The concentration of *ptc* in the first cell (solid lines) and en in the second cell (dashed lines) with respect to time. Pink lines and green lines represent autonomous trajectories that start from different initial conditions (a wild type initial condition and a nearly null, respectively) and converge to different attractors (the wild type limit cycle and an unpatterned limit cycle, respectively). Blue lines represent the case when the system starts from the nearly null initial condition, and after applying FC, evolves into the wild type limit cycle. Inset: evolution of the norm of the difference between the desired attractor and the controlled state trajectory using FC.

cell is considered cell 1) and is absent from the other two. There are ten steady states for this Boolean network model when considering the biologically relevant pattern of the source node states. Starting from the biologically known wild type initial condition, which consists of the expression (ON state) of SLP_0, SLP_1, wg_1, en_2, hh_2, ci_0, ci_1, ci_3, ptc_0, ptc_1, ptc_3, the model converges into the biologically known wild type steady state illustrated on Fig. 4.2c.

Specifically, the wild type steady state of the Albert & Othmer model consists of the expression of

$$SLP_0, SLP_1, wg_1, WG_1, en_2, EN_2, hh_2, HH_2,$$
$$ci_0, ci_1, ci_3, CI_0, CI_1, CI_3, CIA_1, CIA_3, CIR_0,$$
$$ptc_1, ptc_3, PTC_0, PTC_1, PTC_3, PH_1, PH_3.$$

Analytical solution reported in [154] indicated that the states of the *wg* and *PTC* nodes, each of which has a positive auto-regulatory loop, determine the steady state for the given source node (*SLP*) configuration [1]. For example, any initial condition with no *wg* expression leads to an unpatterned steady state wherein *ptc*, *ci*, *CI* and *CIR* are expressed in each cell, and the rest of the nodes are not expressed in any cell.

The FC method predicts that $N_{FC} = 14$ nodes need to be controlled, including the 4 source nodes (*SLP*), the 8 self-sustaining nodes (all *wg* and *PTC*), and 2 additional nodes (with one possibility being CIR_1 and CIR_3). Since the FC set contains all *wg* and PTC nodes, which were shown to determine the steady states under the indicated source node states, we can conclude that controlling the nodes in the FC set is enough to drive any initial condition to the desired steady state in the Albert & Othmer model. The simulation result is consistent with the theoretical result, as shown in Fig. 4.2e. The wild type initial condition leads to the wild type steady state (pink lines). The null initial condition used in the Boolean model is that all the nodes are in the OFF state; the resulting steady state is the unpatterned steady state (green lines). The controlled trajectory with FC is shown in blue lines. We obtained the same success of FC control when starting from 100 different random initial conditions, as shown in 4.6a. Moreover, the 12 nodes consisting of *SLP*, *wg* and *PTC* in each cell (which we refer to as the reduced FC set) are enough to drive all the random initial conditions to the desired steady state in this particular model, as shown in 4.6b.

SC control predicts that we only need to control the four source nodes (*SLP*), as

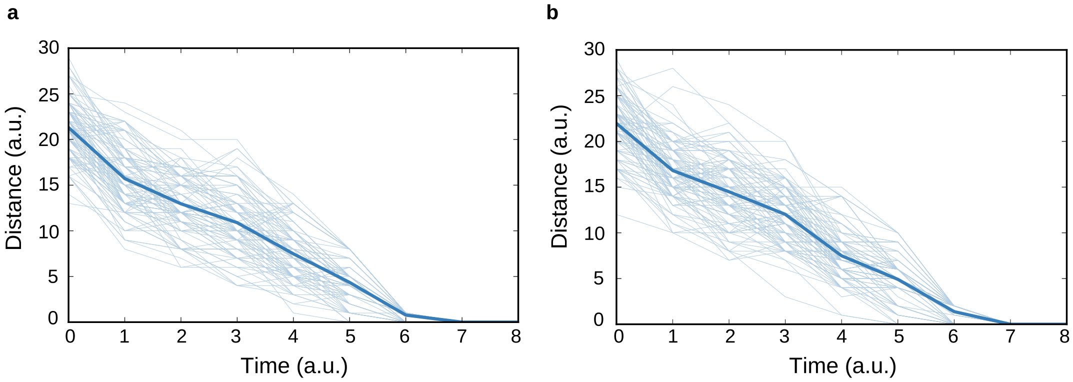


Figure 4.6: Control of the Boolean model of the Drosophila segment polarity genes. The light blue thin lines show the evolution of the norm of the difference between the wild type steady state and the controlled state trajectory using feedback vertex set control (FC) for 100 randomly chosen initial conditions, in which the concentration of each quantity is chosen between ON and OFF with equal odds. The thick blue line is the average of the 100 realizations. (a) Control using the feedback vertex set (b) Control using the reduced feedback vertex set.

the network can be covered by four branches and one loop. Relevant to this, Albert & Othmer studied three scenarios of fixed states of the source nodes. If the source nodes are locked into their respective states in the wild type steady state (two ON and two OFF), there are six reachable attractors, one of which is the wild type steady state. If all source nodes are locked into the OFF state, there are seven attractors, but none of them is the wild type steady state. If all source nodes are locked into the ON state, the unpatterned state is the only attractor. These results suggest that the correct expression of the source nodes is necessary, but not sufficient for attractor control of the system. Indeed, SC can make no such guarantee, since for general nonlinear systems it only provides sufficient conditions for local controllability around a steady state or a system trajectory.

For a simplified, single-cell version of the Albert & Othmer model, Gates and Rocha showed that the SC node set is sufficient for attractor control, but does not fully control this system [131]. Thus, a control method such as [157, 158] seems to be required for correctly predicting full control node sets in Boolean models.

4.4.3 Discussion

Using FC on these network models, we find $N_{FC} = 52$ (14) for the ODE (discrete) model (Fig. 4.2a-c). Both model networks have a large SCC, and thus, a significant FVS contribution to the FC node set. In FC, locking the FC nodes into their trajectory in the wild type attractor successfully steers the system to the wild type attractor (Fig. 4.2d-e). Thus, FC gives a control intervention that is directly applicable to dynamic models and that is directly linked to their long-term behavior.

FC gives a sufficiency condition about the ensemble of all models with a given network structure, and consequently, a subset of the FC node set can often be sufficient for a given model and an attractor of interest (i.e. FC provides an upper limit for the size of the control node set). For the fruit fly gene regulatory models we show that 16 (12) nodes are sufficient for the continuous (discrete) model, respectively, which is a 66% (14%) reduction (Fig. 4.2a-c). Similar results were obtained in [141], who found that 5 nodes (out of 7 in the FVS) are sufficient for attractor-based control in a model of the mammalian circadian rhythm. The generality of these findings is supported by a recently developed control method in which controlling a subset of the cycles (and, thus, a subset of the FVS) in Boolean dynamic models was proven to be sufficient for attractor control ([6]). This shows that FC provides a benchmark of attractor control node sets that are model independent, as well as an upper limit to model dependent control sets.

Chapter 5 | Conclusions and Future Works

In my dissertation, I made progress in several projects related to control problems in intra-cellular systems, where a variety of proteins and molecules interact in a diverse way and complete quantitative information about the dynamics is often unavailable. In spite of these difficulties, one way to proceed is to consider control strategies on logical network models such as Boolean network models, which can be constructed through currently available experimental data. Chapter 2 and Chapter 3 consider complementary damage mitigation problems in Boolean network models. In Chapter 2, I designed compensatory interactions to try to immediately stabilize the system under a permanent damage. In Chapter 3, I applied a heuristic algorithm to solve the target control problem in Boolean network models, which can be adapted to design strategies to mitigate a long-term effect of a permanent or temporary damage. Another way to proceed is to consider structure-based control methods for ODE models, where we assume we are agnostic to certain details of the dynamics. In Chapter 4, we adapted and implemented the feedback-vertex control, which is designed for attractor control in non-linear systems, in a real biological system. We also compared it with another popular method called structural controllability, which is designed to achieve full control in linear time-invariant systems. We illustrated the dramatic difference in the predictions given by different methodology and cautioned against inappropriate use of one method in a different background. In the following, I discuss possible future works and selectively provide some preliminary insight or results.

5.1 Future works about control in Boolean networks

The work in Chapter 3 suggests interesting questions as future research directions. First, though evaluating DOI of a node (set) is computationally hard, a better estimation of

DOI (than LDOI) is desirable and can be used to reduce the size of the solution given by our current target control algorithm. Second, the requirement that the solution works for all initial conditions in the setup of the target control problem gives robust solutions, however it may be conservative for biological systems in certain applications, especially if one is certain about the relevant initial condition subspace. A semi-structural approach without doing dynamical simulations to solve the target control problem starting from a subspace of initial conditions is also desirable.

Third, we transformed the target control problem into a planning search problem. It is established that such a planning search problem can be solved in both a forward propagation and a backward propagation approach, or even a mixed approach [123]. It will be an interesting future work to test wether such techniques can improve the efficiency of the algorithm. In fact, an ESM can be constructed through a backward search from an output node [128], which can be adapted to find solutions of the target control problem of a single output. If we treat the output node as the root in the backward search, the nodes found in the ESM in each search depth can serve as a solution. Similar truncation techniques need to be applied to deal with inconsistent feed-forward or feedback loops. This algorithm can be generalized to solve the target control problem of a target set by simultaneously updating each target node.

Last but not least, we assume an existing discrete dynamical model for our discussion of damage mitigation and target control problem. However, we still bear the risk of having an incomplete model. It will be interesting to see control strategies that can be robust to missing information.

5.2 Future works about structure-based control

As structure-based methods and feedback-vertex control were just recently established for non-linear systems, there are still a lot of interesting questions to be explored, as discussed in the following.

5.2.1 Domain of Influence and target control in non-linear system

The essential idea of feedback vertex control, that controlling the node states of the feedback vertex set to adopt their states in the natural attractor will eventually drive the system towards that natural attractor, remind us the concept of a driver set of a stable motif in Boolean network model. As we have seen in Chapter 3, the domain of influence of this driver node set contains the stable motif.. Also one would wonder how one might approach the target control problem in networked systems with non-linear dynamics. All these prompt us to extend the concept of domain of influence and its use in target control from logical models (studied in Chapter 3) to non-linear ordinary differential equations systems (considered in Chapter 4).

Let us frame the questions that we want to solve. Consider a system whose dynamics is described by equation 4.3;we do not know the specific functional form and parameters. We want to know the effect of fixing the state of a node to the value corresponding to one of the system′s natural fixed-point attractors. (This can be generalized to include complex attractors.) Specifically, we want to know which other nodes will be guaranteed to stabilize. Also regardless of the specific functional forms and parameters, we want to know which nodes we need to control in order to drive the nodes of a target set to stabilize in the states corresponding to one of the system′s natural fixed-points.

However, we note that there are key differences between the above scenario and the target control problem of Chapter 3 in addition to the difference in the dynamics of the system, i.e., Boolean network model versus non-linear ordinary differential equation model. First, when we defined the concept of domain of influence in chapter 3, we wee considering a specific Boolean dynamical model with given Boolean rules instead of a class of possible Boolean network models sharing the same network topology. In contrast, continuing along the line of finding structure-based methods, we consider non-linear models without any functional form or parameters specified. We are only certain about the underlying network topology (regulatory relationships) and the general requirement of a dissipative non-linear dynamical system. Second, for Boolean network systems, we have considered the domain of influence of a specific node state, regardless of whether it is possible to be fixed in that state. For non-linear systems, we restrict ourselves to consider controlling a node to be fixing the node to its state in one of the system′s natural fixed-points, as in one hand we do not know which fixed node value to consider (since we do not know the specific functional form), and in the other hand, fixing the node in other values is potentially inconsistent with the given differential equation for that node. Without changing the function, it is not possible to stabilize a node in other value not permitted in its natural attractor. In fact, the inconsistent node state interventions defined in Chapter 3 are representatives of the situation that the specific node state does not appear in any natural fixed point. We remarked there that it is practically hard to apply these interventions as they act counter to the system′s natural dynamics; applying them is essentially changing the regulatory function ofthat node.

Consider a node of the network with k regulators. Since we do not know the specific dynamics involved, the worst scenario is that all its k regulators are necessary to control this node. That is, the value of this child node cannot be inferred unless the values of all of its regulators are known. Mathematically, in the dynamical equation for node i $dX_i/dt = F_i(X_i, X_{I_i}, t)$, the long term behavior of X_i can be explicitly determined for all functional forms F_i only if the trajectory of all of its regulators X_{I_i} are known. This is equivalent to treating the rule for multiple regulators always as an AND rule between them, as this is the potential worst scenario. Thus we can adapt the algorithm of finding the LDOI of a node state in Boolean networks to find the DOI of a node (variable) here. Since all regulators of the same node have the same relationship, we can directly implement the BFS on the network instead of the expanded network in the Boolean network case. The rule to include a node during the search process is straightforward, we only include a node if all of its parents are included. We follow the same convention to include the controlled node as in Chapter 3: we only include the controlled node if we visited this node again during the search process. The adapted pseudocode is written in Algorithm 3.

Similarly, we can generalize the properties established in Chapter 3 for the LDOI in a Boolean network to the current setting. The first property is that $S_i \subseteq S_j$ implies $\mathscr{D}(S_i) \subseteq \mathscr{D}(S_j)$, where S_i and S_j are two sets of nodes and $\mathscr{D}$ denotes domain of influence. The proof is trivial as the DOI can be found on the usual directed network. With respect to the second property in Chapter 3, it is not straightforward to generalize it without defining a stable motif in an ODE system. However, the feedback-vertex control strategy can be rephrased in the language of DOI, that is, the DOI of a set composed by the FVS and all source nodes is the entire network, which obviously includes the FVS and all the source nodes. Considering a set of nodes S_i, we propose that $S_i \subseteq \mathscr{D}(S_i)$ implies that S_i is (contains) a FVS of the sub-network spanned by the nodes in $\mathscr{D}(S_i)$, though S_i may not be the minimal FVS is. Although the proof is purely topological (it can be seen from the search method), this property has its corresponding dynamical implications as the DOI has its dynamical meaning.

With respect to the target control problem, we can still use the GRASP algorithm established in chapter 3. The only difference is that we are using the new DOI measure

Algorithm 3 Algorithm of calculating DOI in non-linear ODE system

```
procedure DOI($G, source$)
    $queue \leftarrow Queue(source)$
    $visited \leftarrow Set()$
    $visited\_count \leftarrow HashMap()$
    while $queue$ $is$ $not$ $empty$ do
        $node \leftarrow queue.pop()$
        if $node \notin visited$ then
            $visited \leftarrow visited \cup \{node\}$
            for $child$ $in$ $Children(G, node)$ do
                if $visited\_count[child] = G.InDegree(child) - 1$ then
                    $queue.append(child)$
                else
                    $visited\_count[child] += 1$
                end if
            end for
        else if $node \in source$ then
            $visited \leftarrow visited \cup \{node\}$
        end if
    end while
    return $visited$
end procedure
```

on the original network instead of the LDOI/DOI measure on the expanded network as in chapter 3. That is, just replace the *G_expanded* by *G* and replace LDOI by the new DOI in Algorithm 1 and 2 in chapter 3. We do not repeat the pseudocode here.

5.2.2 Dependence of the FVS size on network topological features

Evaluation of how hard to control a system is useful information for real applications of control theory. The evaluation can be done from different perspectives, including control energy [73]. In the framework of FVS control of non-linear systems, the number of nodes that need to be controlled is one unquestionably important factor, which boils down to the number of source nodes and the size of FVS. We analyzed why different real networks have different FVS size based on randomization of strongly connected component and small cycles [115]. However, we lack a deeper understanding of how the FVS size depends on or correlates with the other topological features of the network.

As machine learning and deep learning techniques are thriving, it might be possible to build a supervised machine learning model that can predict the minimum FVS size of a network based on network topological features. Such a model would have many applications.. First, this would help us identify essential topological features that affect the FVS size and its dynamical implications such as the difficulty of applying FVS control. Second, minimum FVS problem is known to be a NP hard problem, accurately predicting the minimum size of FVS through other easy-to-calculate topological features would inspire efficient heuristic algorithms to approximate FVS. Third, from the evolutionary perspective, the topology of real networks is shaped to serve their function and desired dynamical behaviors (e.g. attractors) . Thus understanding the relationship between FVS size and topological features might help us understand the evolutionary mechanisms shaping the topology of certain real networks.

Even more ambitiously, we wonder whether a neural network or a deep learning method can predict the FVS (size) of a network if we provide the network structure as the input. If such a predictive model is successful, analyzing the "neurons" constructed by the neural network will be informative to see which topological features play an important role in determining FVS (size).

Here we try to approach this problem through studying simple network ensembles that we know how to generate, for example, random network and small networks. We show some preliminary results as below.

We generate random directed network ensemble by the N-M model, where N is network size and M is the number of edges. The network size ranges from 50 to 1000 and network degree ranges from 1 to 15. First, notice, the size of the FVS of a network should be the sum of the size of the FVS of each strongly-connected component (SCC) as there are no feedback loop between each individual strongly connected component otherwise they will be one strongly connected component. Thus we study how the FVS size of each SCC depends on the topological features of each SCC. All SCC of a network can be obtained in $O(|V|+|E|)$. As shown in Fig 5.1, we found out the SCC size and node in-degree of the SCC size largely determine the FVS size of a SCC, i.e. different FVS size is well separated in different regions in the space spanned by the SCC size and the SCC degree. We also confirmed this by a supervised learning model such as linear regression and support vector machine. To be specific, a support vector regression model can obtain an root mean square error (RMSE) of 2.756 and mean absolute error (MAE) of 2.220 for in-sample data. A comparison between the predicted value and the real FVS

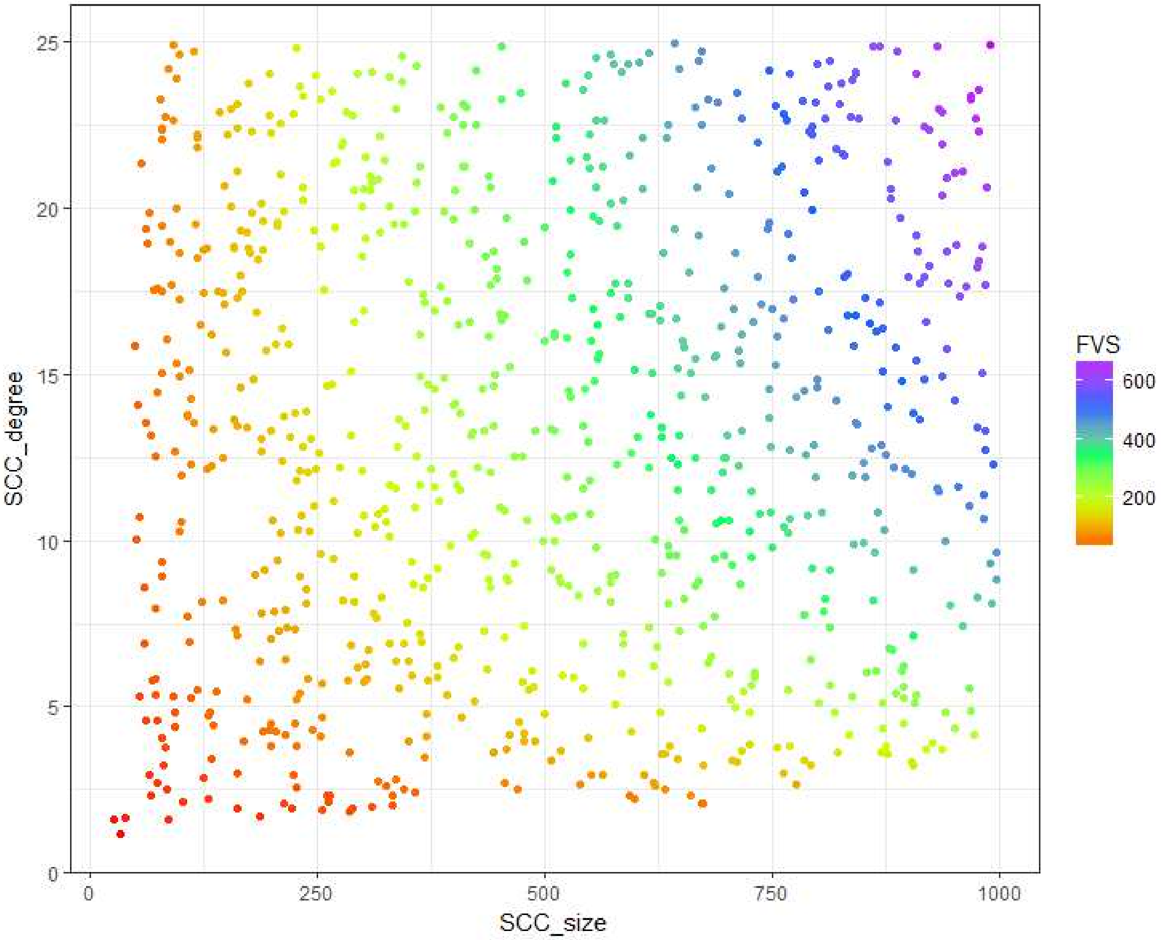


Figure 5.1: Dependency of FVS size of a SCC on the SCC size and the node degree of the SCC.

size is shown in Fig. 5.2We also use the SVM model to obtain a 10 fold cross-validation RMSE as 2.861. All these evidences suggest that the SCC size and the SCC degree almost determine the size of the FVS for a random ensemble and thus an analytical form based on probability theory and combinatorics should be expected. One can also use Eureqa to extract the functional form of this relationship, however, this is still based on statistical inference and lacks the graphical interpretation.

As we expect, the SCC size and the SCC degree should not determine the FVS size for other ensemble. Indeed, this is the case for small-world network. We identify two other features that can be used to infer the FVS size. The first one is the mean path difference between any two nodes in the network, which calculates how asymmetric the network is. Mathematically, it is defined as$A =< |d_{ij} - d_{ji}|>$. Another useful feature is

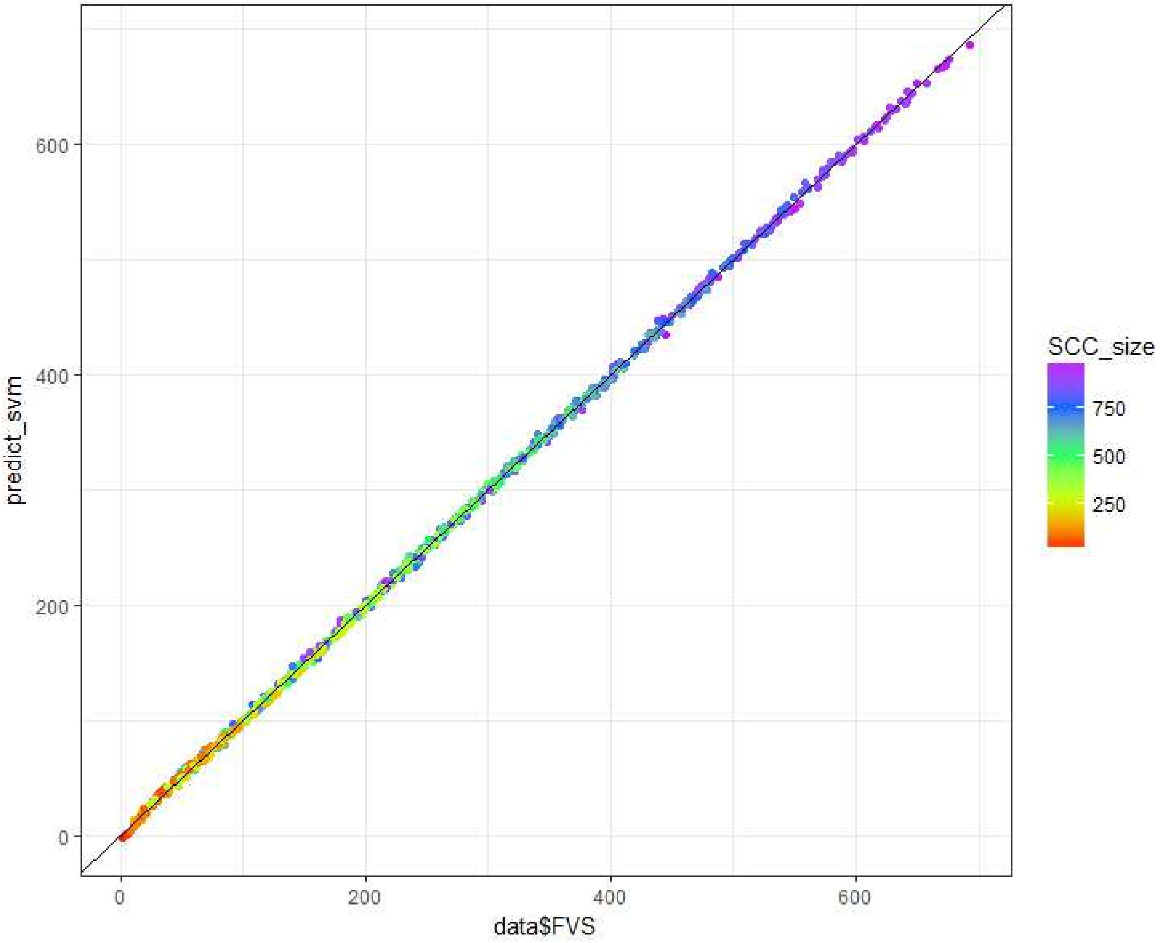


Figure 5.2: Prediction of the SVM model vesus the original model.

the standard deviation of all the page rank score of the SCC.

5.2.3 Evaluation of a node′s dynamical importance and inference from network topological features

We are not only interested in global dynamical properties, but also the dynamical importance of each node. DOI of each single node potentially provides some insight into this. However, one can readily see that the DOI of a single node may small for networks that are not sparse (i.e. networks with a slightly larger average degree). In order to evaluate the importance of a node i, we can define the following function: the average size of the DOI of a node set that contains node i plus n-1 other nodes, incrementing n from 1 to the size of FVS. The calculation can be down by Monte Carlo simulation and the hope is we can efficiently determine these quantities numerically. This new measure will be more informative in a relatively dense network, which can serve as the basis to determine the dynamical importance of each node. An interesting follow-up work will be studying how the dynamical importance of each node depends on or correlates with its local topological feature, especially centrality measures. Again a machine learning model or a deep learning model would be insightful.

5.2.4 Evaluation of a node′s dynamical importance in a temporal network

The topology of real networks may change with time. Although the topology of a biological network is rather robust for a given system, a biological network viewed in a larger time scale, or a technology network or social network may change as time proceeds. Thus it is also insightful to evaluate the dynamical importance of existing nodes or edges through the changes brought by removing them. Also it is interesting to evaluate the change of the dynamical importance of existing nodes when new nodes or edges are introduced to the system. Such study has been done for structural controllability [24] and an analog would be desired for the framework of FVS control.

5.2.5 Other interesting topics

Structure-based control method is a model-independent method: the control strategy does not depend on the specific dynamics and parameters of the model. Thus it will provide a necessary solution for attractor (or target) control and set an upper bound for the size of solution. It would be interesting to see how the model-independent method and model-dependent methods compare in different dynamical systems. Along this line, an interesting topic is to understand how many and which nodes do not need to be controlled if we add extra restriction on the dynamics or initial conditions. For example, the dependence of one bio-molecule on another bio-molecule is often monotonous in biological systems (the partial derivative does not change sign for feasible variable domains). We would wonder how the structure-based solutions change if we add this condition for models with non-linear dynamics. Also, simply put, structural controllability tells us to control source nodes, extra sink nodes and internal dilatations [130], while FVS control tells us to control source nodes and FVS [77]. It will be interesting to see a similar simple network algorithm to identify control set and its graphical interpretation for a “mixed” system, that is, we know for sure that the functional dependence of some nodes on their regulators are linear time-invariant and the remaining are assumed to be generally non-linear. This can be treated as another kind of restriction on the functional forms of the dynamics. Such dynamical models are prevalent in biological systems. For example, the ODE model of the segment polarity network falls in this category. [19]

5.3 summary

In summary, my dissertation demonstrates how computational network algorithms and dynamical modeling in physics interplay to provide insight to biological systems, especially those involved in complex diseases. Due to the variety of the participating bio-molecules and interaction mechanisms, it is challenging to build dynamical models and design control strategies with incomplete information for the system. The methods developed in my research add new tools to the computational toolbox of system biology. We apply these tools to several established network models and the results are consistent with previous results. Also we have made new predictions to be confirmed by experimentalists. This research also opens new research directions, where computer science and dynamical modeling are combined to solve practical problems in complex disease and systems biology.

Appendix A
Appendix of Chapter Two

A.1 Additional simplification of T-LGL leukemia network

When the sink node Apoptosis is activated, the cell is going to die. Zhang *et al.* chose to represent cell death by a state in which Apoptosis is ON and all the other nodes are OFF and implemented it by adding to every node′s Boolean function the clause “AND (NOT Apoptosis)” [2]. Here for simplicity we do not use this additional clause; this is equivalent with considering any steady state that includes Apoptosis=ON as a normal steady state. In the reduced network (Fig. 2.5) a small motif consisting of TCR and CTLA4 is isolated from the main part of the network. Since the small motif does not influence the apoptotic decision, we ignore it in the analysis. An auxiliary node P2 in [2] is removed and we incorporate the effect in the Boolean rule of IFNG. Also, if the cell is already dead, node knockout and constitutive expression have no biological meaning. However, the activation of Apoptosis requires the node Caspase to be ON first. Thus, we delete the node Apoptosis and consider that Caspase is determining the state of the cell.

A.2 Modifications for node Ceramide after knockout of Fas in the T-LGL leukemia network:

All the solution have the same format: “Ceramide = $\cdots$ OR New Rule”, where $\cdots$ stands for the original rule.
Ceramide=$\cdots$ OR BID

Ceramide=$\cdots$ OR Caspase
Ceramide=$\cdots$ OR DISC
Ceramide=$\cdots$ OR NOT FLIP
Ceramide=$\cdots$ OR NOT GPCR
Ceramide=$\cdots$ OR NOT IAP
Ceramide=$\cdots$ OR NOT MCL1
Ceramide=$\cdots$ OR NOT SMAD
Ceramide=$\cdots$ OR NOT sFas

A.3 Classification of double knockout pairs in the T-LGL leukemia network example.

All ten node pairs (A, B) are list in the first column. The node to be repaired after knocking out A, B, both A and B are listed in the second to fourth column respectively, where $\emptyset$ means that no node need to be repaired. The class and subclass index as in Table 1 is listed in the fifth column.

Node pair	S_A	S_B	S_{AB}	class
BID,Caspase	IAP	$\emptyset$	IAP	4b
BID,Ceramide	IAP	S1P	IAP,S1P	6b
BID,DISC	IAP	MCL1,FLIP	IAP,Caspase, MCL1,FLIP	6c
BID,Fas	IAP	Ceramide	IAP,Ceramide	6b
Caspase,Ceramide	$\emptyset$	S1P	S1P	4b
Caspase,DISC	$\emptyset$	MCL1,FLIP	MCL1,FLIP	4b
Caspase,Fas	$\emptyset$	Ceramide	Ceramide	4b
Ceramide,DISC	S1P	MCL1,FLIP	MCL1,S1P,FLIP	6b
Ceramide,Fas	S1P	Ceramide	S1P,DISC,	6d
DISC,Fas	MCL1, FLIP	Ceramide	MCL1,FLIP, Ceramide,	6b

A.4 Simple solutions after knocking out GSK3β in the healthy steady state of EMT network

All the solutions have similar format as above, where $\cdots$ stands for the original rule for that node. Solutions in normal text format are simple solutions compatible with disease steady state after knockout GSK3β ïĂăïĂăin the healthy steady state, i.e., solutions in *italics* are simple solutions incompatible with disease steady state after knockout GSK3β ïĂăïĂăin the healthy steady state.

Modifications for node AKT:

AKT=$\cdots$ AND GLI

AKT=$\cdots$ AND MEK

AKT=$\cdots$ AND NOTCH

AKT=$\cdots$ AND SNAI1

AKT=$\cdots$ AND TGFβR

AKT=$\cdots$ AND TWIST1

AKT=$\cdots$ AND ZEB1

AKT=$\cdots$ AND ZEB2

AKT=$\cdots$ AND NOT β-catenin_memb

AKT=$\cdots$ AND NOT E-cadherin

AKT=$\cdots$ AND NOT miR200

AKT $= \cdots$*AND Dest_compl*

AKT $= \cdots$*AND NOT AXIN2*

AKT $= \cdots$*AND NOT SNAI2*

Modifications for node MEK:

MEK= $\cdots$ AND AKT

MEK= $\cdots$ AND GLI

MEK= $\cdots$ AND NOTCH

MEK= $\cdots$ AND SMAD

MEK= $\cdots$ AND TGFβR

MEK= $\cdots$ AND TWIST1

MEK= $\cdots$ AND ZEB1

MEK= $\cdots$ AND ZEB2

MEK= $\cdots$ AND NOT β-catenin_memb

MEK= ··· AND NOT E-cadherin

MEK= ··· AND NOT miR200

MEK = ···AND Dest_compl

MEK = ···AND NOT AXIN2

MEK = ···AND NOT SNAI2

Modifications for node SNAI1:

SNAI1= ··· AND TWIST1

SNAI1= ··· AND ZEB1

SNAI1= ··· AND ZEB2

SNAI1= ··· AND NOT β-catenin_memb

SNAI1= ··· AND NOT E-cadherin

SNAI1= ··· AND NOT miR200

SNAI1 = ···AND Dest_compl

SNAI1 = ···AND SOS/GRB2

SNAI1 = ···AND NOT AXIN2

SNAI1 = ···AND NOT SNAI2

Modifications for node NOTCH:

NOTCH= ··· AND AKT

NOTCH= ··· AND GLI

NOTCH= ··· AND MEK

NOTCH= ··· AND SNAI1

NOTCH= ··· AND TGFβR

NOTCH= ··· AND TWIST1

NOTCH= ··· AND ZEB1

NOTCH= ··· AND ZEB2

NOTCH= ··· AND NOT β-catenin_memb

NOTCH= ··· AND NOT E-cadherin

NOTCH= ··· AND NOT miR200

NOTCH = ···AND Dest_compl

NOTCH = ···AND NOT AXIN2

NOTCH = ···AND NOT SNAI2

Appendix B
Appendix of Chapter Three

B.1 Algorithms

B.1.1 Algorithm for calculating LDOI

In Algorithm 4, we present the pseudocode for calculating the LDOI of a set of nodes, referred to as *source*, on the expanded network $G\_expanded$. $Negation(node)$ in line 8 calculates the negation of a non-composite *node* on the expanded network. The $Children(G\_expanded, node)$ function in line 11 returns all direct neighbors that *node* points to on the expanded network. The *InDegree* function in line 14 calculates the in-degree of a node on the expanded network (number of incoming edges or number of parent nodes).

B.1.2 Algorithm for local search of solution reduction

Pseudocode to obtain a reduced solution from a candidate solution, referred to as $solution$, in the target control problem on the expanded network *G_expanded* with *Target* as the target, is shown in Algorithm 5.

Algorithm 4 Algorithm of calculating LDOI

```
procedure LDOI($G\_expanded, source$)
    $queue \leftarrow Queue(source)$
    $visited \leftarrow Set()$
    $visited\_list \leftarrow List()$
    $cpnode\_count \leftarrow HashMap()$
    while $queue$ $is$ $not$ $empty$ do
        $node \leftarrow queue.pop()$
        if $node \notin visited$ AND $Negation(node) \notin visited$ then
            $visited \leftarrow visited \cup \{node\}$
            $visited\_list.append(node)$
            for $child$ $in$ $Children(G\_expanded, node)$ do
                if $child$ $is$ $not$ $composite$ $node$ then
                    $queue.append(child)$
                else if $cpnode\_count[child] = G\_expanded.InDegree(child) - 1$ then
                    $queue.append(child)$
                else
                    $cpnode\_count[child] += 1$
                end if
            end for
        else if $node \in source$ then
            $visited\_list.append(node)$
        end if
    end while
    return $Set(visited\_list[length(source):])$
end procedure
```

B.2 Definition and properties of the LDOI

B.2.1 Mathematical description of the level-order preserving updating regime

Mathematically, let s^{ex} be the node in the expanded network whose logical domain of influence we are seeking to determine, and s the corresponding node in the original network. Let $d(s,i)$ represent the distance (shortest path length) from s^{ex} to n_i or $\sim n_i$ on the expanded network. $d(s,s) = 0$ when there is no feedback loop from node s to node s, otherwise, $d(s,s) = l_s$, which is the length of the shortest feedback loop passing s. Let t_i be the first time node i is updated. In a level order preserving updating regime, $t_i \leq t_j$ if

Algorithm 5 Algorithm for local search of solution reduction

```
procedure LocalSearch($G\_expanded, Target, solution$)
    $reduced\_solution \leftarrow solution$
    if $solution.length() \leq 1$ then
        return $reduced\_solution$
    end if
    for node in random order of solution do
        $temp\_solution \leftarrow reduced\_solution$
        $temp\_solution.remove(node)$
        if $Target \subset LDOI(solution)$ then
            $reduced\_solution \leftarrow temp\_solution$
        end if
    end for
    return $reduced\_solution$
end procedure
```

and only if $d(s,i) \leq d(s,j)$.

B.2.2 Mathematical proof of the second property of LDOI

We prove the property by contradiction. The property is: if the LDOI of a node state set S contains itself, then the LDOI of S contains a stable motif. The contradiction statement is that $S \subset \mathscr{LD}(S)$ implies that $\mathscr{LD}(S)$ does not contain a stable motif. $S \subset \mathscr{LD}(S)$ implies that there exists a level order preserving updating regime in which node states in S will not change when these nodes are updated and are not externally controlled (fixed). This level order preserving updating regime can be constructed by periodically repeating the first round of updating. In such an updating regime, the system will evolve into a (partial) fixed point as node states in S does not change. However, in the contradiction statement, $\mathscr{LD}(S)$ does not contain a stable motif. Thus no subset of nodes belonging to S can be a (partial) fixed point since a stable motif corresponds to a (partial) fixed point (proved in [54]), where we find the contradiction. Thus the contradiction statement is false and the original property holds.

B.3 Additional information on the biological examples

B.3.1 EMT network

Detailed information about the EMT network can be found in [5, 109]. The Boolean regulatory functions are presented in the Supplemental Table 1 of [109]. The two steady states of the EMT network model are shown in Supplemental Table 2 of [109]. Previous knockout intervention results are illustrated in Figure 1 of [109].

B.3.2 Breast cancer network

Detailed information about the breast cancer network, including the Boolean regulatory functions of each node, can be found in the section “Regulatory functions in the model" of the additional file in [110]. Relevant initial conditions that lead to cancerous steady states, which have slight variations among themselves, are stated in the section “Initial or externally controlled states" of the additional file in [110]. For our purpose of target control, we chose the initial condition that leads to the most cancerous steady state and obtained a (relatively large) reduced network after plugging in these initial conditions. Specifically, among the 12 source nodes, IGFR1_T, PBX1 and ER are fixed to be ON. HER2, HER3_T, PTEN, SGK1_T, PIM, PDK1 and mTORC2 are fixed to be OFF. BIM_T is assumed to be OFF while BCL2_T is assumed to be ON. The seven source nodes corresponding to the drugs are still kept as source nodes in the network. The outcome nodes “Apoptosis" and “Proliferation" are multi-level nodes representing a cell′s propensity to commit programmed cell death or cell cycle progression, respectively. Apoptosis has four levels and is represented by three Boolean nodes. Proliferatoin has five levels and is represented by four Boolean nodes. The highest level of apoptosis corresponds to full commitment to apoptosis and lower levels correspond to partial commitment. The regulatory function that represents a higher level of apoptosis or proliferation is more stringent compared to the regulatory function of a lower level. Under the current chosen initial condition and without any drugs, the natural attractors are cancerous steady states with low level of apoptosis (Apoptosis = 0) and high level of proliferation (Proliferation = 3 or 4). In the reduced network under the current chosen initial condition, PI3K inhibitor can only lead to apoptosis level 2 and no drug (combinations) can lead to apoptosis level 3.

B.3.3 ABA

On the epidermes of leaves and other aerial plant parts, a pair of guard cells modulate the size of natural openings known as stomata, which are the entry and exit points of gas exchange. Albert et al. constructed a Boolean model of the signaling process involved in stomatal closure in response to the drought hormone abscisic acid(ABA) [111]. The nodes of the network include signaling proteins, small molecules, effectors of ion flow and conceptual nodes such as the outcome node âĂIJClosureâĂİ. The activation of ABA will lead to stomatal closure starting from a partially specified initial conditions, which is also captured by the dynamical model. Albert et al. also systematically studied the effect of knockout and constitutive activity in the presence or absence of ABA, obtaining many results consistent with experimental results. In the absence of ABA, only the constitutive activation of reactive oxygen species (ROS) will lead to a high closure probability similar to the case in the presence of ABA. The detailed Boolean rules are given in Table 2 and S1 Text of [111]. The relevant initial conditions (which specify the state of 54 nodes based on biological knowledge and leave 26 unspecified) and corresponding natural attractors are given in Table S6 and S7 of [111].

A first interesting question to apply our target control algorithm is what nodes need to be controlled to lead to closure in the absence of ABA. The minimal solution given by our algorithm involves at least two nodes. There are only two compatible solutions with size two, that is, activation of H_2O efflux and Microtubule or TCTP. This solution is not very insightful as these nodes are direct regulators of the node Closure. The algorithm gives several incompatible solutions: Ca^{2+} and any of ROS, RBOH, OST1, $\sim$ABI2, $\sim$HAB1, $\sim$PP2CA or H_2O efflux. The solutions are incompatible as Ca^{2+} is regulated by a negative feedback loop and oscillates in the natural dynamics of the system. The fact that the target control solution requires more than activation of ROS is due to the fact that the target control solution is applicable for all initial conditions, while the knockout or constitutive activation study starts with biologically relevant initial conditions.

Another interesting target is the activation of H_2O efflux, a key contributor (and upstream node) of stomatal closure. The incompatible solutions indicated above still hold (except Ca^{2+} and H_2O efflux since H_2O efflux is the target). The minimal compatible solutions involve three nodes, as shown in Table B.1. Interestingly, the LDOI of these solutions do not contain closure, meaning that they are enough to activate H_2O efflux however not enough to activate closure.

Table B.1: The minimal compatible solutions for activating H_2O efflux. The three-node solution comprises of the activation of any node from each column in the same row.

A	B	C
Kefflux	AnionEM, SLAC1	ROS, RBOH, OST1, PIP21
VATPase, Vacidification	AnionEM, SLAC1, MAPK912	ROS, RBOH

B.3.4 T-LGL leukemia network

T cell large granular lymphocyte (T-LGL) leukemia is a rare blood cancer, where leukemic T-LGL cells do not undergo activation induced cell death (apoptosis) after successfully fighting a virus like a normal T cells do. Zhang *et al.* [2] construct a network model consisting of the proteins involved in the activation of T cells, in activation induced cell death, as well as a number of proteins that were observed to be abnormally highly expressed or active in T-LGL cells. The original network has 60 nodes and 142 regulatory edges, details of which and the regulatory functions can be found in Tables S1-S4 of [2]. The model captures the normal (apoptosis) and leukemic (survival) states of the system [2, 3]. For the discussion below, we consider the biological relevant initial condition where the source nodes IL15, Stimuli are ON and PDGF, Stimuli2, CD45 and TAX are OFF and the cell is in a resting (inactive) state.

Previous research established that S1P = 0, PDGFR = 0, SPHK1 = 0 forms a stable motif and fixing any of the three nodes in the OFF state can control the stable motif, which can drive the system to the normal steady state. The target control algorithmâĂŹs minimal solutions for the target Apoptosis involve control of two nodes, which can involve any of the above three nodes combined with any of IL2RB, IL2RBT, RAS, GRB2, PI3K, NFκB. This discrepancy illustrates the difference between LDOI and DOI. Stable motif analysis indicates that once the $\sim$S1P, $\sim$PDGFR, $\sim$SPHK1 stable motif is established, the systemâĂŹs natural dynamics will lead to the sequential stabilization of two other stable motifs and finally will converge to Apoptosis [6]. This suggests that the DOI of any of $\sim$S1P, $\sim$PDGFR, $\sim$SPHK1 will be the whole normal steady state. However, the LDOI of any of $\sim$S1P, $\sim$PDGFR, $\sim$SPHK1 is only part of the normal steady state and does not include the following two stable motifs. Thus in this case our target control algorithm gives a solution that is more stringent than necessary, which can guarantee to reach apoptosis based solely on the logical regulatory functions. This solution is expected to be more robust to possible stochastic fluctuations or perturbations in the system.

Appendix C
Appendix of Chapter Four

C.1 Relevant initial conditions and their corresponding steady states of of the von Dassow et al. ODE model

The “ crisp” initial condition is that *wg*/*IWG* in the first cell is at maximal concentration (1), *en*/*EN* in the second cell has concentration 1, the source nodes B are fixed at 0.4 in each cell and all the other nodes have zero concentration.

Wild type steady state of the von Dassow et al. model for the second parameter set provided by the *Ingeneue* program [153, 155], using normalized concentration variables

$$\begin{aligned}
c(en_2) &= c(EN_2) = 0.986,\\
c(wg_1) &= 0.857,\\
c(IWG_1) &= 0.006,\\
c(EWG_{0,0}) &= c(EWG_{0,3-5}) = 0.005,\\
c(EWG_{0,1}) &= c(EWG_{0,2}) = 0.011,\\
c(EWG_{1,0}) &= c(EWG_{1,3}) = 0.269,\\
c(EWG_{1,1-2}) &= c(EWG_{1,4-5}) = 0.264,\\
c(EWG_{2,0-3}) &= 0.005,\\
c(EWG_{2,4}) &= c(EWG_{2,5}) = 0.011,\\
c(ptc_0) &= c(ptc_1) = c(ptc_3) = 0.995,\\
c(ptc_2) &= 0.001,
\end{aligned}$$

$$\begin{aligned}
c(PTC_{0,*}) &= c(PTC_{1,*}) = c(PTC_{3,*}) = 0.166,\\
c(ci_0) &= c(ci_1) = c(ci_3) = 0.868,\\
c(ci_2) &= 0.007,\\
c(CI_0) &= c(CI_1) = c(CI_3) = 0.057,\\
c(CI_2) &= 0.005,\\
c(CN_0) &= c(CN_1) = c(CN_3) = 0.42,\\
c(CN_2) &= 0.001,\\
c(hh_2) &= 1,\\
c(HH_{2,0}) &= c(HH_{2,3}) = 0.072,\\
c(PH_{1,1-2}) &= c(PH_{3,4-5}) = 0.001,
\end{aligned}$$

where $i,*$ represents all sides of the ith cell. The concentration of the other nodes is smaller than 10^{-5}.

Another initial condition considered here is a nearly-null initial condition, wherein intra-cellular nodes have a concentration of 0.05 in the first and third cell and 0.15 in the second and fourth (zeroth) cell; membrane-localized nodes have concentration of 0.15 for even-numbered sides and 0.05 for odd-numbered sides in every cell. This initial condition yields an unpatterned steady state for the majority of parameter sets.

Unpatterned steady state of the von Dassow et al. model, for the second parameter set provided by the *Ingeneue* program [153, 155], using normalized concentrations:

$$\begin{aligned}
c(wg_*) &= 0.857,\\
c(IWG_*) &= 0.007\\
c(EWG_{*,*}) &= 0.28,\\
c(ptc_*) &= 0.996,\\
c(PTC_{*,*}) &= 0.166,\\
c(ci_*) &= 0.868,\\
c(CI_*) &= 0.057,\\
c(CN_*) &= 0.42,
\end{aligned}$$

where $*$ represents for all cells, and $*,*$ represents for all sides in all cells. The concentration of the other nodes is smaller than 10^{-5}.

Bibliography

[1] ALBERT, R. and H. G. OTHMER (2003) "The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster," *Journal of Theoretical Biology*, **223**(1), pp. 1–18.

[2] ZHANG, R., M. V. SHAH, J. YANG, S. B. NYLAND, X. LIU, J. K. YUN, R. ALBERT, and T. P. LOUGHRAN (2008) "Network Model of Survival Signaling in Large Granular Lymphocyte Leukemia," *Proc. Natl. Acad. Sci. USA*, **105**(42), pp. 16308–16313.

[3] SAADATPOUR, A., R.-S. WANG, A. LIAO, X. LIU, T. P. LOUGHRAN, I. ALBERT, and R. ALBERT (2011) "Dynamical and structural analysis of a T cell survival network identifies novel candidate therapeutic targets for large granular lymphocyte leukemia," *PLoS Computational Biology*, **7**(11), p. e1002267.

[4] ALBERT, R. and R. ROBEVA (2015) "Chapter 4 - Signaling Networks: Asynchronous Boolean Models," in *Algebraic and Discrete Mathematical Methods for Modern Biology* (R. S. Robeva, ed.), Academic Press, Boston, pp. 65 – 91.

[5] STEINWAY, S. N., J. G. ZAÑUDO, W. DING, C. B. ROUNTREE, D. J. FEITH, T. P. LOUGHRAN, and R. ALBERT (2014) "Network Modeling of TGFβ Signaling in Hepatocellular Carcinoma Epithelial-to-Mesenchymal Transition Reveals Joint Sonic Hedgehog and Wnt Pathway Activation," *Cancer Research*, **74**(21), pp. 5963–5977.

[6] ZAÑUDO, J. G. T. and R. ALBERT (2015) "Cell Fate Reprogramming by Control of Intracellular Network Dynamics," *PLoS Computational Biology*, **11**(4).

[7] ALON, U. (2006) *An Introduction to Systems Biology: Design Principles of Biological Circuits*, 1 ed., Chapman and Hall/CRC.

[8] PALSSON, B. (2006) *Systems Biology: Properties of Reconstructed Networks*, Cambridge University Press, Cambridge;New York;.

[9] WANG, R.-S., A. SAADATPOUR, and R. ALBERT (2012) "Boolean modeling in systems biology: an overview of methodology and applications," *Physical Biology*, **9**(5), p. 055001.

[10] Barabási, A.-L. and M. Pósfai (2016) *Network science*, Cambridge University Press, Cambridge.

[11] Newman, M. E. J. (2010) *Networks: an introduction*, Oxford University Press, Oxford;New York;.

[12] Liu, J. O. (1998) "Everything you need to know about the yeast two-hybrid system," *Nat. Struct. Mol. Biol.*, **5**(7), pp. 535–536.

[13] Kerppola, T. K. (2006) "Design and implementation of bimolecular fluorescence complementation (BiFC) assays for the visualization of protein interactions in living cells," *Nat. Protocols*, **1**(3), pp. 1278–1286, protocol.

[14] Markham, K., Y. Bai, and G. Schmitt-Ulms (2007) "Co-immunoprecipitations revisited: an update on experimental concepts and their implementation for sensitive interactome investigations of endogenous proteins," *Analytical and Bioanalytical Chemistry*, **389**(2), pp. 461–473.

[15] Uetz, P., L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J. M. Rothberg (2000) "A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae," *Nature*, **403**(6770), pp. 623–627.

[16] Guruharsha, K., J.-F. Rual, B. Zhai, J. Mintseris, P. Vaidya, N. Vaidya, C. Beekman, C. Wong, D. Y. Rhee, O. Cenaj, E. McKillip, S. Shah, M. Stapleton, K. H. Wan, C. Yu, B. Parsa, J. W. Carlson, X. Chen, B. Kapadia, K. VijayRaghavan, S. P. Gygi, S. E. Celniker, R. A. Obar, and S. Artavanis-Tsakonas (2011) "A Protein Complex Network of Drosophila melanogaster," *Cell*, **147**(3), pp. 690 – 703.

[17] Rual, J.-F., K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, G. F. Berriz, F. D. Gibbons, M. Dreze, N. Ayivi-Guedehoussou, N. Klitgord, C. Simon, M. Boxem, S. Milstein, J. Rosenberg, D. S. Goldberg, L. V. Zhang, S. L. Wong, G. Franklin, S. Li, J. S. Albala, J. Lim, C. Fraughton, E. Llamosas, S. Cevik, C. Bex, P. Lamesch, R. S. Sikorski, J. Vandenhaute, H. Y. Zoghbi, A. Smolyar, S. Bosak, R. Sequerra, L. Doucette-Stamm, M. E. Cusick, D. E. Hill, F. P. Roth, and M. Vidal (2005) "Towards a proteome-scale map of the human protein-protein interaction network," *Nature*, **437**(7062), pp. 1173–1178.

[18] Jeong, H., S. P. Mason, A.-L. Barabási, and Z. N. Oltvai (2001) "Lethality and centrality in protein networks," *Nature*, **411**(6833), pp. 41–42, `cond-mat/0105306`.

[19] von Dassow, G., E. Meir, E. M. Munro, and G. M. Odell (2000) "The segment polarity network is a robust developmental module," *Nature*, **406**(6792), pp. 188–192.

[20] Gomperts, B. D., P. E. R. Tatham, and I. M. Kramer (2009) *Signal transduction*, Elsevier/Academic Press.

[21] Ikushima, H. and K. Miyazono (2010) "TGFβ signalling: a complex web in cancer progression." *Nature reviews. Cancer*, **10**(6), pp. 415–424.

[22] Albert, R. and A.-L. Barabási (2002) "Statistical mechanics of complex networks," *Rev. Mod. Phys.*, **74**(1), pp. 47–97.

[23] Jeong, H., B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási (2000) "The large-scale organization of metabolic networks," *Nature*, **407**(6804), pp. 651–654.

[24] Liu, Y.-Y., J.-J. Slotine, and A.-L. Barabási (2011) "Controllability of complex networks," *Nature*, **473**(7346), pp. 167–173.

[25] Zañudo, J. G. T., G. Yang, and R. Albert (2016), "Structure-based control of complex networks with nonlinear dynamics," 1605.08415.

[26] Aldana, M., S. Coppersmith, and L. P. Kadanoff (2003) "Perspectives and Problems in Nolinear Science: A Celebratory Volume in Honor of Lawrence Sirovich," chap. Boolean Dynamics with Random Couplings, pp. 23–89.

[27] Gilmore, T. D. (2006) "Introduction to NF-κB: players, pathways, perspectives." *Oncogene*, **25**(51), pp. 6680–6684.

[28] Ma'ayan, A., S. L. Jenkins, S. Neves, A. Hasseldine, E. Grace, B. Dubin-Thaler, N. J. Eungdamrong, G. Weng, P. T. Ram, J. J. Rice, A. Kershenbaum, G. A. Stolovitzky, R. D. Blitzer, and R. Iyengar (2005) "Formation of regulatory patterns during signal propagation in a Mammalian cellular network." *Science (New York, N.Y.)*, **309**(5737), pp. 1078–1083.

[29] Freeman, L. C. (1977) "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, **40**(1), pp. 35–41.

[30] Palla, G., I. DerÃl'nyi, I. Farkas, and T. Vicsek (2005) "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, **435**(7043), pp. 814–818.

[31] Thomas, R. and R. d'Ari (1990) *Biological feedback*, CRC press.

[32] SMOOT, M. E., K. ONO, J. RUSCHEINSKI, P.-L. WANG, and T. IDEKER (2011) “Cytoscape 2.8: new features for data integration and network visualization,” *Bioinformatics*, **27**(3), p. 431.

[33] HAGBERG, A. A., D. A. SCHULT, and P. J. SWART (2008) “Exploring network structure, dynamics, and function using NetworkX,” in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Pasadena, CA USA, pp. 11–15.

[34] BATAGELJ, V. and A. MRVAR (2002) *Pajek— Analysis and Visualization of Large Networks*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 477–478.

[35] KESTLER, H. A., C. WAWRA, B. KRACHER, and M. KÃIJHL (2008) “Network modeling of signal transduction: establishing the global view,” *BioEssays*, **30**(11-12), pp. 1110–1125.

[36] KARLEBACH, G. and R. SHAMIR (2008) “Modelling and analysis of gene regulatory networks,” *Nat. Rev. Mol. Cell Biol.*, **9**(10), pp. 770–780.

[37] TYSON, J. J., K. C. CHEN, and B. NOVAK (2003) “Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell,” *Current Opinion in Cell Biology*, **15**(2), pp. 221 – 231.

[38] KAUFFMAN, S. A. (1993) *The Origins of Order: Self-Organization and Selection in Evolution*, 1 ed., Oxford University Press, USA.

[39] ALDRIDGE, B. B., J. SAEZ-RODRIGUEZ, J. L. MUHLICH, P. K. SORGER, and D. A. LAUFFENBURGER (2009) “Fuzzy Logic Analysis of Kinase Pathway Crosstalk in TNF/EGF/Insulin-Induced Signaling,” *PLOS Computational Biology*, **5**, pp. 1–13.

[40] CHAOUIYA, C. (2007) “Petri net modelling of biological networks,” *Briefings in Bioinformatics*, **8**(4), p. 210.

[41] LI, F., T. LONG, Y. LU, Q. OUYANG, and C. TANG (2004) “The yeast cell-cycle network is robustly designed,” *Proc. Natl. Acad. Sci*, **101**(14), pp. 4781–4786.

[42] SAMAGA, R., J. SAEZ-RODRIGUEZ, L. G. ALEXOPOULOS, P. K. SORGER, and S. KLAMT (2009) “The Logic of EGFR/ErbB Signaling: Theoretical Properties and Analysis of High-Throughput Data,” *PLOS Computational Biology*, **5**(8), pp. 1–19.

[43] SCHLATTER, R., K. SCHMICH, I. AVALOS VIZCARRA, P. SCHEURICH, T. SAUTER, C. BORNER, M. EDERER, I. MERFORT, and O. SAWODNY (2009) “ON/OFF and Beyond - A Boolean Model of Apoptosis,” *PLOS Computational Biology*, **5**(12), pp. 1–13.

[44] THAKAR, J., A. K. PATHAK, L. MURPHY, R. ALBERT, and I. M. CATTADORI (2012) "Network Model of Immune Responses Reveals Key Effectors to Single and Co-infection Dynamics by a Respiratory Bacterium and a Gastrointestinal Helminth," *PLOS Computational Biology*, **8**(1), pp. 1–19.

[45] CAMPBELL, B. J., L. YU, J. F. HEIDELBERG, and D. L. KIRCHMAN (2011) "Activity of abundant and rare bacteria in a coastal ocean," *Proceedings of the National Academy of Sciences*, **108**(31), pp. 12776–12781.

[46] SAADATPOUR, A. and R. ALBERT (2013) "Boolean modeling of biological regulatory networks: A methodology tutorial," *Methods*, **62**(1), pp. 3 – 12, modeling Gene Expression.

[47] LI, S., S. M. ASSMANN, and R. ALBERT (2006) "Predicting Essential Components of Signal Transduction Networks: A Dynamic Model of Guard Cell Abscisic Acid Signaling," *PLOS Biology*, **4**(10), pp. 1–17.

[48] ALBERT, R., B. DASGUPTA, R. DONDI, S. KACHALO, E. SONTAG, A. ZELIKOVSKY, and K. WESTBROOKS (2007) "A Novel Method for Signal Transduction Network Inference from Indirect Experimental Evidence," *Journal of Computational Biology*, **14**(7), pp. 927–949.

[49] KACHALO, S., R. ZHANG, E. SONTAG, R. ALBERT, and B. DASGUPTA (2008) "NET-SYNTHESIS: a software for synthesis, inference and simplification of signal transduction networks," *Bioinformatics*, **24**(2), p. 293.

[50] AKMAN, O. E., S. WATTERSON, A. PARTON, N. BINNS, A. J. MILLAR, and P. GHAZAL (2012) "Digital clocks: simple Boolean models can quantitatively describe circadian systems," *J. R. Soc. Interface*, **9**.

[51] KLEMM, K. and S. BORNHOLDT (2005) "Stable and unstable attractors in Boolean networks," *Phys. Rev. E*, **72**, p. 055101.

[52] CLARKE, E. M., O. GRUMBERG, and D. PELED (1999) *Model-checking*, MIT Press, Cambridge, MA.

[53] BORNHOLDT, S. (2008) "Boolean network models of cellular regulation: prospects and limitations," *Journal of The Royal Society Interface*, **5**(Suppl 1), pp. S85–S94.

[54] ZAñUDO, J. G. T. and R. ALBERT (2013) "An effective network reduction approach to find the dynamical repertoire of discrete dynamic networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **23**(2), p. 025111, http://dx.doi.org/10.1063/1.4809777.

[55] CAMPBELL, C. and R. ALBERT (2014) "Stabilization of perturbed Boolean network attractors through compensatory interactions," *BMC systems biology*, **8**(1), p. 53.

[56] CHAOUIYA, C., D. BÉRENGUIER, S. M. KEATING, A. NALDI, M. P. VAN IERSEL, N. RODRIGUEZ, A. DRÄGER, F. BÜCHEL, T. COKELAER, B. KOWAL, B. WICKS, E. GONÇALVES, J. DORIER, M. PAGE, P. T. MONTEIRO, A. VON KAMP, I. XENARIOS, H. DE JONG, M. HUCKA, S. KLAMT, D. THIEFFRY, N. LE NOVÈRE, J. SAEZ-RODRIGUEZ, and T. HELIKAR (2013) "SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools," *BMC Systems Biology*, **7**(1), p. 135.

[57] NALDI, A., D. BERENGUIER, A. FAURÉ, F. LOPEZ, D. THIEFFRY, and C. CHAOUIYA (2009) "Logical modelling of regulatory networks with GINsim 2.3," *Biosystems*, **97**.

[58] CHAOUIYA, C., A. NALDI, and D. THIEFFRY (2012) "Logical modelling of gene regulatory networks with GINsim," *Methods Mol Biol*, **804**.

[59] MÜSSEL, C., M. HOPFENSITZ, and H. A. KESTLER (2010) "BoolNet–an R package for generation, reconstruction and analysis of Boolean networks," *Bioinformatics*, **26**.

[60] ALBERT, I., J. THAKAR, S. LI, R. ZHANG, and R. ALBERT (2008) "Boolean network simulations for life scientists," *Source Code Biol. Med.*, **3**.

[61] HINKELMANN, F., M. BRANDON, B. GUANG, R. MCNEILL, G. BLEKHERMAN, A. VELIZ-CUBA, and R. LAUBENBACHER (2011) "ADAM: analysis of discrete models of biological systems using computer algebra," *BMC Bioinforma*, **12**.

[62] HELIKAR, T., B. KOWAL, and J. A. ROGERS (2013) "A cell simulator platform: the cell collective," *Clin Pharmacol Ther*, **93**.

[63] KLAMT, S., J. SAEZ-RODRIGUEZ, and E. D. GILLES (2007) "Structural and functional analysis of cellular networks with cell NetAnalyzer," *BMC System Biology*, **1**.

[64] TERFVE, C. D. A., T. COKELAER, D. HENRIQUES, A. MACNAMARA, E. GONÇALVES, M. K. MORRIS, M. VAN IERSEL, D. A. LAUFFENBURGER, and J. SAEZ-RODRIGUEZ (2012) "CellNOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms," *BMC Syst Biol*, **6**.

[65] HELIKAR, T. and J. A. ROGERS (2009) "ChemChains: a platform for simulation and analysis of biochemical networks aimed to laboratory scientists," *BMC Syst Biol*, **3**.

[66] KRUMSIEK, J., S. PÖLSTERL, D. M. WITTMANN, and F. J. THEIS (2010) "Odefy–from discrete to continuous models," *BMC Bioinform*, **11**.

[67] ZHENG, J., D. ZHANG, P. F. PRZYTYCKI, R. ZIELINSKI, J. CAPALA, and T. M. PRZYTYCKA (2010) "SimBoolNet–a cytoscape plugin for dynamic simulation of signaling networks," *Bioinformatics*, **26**.

[68] DI CARA, A., A. GARG, G. DE MICHELI, I. XENARIOS, and L. MENDOZA (2007) "Dynamic simulation of regulatory networks using SQUAD," *BMC Bioinform*, **8**.

[69] SAADATPOUR, A., R. ALBERT, and T. C. RELUGA (2013) "A Reduction Method for Boolean Network Models Proven to Conserve Attractors," *SIAM Journal on Applied Dynamical Systems*, **12**(4), pp. 1997–2011.

[70] WANG, R.-S. and R. ALBERT (2011) "Elementary signaling modes predict the essentiality of signal transduction network components," *BMC Systems Biology*, **5**(1), p. 44.

[71] SUN, Z. and R. ALBERT (2016) "Node-independent elementary signaling modes: A measure of redundancy in Boolean signaling transduction networks," *Network Science*, **4**(3), p. 273âĂŞ292.

[72] MCCLUSKEY, E. J. (1956) "Minimization of Boolean Functions," *Bell System Technical Journal*, **35**(6), pp. 1417–1444.

[73] LIU, Y.-Y. and A.-L. BARABÁSI (2016) "Control principles of complex systems," *Rev. Mod. Phys.*, **88**, p. 035006.

[74] MOTTER, A. E. (2015) "Networkcontrology," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **25**(9), p. 097621.

[75] KALMAN, R. E. (1963) "Mathematical Description of Linear Dynamical Systems," *Journal of the Society for Industrial and Applied Mathematics Series A Control*, **1**(2), pp. 152–192.

[76] LIN, C.-T. (1974) "Structural controllability," *IEEE Transactions on Automatic Control*, **19**(3), pp. 201–208.

[77] MOCHIZUKI, A., B. FIEDLER, G. KUROSAWA, and D. SAITO (2013) "Dynamics and control at feedback vertex sets. II: a faithful monitor to determine the diversity of molecular activities in regulatory networks," *Journal of Theoretical Biology*, **335**, pp. 130–146.

[78] GAO, J., Y.-Y. LIU, R. M. D'SOUZA, and A.-L. BARABÁSI (2014) "Target control of complex networks," *Nature Communications*, **5**, pp. 5415 EP –, article.

[79] ANGULO, M. T., C. H. MOOG, and Y.-Y. LIU (2017) "Controlling microbial communities: a theoretical framework," *bioRxiv*.

[80] YANG, G., C. CAMPBELL, and R. ALBERT (2016) "Compensatory interactions to stabilize multiple steady states or mitigate the effects of multiple deregulations in biological networks," *Phys. Rev. E*, **94**, p. 062316.

[81] NIJMEIJER, H. and A. VAN DER SCHAFT (1990) *Nonlinear Dynamical Control Systems*, Springer-Verlag New York, Inc., New York, NY, USA.

[82] ISIDORI, A. (1995) *Nonlinear Control Systems*, 3rd ed., Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[83] SUN, Z., X. JIN, R. ALBERT, and S. M. ASSMANN (2014) "Multi-level Modeling of Light-Induced Stomatal Opening Offers New Insights into Its Regulation by Drought," *PLoS Computational Biology*, **10**(11), p. e1003930.

[84] MICHAL, G. and A. H. BRIAN (2010) "Deregulated signalling networks in human brain tumours," *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics*, **1804**(3), pp. 476 – 483.

[85] MISKOV-ZIVANOV, N., M. S. TURNER, L. P. KANE, P. A. MOREL, and J. R. FAEDER (2013) "The duration of T cell stimulation is a critical determinant of cell fate and plasticity," *Science signaling*, **6**(300), p. ra97.

[86] CREIXELL, P., E. M. SCHOOF, J. T. ERLER, and R. LINDING (2012) "Navigating cancer network attractors for tumor-specific therapy," *Nature biotechnology*, **30**(9), p. 842.

[87] WELLS, D. K., W. L. KATH, and A. E. MOTTER (2015) "Control of Stochastic and Induced Switching in Biophysical Networks," *Phys. Rev. X*, **5**, p. 031036.

[88] HANAHAN, D. and R. A. WEINBERG (2011) "Hallmarks of Cancer: The Next Generation," *Cell*, **144**(5), pp. 646–674.

[89] KINNEY, R., P. CRUCITTI, R. ALBERT, and V. LATORA (2005) "Modeling cascading failures in the North American power grid," *The European Physical Journal B*, **46**(1), pp. 101–107.

[90] DERRIDA, B. and Y. POMEAU (1986) "Random Networks of Automata: A Simple Annealed Approximation," *Europhysics Letters*, **1**(2), pp. 45–49.

[91] ALDANA, M. and P. CLUZEL (2003) "A Natural Class of Robust Networks," *Proc. Natl. Acad. Sci. USA*, **100**(15), pp. 8710–8714.

[92] ALDANA, M. (2003) "Boolean dynamics of networks with scale-free topology," *Physica D*, **185**(1), pp. 45–66.

[93] Pomerance, A., E. Ott, M. Girvan, and W. Losert (2009) “The effect of network topology on the stability of discrete state models of genetic control,” *Proc. Natl. Acad. Sci. USA*, **106**(20), pp. 8209–8214.

[94] Luque, B. and R. V. Solé (1997) “Phase transitions in random networks: Simple analytic determination of critical points,” *Phys. Rev. E*, **55**, pp. 257–260.

[95] Li, S., C. M. Armstrong, N. Bertin, H. Ge, S. Milstein, M. Boxem, P.-O. Vidalain, J.-D. J. Han, A. Chesneau, T. Hao, et al. (2004) “A Map of the Interactome Network of the Metazoan C. elegans,” *Science*, **303**(5657), pp. 540–543.

[96] Barabási, A.-L., N. Gulbahce, and J. Loscalzo (2011) “Network medicine: a network-based approach to human disease,” *Nature Reviews Genetics*, **12**(1), pp. 56–68.

[97] Shmulevich, I. and S. A. Kauffman (2004) “Activities and Sensitivities in Boolean Network Models,” *Phys. Rev. Lett.*, **93**, p. 048701.

[98] Ebadi, H. and K. Klemm (2014) “Boolean networks with veto functions,” *Phys. Rev. E*, **90**, p. 022815.

[99] Kauffman, S., C. Peterson, B. Samuelsson, and C. Troein (2003) “Random Boolean Network Models and the Yeast Transcriptional Network,” *Proc. Natl. Acad. Sci. USA*, **100**(25), pp. 14796–14799.

[100] Li, Y., J. O. Adeyeye, D. Murrugarra, B. Aguilar, and R. Laubenbacher (2013) “Boolean nested canalizing functions: A comprehensive analysis,” *Theoretical Computer Science*, **481**, pp. 24 – 36.

[101] Peixoto, P. T. (2010) “The phase diagram of random Boolean networks with nested canalizingfunctions,” *The European Physical Journal B*, **78**(2), pp. 187–192.

[102] Hartman, J. L., B. Garvik, and L. Hartwell (2001) “Principles for the buffering of genetic variation,” *Science*, **291**(5506), pp. 1001–4.

[103] Chan, D. A. and A. J. Giaccia (2011) “Harnessing synthetic lethal interactions in anticancer drug discovery,” *Nat Rev Drug Discov*, **10**(5), pp. 351–364.

[104] Boldhaus, G., F. Greil, and K. Klemm (2013) “Prediction of lethal and synthetically lethal knock-outs in regulatory networks,” *Theory in Biosciences*, **132**(1), pp. 17–25.

[105] Motter, A. E., N. Gulbahce, E. Almaas, and A.-L. Barabási (2008) “Predicting synthetic rescues in metabolic networks,” *Molecular Systems Biology*, **4**(1).

[106] SOCOLAR, J. E. S. and S. A. KAUFFMAN (2003) "Scaling in Ordered and Critical Random Boolean Networks," *Phys. Rev. Lett.*, **90**, p. 068702.

[107] NAKAYA, Y. and G. SHENG (2008) "Epithelial to mesenchymal transition during gastrulation: An embryological view," *Development, Growth and Differentiation*, **50**(9), p. 755.

[108] NALDI, A., J. CARNEIRO, C. CHAOUIYA, and D. THIEFFRY (2010) "Diversity and plasticity of Th cell types predicted from regulatory network modelling," *PLoS computational biology*, **6**(9), p. e1000912.

[109] STEINWAY, S. N., J. G. T. ZAÑUDO, P. J. MICHEL, D. J. FEITH, T. P. LOUGHRAN, and R. ALBERT (2015) "Combinatorial interventions inhibit TGFb-driven epithelial-to-mesenchymal transition and support hybrid cellular phenotypes," *Npj Systems Biology And Applications*, **1**, pp. 15014 EP –, article.

[110] GÓMEZ TEJEDA ZAÑUDO, J., M. SCALTRITI, and R. ALBERT (2017) "A network modeling approach to elucidate drug resistance mechanisms and predict combinatorial drug treatments in breast cancer," *Cancer Convergence*, **1**(1), p. 5. URL `https://doi.org/10.1186/s41236-017-0007-6`

[111] ALBERT, R., B. R. ACHARYA, B. W. JEON, J. G. T. ZAÃŚUDO, M. ZHU, K. OSMAN, and S. M. ASSMANN (2017) "A new discrete dynamic model of ABA-induced stomatal closure predicts key feedback loops," *PLOS Biology*, **15**(9), pp. 1–35.

[112] MISKOV-ZIVANOV, N., D. MARCULESCU, and J. R. FAEDER (2013) "Dynamic Behavior of Cell Signaling Networks: Model Design and Analysis Automation," in *Proceedings of the 50th Annual Design Automation Conference*, DAC '13, ACM, New York, NY, USA, pp. 8:1–8:6.
URL `http://doi.acm.org/10.1145/2463209.2488743`

[113] CORNELIUS, S. P., W. L. KATH, and A. E. MOTTER (2013) "Realistic control of network dynamics," *Nature Communications*, **4**, pp. 1942 EP –, article.

[114] WANG, L.-Z., R.-Q. SU, Z.-G. HUANG, X. WANG, W.-X. WANG, C. GREBOGI, and Y.-C. LAI (2016) "A geometrical approach to control and controllability of nonlinear dynamical networks," *Nature Communications*, **7**, pp. 11323 EP –, article.

[115] ZAÃŚUDO, J. G. T., G. YANG, and R. ALBERT (2017) "Structure-based control of complex networks with nonlinear dynamics," *Proceedings of the National Academy of Sciences*, **114**(28), pp. 7234–7239.

[116] MURRUGARRA, D., A. VELIZ-CUBA, B. AGUILAR, and R. LAUBENBACHER (2016) "Identification of control targets in Boolean molecular network models via computational algebra," *BMC Systems Biology*, **10**(1), p. 94.
URL `https://doi.org/10.1186/s12918-016-0332-x`

[117] MURRUGARRA, D. and E. S. DIMITROVA (2015) "Molecular network control through boolean canalization," *EURASIP Journal on Bioinformatics and Systems Biology*, **2015**(1), p. 9.

[118] NICHOLL, D. S. T. (2008) *An Introduction to Genetic Engineering*, 3 ed., Cambridge University Press.

[119] KAUFFMAN, S. (1969) "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, **22**(3), pp. 437 – 467.

[120] GLASS, L. and S. A. KAUFFMAN (1973) "The logical analysis of continuous, non-linear biochemical control networks," *Journal of Theoretical Biology*, **39**(1), pp. 103 – 129.

[121] GLASS, L. (1975) "Classification of biological networks by their qualitative dynamics," *Journal of Theoretical Biology*, **54**(1), pp. 85 – 107.

[122] PAPIN, J. A., T. HUNTER, B. O. PALSSON, and S. SUBRAMANIAM (2005) "Reconstruction of cellular signalling networks and analysis of their properties," *Nature Reviews Molecular Cell Biology*, **6**(2), pp. 99–111.

[123] RUSSELL, S. J. and P. NORVIG (2003) *Artificial intelligence : a modern approach*, Prentice Hall series in artificial intelligence, Prentice Hall/Pearson Education.

[124] PARDALOS, P. M., T. QIAN, and M. G. RESENDE (1998) "A Greedy Randomized Adaptive Search Procedure for the Feedback Vertex Set Problem," *Journal of Combinatorial Optimization*, **2**(4), pp. 399–412.

[125] FESTA, P., P. M. PARDALOS, and M. G. C. RESENDE (2001) "Algorithm 815: FORTRAN Subroutines for Computing Approximate Solutions of Feedback Set Problems Using GRASP," *ACM Trans. Math. Softw.*, **27**(4), pp. 456–464.

[126] ZERTUCHE, F. (2009) "On the robustness of NK-Kauffman networks against changes in their connections and Boolean functions," *Journal of Mathematical Physics*, **50**(4), p. 043513, `https://doi.org/10.1063/1.3116166`.
URL `https://doi.org/10.1063/1.3116166`

[127] MAHESHWARI, P. and R. ALBERT (2017) "A framework to find the logic backbone of a biological network," *BMC Systems Biology*, **11**(1), p. 122.
URL `https://doi.org/10.1186/s12918-017-0482-5`

[128] WANG, R.-S., Z. SUN, and R. ALBERT (2013) “Minimal functional routes in directed graphs with dependent edges,” *International Transactions in Operational Research*, **20**(3), pp. 391–409.

[129] NEPUSZ, T. and T. VICSEK (2012) “Controlling edge dynamics in complex networks,” *Nature Physics*, **8**(7), pp. 568–573.

[130] RUTHS, J. and D. RUTHS (2014) “Control Profiles of Complex Networks,” *Science*, **343**(6177), pp. 1373–1376.

[131] GATES, A. J. and L. M. ROCHA (2016) “Control of complex networks requires both structure and dynamics,” *Scientific Reports*, **6**, pp. 24456 EP –, article.

[132] SLOTINE, J. . E. and W. LI (1991) *Applied nonlinear control*, Prentice Hall, Englewood Cliffs, N.J.

[133] SONTAG, E. D. and S. O. SERVICE) (1998) *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, vol. 6;6.;, second ed., Springer New York, New York, NY.

[134] SCHÃĎTTLER, H. M., U. LEDZEWICZ, and S. O. SERVICE) (2012) *Geometric Optimal Control: Theory, Methods and Examples*, vol. 38.;38;, Springer New York, New York, NY.

[135] SHIELDS, R. and J. PEARSON (1976) “Structural controllability of multiinput linear systems,” *IEEE Transactions on Automatic Control*, **21**(2), pp. 203–212.

[136] VINAYAGAM, A., T. E. GIBSON, H.-J. LEE, B. YILMAZEL, C. ROESEL, Y. HU, Y. KWON, A. SHARMA, Y.-Y. LIU, N. PERRIMON, and A.-L. BARABÃĄSI (2016) “Controllability analysis of the directed human protein interaction network identifies disease genes and drug targets,” *Proceedings of the National Academy of Sciences*, **113**(18), pp. 4976–4981.

[137] KAWAKAMI, E., V. K. SINGH, K. MATSUBARA, T. ISHII, Y. MATSUOKA, T. HASE, P. KULKARNI, K. SIDDIQUI, J. KODILKAR, N. DANVE, I. SUBRAMANIAN, M. KATOH, Y. SHIMIZU-YOSHIDA, S. GHOSH, A. JERE, and H. KITANO (2016) “Network analyses based on comprehensive molecular interaction maps reveal robust control structures in yeast stress response pathways,” *Npj Systems Biology And Applications*, **2**, pp. 15018 EP –, article.

[138] GU, S., F. PASQUALETTI, M. CIESLAK, Q. K. TELESFORD, A. B. YU, A. E. KAHN, J. D. MEDAGLIA, J. M. VETTEL, M. B. MILLER, S. T. GRAFTON, and D. S. BASSETT (2015) “Controllability of structural brain networks,” *Nature Communications*, **6**, pp. 8414 EP –, article.

[139] NACHER, J. C. and T. AKUTSU (2013) "Structural controllability of unidirectional bipartite networks," *Scientific Reports*, **3**, pp. 1647 EP –, article.

[140] MÃIJLLER, F.-J. and A. SCHUPPERT (2011) "Few inputs can reprogram biological networks," *Nature*, **478**(7369), p. E4; discussion E4.

[141] FIEDLER, B., A. MOCHIZUKI, G. KUROSAWA, and D. SAITO (2013) "Dynamics and Control at Feedback Vertex Sets. I: Informative and Determining Nodes in Regulatory Networks," *Journal of Dynamics and Differential Equations*, **25**(3), pp. 563–604.

[142] ALLEN, L. J. S. (2011) *An introduction to stochastic processes with applications to biology*, 2nd ed., Chapman and Hall/CRC, Boca Raton, FL.

[143] NOVOZHILOV, A. S., G. P. KAREV, and E. V. KOONIN (2006) "Biological applications of the theory of birth-and-death processes," *Briefings in Bioinformatics*, pp. 70–85.

[144] DALEY, D. J. and J. M. GANI (1999) *Epidemic modelling: an introduction*, vol. 15.;15;, Cambridge University Press, New York;Cambridge, U.K;.

[145] CASTELLANO, C., S. FORTUNATO, and V. LORETO (2009) "Statistical physics of social dynamics," *Rev. Mod. Phys.*, **81**, pp. 591–646.

[146] TYSON, J. J., K. C. CHEN, and B. NOVAK (2003) "Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell," *Current Opinion in Cell Biology*, **15**(2), pp. 221 – 231.

[147] THOMAS, R. (1978) "Logical analysis of systems comprising feedback loops," *Journal of Theoretical Biology*, **73**(4), pp. 631 – 656.

[148] ZHANG, J., P. L. YANG, and N. S. GRAY (2009) "Targeting cancer with small molecule kinase inhibitors," *Nature Reviews Cancer*, **9**, pp. 28 EP –, review Article.

[149] PARDALOS, P. M., T. QIAN, and M. G. C. RESENDE (1998) "A Greedy Randomized Adaptive Search Procedure for the Feedback Vertex Set Problem," *Journal of Combinatorial Optimization*, **2**(4), pp. 399–412.

[150] GALINIER, P., E. LEMAMOU, and M. W. BOUZIDI (2013) "Applying local search to the feedback vertex set problem," *Journal of Heuristics*, **19**(5), pp. 797–818.

[151] BASSETT, D. S., D. L. ALDERSON, and J. M. CARLSON (2012) "Collective decision dynamics in the presence of external drivers," *Phys. Rev. E*, **86**, p. 036105.

[152] CHAVES, M., E. D. SONTAG, and R. ALBERT (2006) "Methods of robustness analysis for Boolean models of gene control networks," *Systems biology*, **153**(4), p. 154.

[153] VON DASSOW, G. and G. ODELL (2002) "Design and constraints of the Drosophila segment polarity module: Robust spatial patterning emerges from intertwined cell state switches," *Journal of Experimental Zoology*, **294**(3), pp. 179–215.

[154] DANIELS, B. C., Y.-J. CHEN, J. P. SETHNA, R. N. GUTENKUNST, and C. R. MYERS (2008) "Sloppiness, robustness, and evolvability in systems biology," *Current Opinion in Biotechnology*, **19**(4), pp. 389–395.

[155] MEIR, E., E. M. MUNRO, G. M. ODELL, and G. VON DASSOW (2002) "Ingeneue: A versatile tool for reconstituting genetic networks, with examples from the segment polarity network," *Journal of Experimental Zoology*, **294**(3), pp. 216–251.

[156] KIRK, D. E. (2004) *Optimal control theory: an introduction*, Dover Publications, Mineola, N.Y.

[157] AKUTSU, T., M. HAYASHIDA, W.-K. CHING, and M. K. NG (2007) "Control of Boolean networks: Hardness results and algorithms for tree structured networks," *Journal of Theoretical Biology*, **244**(4), pp. 670–679.

[158] CHENG, D. and H. QI (2009) "Controllability and observability of Boolean control networks," *Automatica*, **45**(7), pp. 1659 – 1667.

VITA

GANG YANG

yanggangthu@gmail.com

EDUCATION

The Pennsylvania State University, University Park, PA 2012 - 2017(anticipated)
Ph.D. Candidate in Physics GPA: 3.98/4
Ph.D. Minor in Computational Science

Tsinghua University, Beijing, China 2008 - 2012
B.S. in Physics and Mathematics GPA: 86.5%

PUBLICATIONS

1. Structure-based control of complex networks with nonlinear dynamics, Jorge G.T. Zañudo, **Gang Yang**, Réka Albert, *Proceedings of the National Academy of Sciences*, 114, 28, 7234 (2017)
2. Compensatory interactions to Stabilize Multiple Steady States or Mitigate the Effects of Multiple Deregulations in Biological Networks, **Gang Yang**, Colin Campbell, and Réka Albert, *Physical Review E* 94, 062316 (2016)
3. Target Control in Logical Models Using the Domain of Influence of Nodes, **Gang Yang**, and Réka Albert, *Submitted to the special issue "Logical Modeling of Cellular Processes" of Frontiers in Physiology*
4. Modeling of molecular networks, **Gang Yang**, and Réka Albert, *Submitted as a book chapter in a book "The Dynamics of Biological Systems" at Springer*
5. Weak Topological Insulators in PbTe/SnTe Superlattices, **Gang Yang**, Junwei Liu, Liang Fu, Wenhui Duan, Chaoxing Liu, *Physical Review B* 89, 085312 (2014)
6. Heavy Dirac Fermions in a Graphene/Topological Insulator Hetero-Junction, Wendong Cao, Ruixing Zhang, Peizhe Tang, **Gang Yang**, Jorge Sofo, Wenhui Duan, Chaoxing Liu, *2D Materials*, 3, 3 (2016)

PRESENTATION

Damage Mitigation in complex networked system 2017 APS March Meeting, New Orleans
.. NetSci 2017, June, Indiana

TEACHING EXPERIENCE

Teaching Assistant of Introductory General Physics Course Aug 2012 - May 2016
Including Mechanics, Electromagnetism, Thermodynamics, Optics and Quantum Mechanics

HONORS AND AWARDS

Penn State David C. Duncan Graduate Fellowship in Physics 2013, 14, 16
Tsinghua Friends-Zhuyou Scholarship, Tsinghua Friends-HuangyicongKangli Scholarship 2009 - 2011