The Potential Power of Dynamics in Epistasis Analysis

Submitted by
Aseel R. Awdeh

A thesis submitted in partial fulfilment of the requirements
for the degree of Masters in Computer Science with specialization in Bioinformatics

Faculty of Graduate Studies
University of Ottawa
Ottawa, Ontario, Canada

©Aseel R. Awdeh, Ottawa, Canada, 2015

ACKNOWLEDGMENTS

Thank you for investing your time in reading my thesis. I would like to thank my supervisor, Theordore J. Perkins, for his assistance, supervision and valuable feedback throughout the duration of this research project. I would like to thank my co-supervisor, Marcel Turcotte, our collaborating team and members of our lab. I would also like to thank my parents for their ongoing encouragement and support.

ABSTRACT

Inferring regulatory relationships between genes, including the direction and the nature of influence between them, is the foremost problem in the field of genetics. One classical approach to this problem is epistasis analysis. Broadly speaking, epistasis analysis infers the regulatory relationships between a pair of genes in a genetic pathway by considering the patterns of change in an observable trait resulting from single and double deletion of genes. More specifically, a "surprising" situation occurs when the phenotype of a double mutant has a similar, aggravating or alleviating effect compared to the phenotype resulting from the single deletion of either one of the genes. As useful as this broad approach has been, there are limits to its ability to discriminate alternative pathway structures, meaning it is not always possible to infer the relationship between the genes. Here, we explore the possibility of dynamic epistasis analysis. In addition to performing genetic perturbations, we drive a genetic pathway with a dynamic, time-varying upstream signal, where the phenotypic consequence is measured at each time step. We explore the theoretical power of dynamic epistasis analysis by conducting an identifiability analysis of Boolean models of genetic pathways, comparing static and dynamic approaches. We also explore the identifiability of individual links in the pathway. Through these evaluations, we quantify how helpful the addition of dynamics is. We believe that a dynamic input in addition to epistasis analysis is a powerful tool to discriminate between different networks. Our primary findings show that the use of a dynamic input signal alone, without genetic perturbations, appears to be very weak in comparison with the more traditional genetic approaches based on the deletion of genes. However, the combination of dynamical input with genetic perturbations is far more powerful than the classical epistasis analysis approach. In all cases, we find that even relatively simple input dynamics with gene deletions greatly increases the power of epistasis analysis to discriminate alternative network structures and to confidently identify individual links in a network. Our positive results show the potential value of dynamics in epistasis analysis.

Table of Contents

List of Tables

List of Figures

1 INTRODUCTION

This thesis presents a theoretical investigation of how to design experiments aimed at revealing the structure of gene regulatory networks (GRNs). In order to better understand the motivation of this study, we first briefly describe the meaning of a gene and a GRN. Genetic information in the human cell is stored in the form of DNA (or deoxyribonucleic acid). DNA is a biomolecule that is composed of billions of the nitrogen bases (or nucleotides): adenine (A), guanine (G), cytosine (C) and thymine (T). Different sequences of these nucleotides correspond to different genes (where a gene is part of the DNA). A gene (See Figure 1.1A) consists of coding regions (equivalently exons), which translate to a sequence of amino acids, non-coding regions (equivalently introns), which do not specify amino acids, and regulatory sequences, which determine when and where the protein is synthesized. Different genes are made of different sequences of nucleotides, which code for different proteins, and in turn perform different functions. The functions have many critical roles, such as the maintenance, survival, growth, replication, and movement of the cell [1].

Imagine that a human cell is a computer, where the software of the cell is the DNA. Similar to any computer, to be able to execute tasks, the sequences which make up the DNA (software instructions), more specifically the genes, need to be interpreted and decoded by the cell's hardware. The interpretation and conversion of a sequence of nucleotides into protein is known as gene expression. There are two major steps in the process of gene expression: transcription and translation [1, 2].

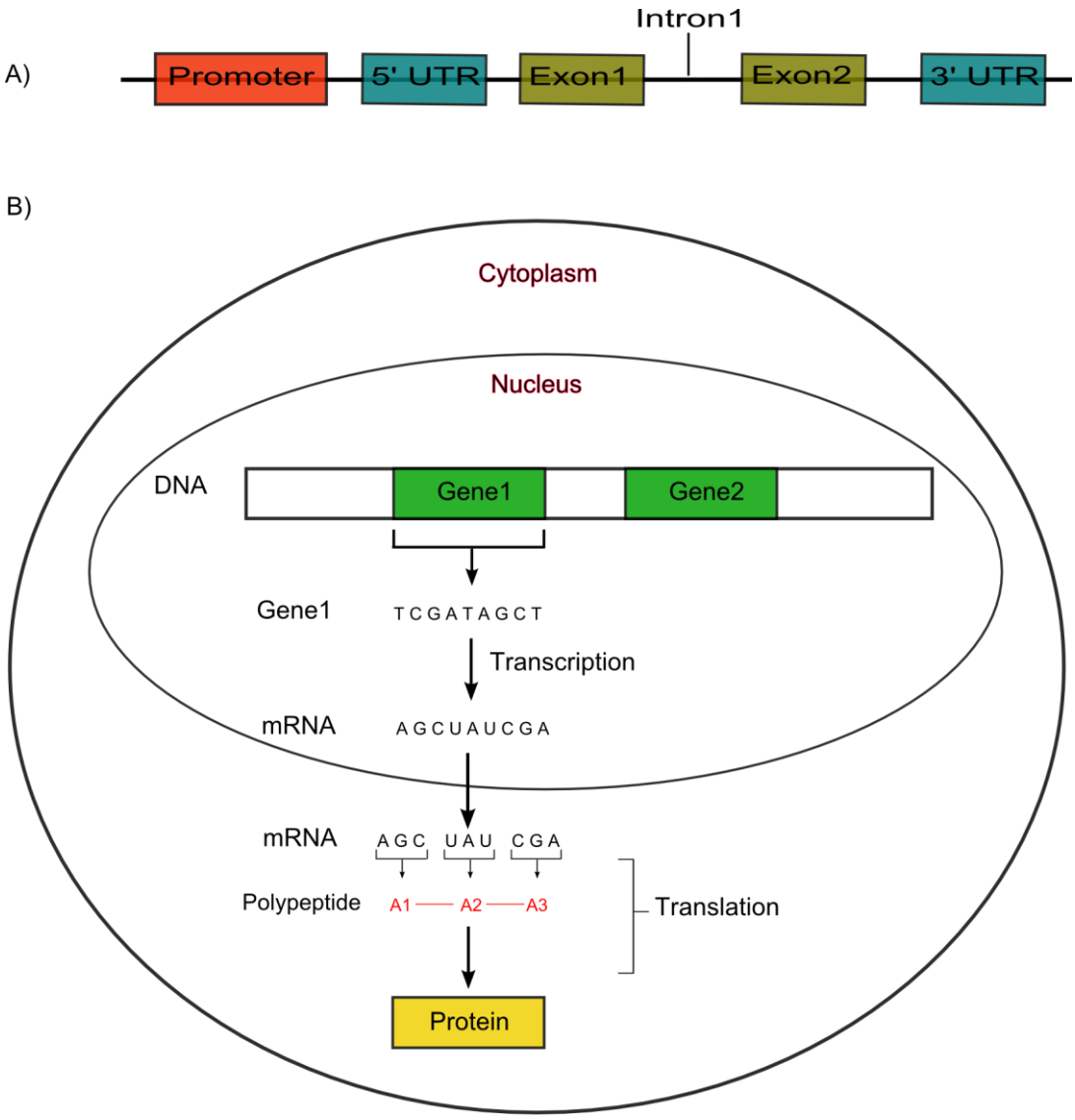

Figure 1.1: Simple gene expression.

A) An example of the structure of a gene is shown. The gene is composed of exons, introns, untranslated regions (UTRs) and a regulatory sequence known as the promoter. The 5' and 3' denote the directionality of the gene. B) DNA composed of genes is found in the nucleus of the cell. (The chemical backbone of DNA is composed of sugar and phosphate molecules.) A gene is transcribed into RNA, and then spliced into mRNA. The sequence of nucleotides in the mRNA is complementary to those in the gene (where A pairs with T, G pairs with C, C pairs with G, and U pairs with A). The mRNA molecule enters the cytoplasm and is translated into a chain of amino acids (A1, A2 and A3), constituting a protein. The protein may be a transcription factor involved in the regulation of another gene, including Gene1 itself.

During the process of transcription (See Figure 1.1B), the production of RNA (ribonucleic acid) sequence from the DNA sequence takes place, such that information stored in the DNA is transferred to the RNA. Like DNA, the RNA is made of nucleotides. The building blocks of RNA are the nucleotides adenine, guanine, cytosine, and uracil (U). Proteins, more specifically transcription factors, bind to a regulatory sequence of the gene known as the promoter region to initiate the transcription process and assist in the binding of the enzyme RNA polymerase. The RNA polymerase then assists in the synthesis of RNA [2].

After transcription, the RNA molecule may consist of coding and non-coding regions. The RNA that is eventually translated into protein is known as messenger RNA (mRNA). Thus, to produce a single sequence of amino acids needed for protein synthesis, the mRNA is processed by removing the non-coding regions and keeping the coding regions together to produce a mature mRNA molecule. (The process is called splicing.) The gene may also code for other types of RNA which assist in the process of translation: transfer RNA (tRNA) and ribosomal RNA.

Each group of three nucleotides in the mRNA codes for a single amino acid. (Amino acids are the building blocks of proteins.) The aim in the translation process, as shown in Figure 1.1B, is to read the mRNA and assemble all the amino acids that it codes for in the exact order as they appear in the mRNA. The end product is a linear sequence of amino acids, known as a polypeptide. The process of translation occurs in an organelle in the cytoplasm of the cell called the ribosome. The ribosome, with the assistance of tRNA

molecules, puts together the corresponding chain of amino acids from the mRNA molecule template. The polypeptide then "folds" to produce a very specific 3-dimensional shape, which is now known as a protein. Each polypeptide chain folds in a specific manner to deliver its unique functionality [2].

However, the protein may still not be ready to perform its function. Some additional post-translational modifications [1, 2] are further required to make the protein functional. This includes the addition of chemical functional groups to the protein, such as the addition of carbohydrates in glycosylation, the addition of lipids in lipidation, or the addition of phosphate groups in phosphorylation. In the case of glycosylation, for example, the carbohydrate groups act as identification tags which assist in cell to cell recognition.

We have briefly described the process of gene expression. The control of gene expression can be achieved by regulating any of the steps in the gene expression process. Much of the regulation depends on the control of the rate of transcription initiation. As discussed above, transcription factors are required to facilitate the transcription process. The different transcription factors that bind to the promoter region of a gene affect the binding ability of RNA polymerase to the promoter region, which in turn affects the rate of transcription. Gene expression can also be controlled by regulating the processing of mRNA. For example, the faster the non-coding regions of the mRNA are removed, the faster the mRNA is processed and the greater the quantity of gene product produced. Additionally, gene expression can be regulated by the rate at which the mRNA leaves the nucleus and enters the cytoplasm for the translational process. Moreover, the rate of gene

expression can be modulated by the rate of translation (the rate of protein synthesis), which is affected by the availability of amino acids or proteins [2]. Another means of RNA regulation is small interfering RNAs, which target specific RNAs after transcription and cause them to be degraded (thus pre-empting translation).

So, how is a gene's expression regulated such that it is only expressed when its corresponding protein is needed? As we know, each cell in the human body contains a complete strand of DNA, which contains all the genes present. However, not all the genes are expressed in all the cells. Depending on a cell's specialization, some genes are expressed, while others are not. As mentioned above, transcription factors affect the rate of gene expression (and thus the activity level of a gene). The transcription factors are themselves products of other genes in the DNA, which are also regulated by other transcription factors. Thus, certain genes are responsible for regulating the expression of other genes, and they may also regulate themselves. In this way, all the genes and proteins can be combined together to form a gene regulatory network (GRN). As seen in Figure 1.1A, each gene has a promoter, located upstream of the gene. Transcription factors which bind to the promoter region may either activate or repress the activity of the gene. Some genes may also be regulated by regions that are more distant from the transcription site. Distant regulatory sites may activate or repress (and thus can be called enhancers or silencers) or both (depending on what factors are bound there). A gene is turned on (or expressed) when the process of gene expression occurs and a protein product is produced. The turning off of a gene means that the gene is no longer able to synthesize its products.

Learning the structures of GRNs--that is, which genes which regulate which others--is an important and challenging problem in the field of genetics. The accumulation of data due to either the advancement of next generation sequencing technologies, RNA-seq or mass spectrometry, has made it possible to understand the organization of genes in genetic pathways. The data produced offers an insight of the genes' activities under various biochemical and physiological circumstances. Using the data collected, scientists are able to reverse engineer or infer the structure of the gene networks [3-9].

A GRN [10] describes a group of interacting genes, and is often conceptualized as a directed graph in which links indicate direct regulatory effect, and may be further categorized into activating or repressing. The absence or presence of genes in a network and their response to external stimuli (if any) influence the behavior of the network. To predict the behavior of a network, scientists may observe an output trait (or phenotypic consequence) produced by the genetic pathway in response to an external stimuli [11, 19, 20, 21, 22]. While many genes have been discovered, there still exists situations where the activity of the genes and the direction of regulation remain unknown. GRNs can be modeled in different ways depending on the features the researcher is investigating. They can be discrete or continuous, stochastic or deterministic, with feedback loops or without feedback loops and directed or undirected. In our work, each regulatory gene is regarded as a logic processing unit, which receives input and generates output. The combination of genes produces a gene network, i.e. a logical processing system. The architecture of the GRN, or the causal structure of the network, refers to the type and direction of the relationships between the genes. Although a genetic pathway may be composed of

different network substructures, each performing a different function, we consider a substructure composed of two genes.

There are many computational methods for the mapping of GRNS. One such technique known as epistasis analysis infers the network structure by considering patterns of change in an observable trait resulting from single and double gene deletions. Other techniques not in context of epistasis analysis for GRN inference do exist. However, while epistasis analysis limits the discrimination of networks to only the observation of the phenotypic consequence, other approaches involve the measurement of the expressions of all of the genes in the genetic pathway under study using methods such as microarrays or RNA-seq. One could argue about the possibility of observing the activity of the genes to discriminate between the networks. However, gene expression can be modulated by different factors, such as the transcriptional initiation, RNA processing and post-translational modification of a protein. Each of these factors may result in different forms of gene expression. Thus, the notion that is relevant to the function of a particular GRN is unknown. If one does not know which factor is relevant, the wrong form of the gene's expression may be measured, such that one may conclude that the gene does not affect the observable output trait of interest, when it really does.

Another reason the measurement of genome wide expression is not assumed in epistasis analysis is that epistasis analysis is often applied in the context of large screening experiments. In epistasis analysis, many single and double deletion mutants are tested, while for some of the proposed GRN inference methods double gene deletions are not

accommodated for. We have previously discussed the ability of microarrays, RNA-seq or mass spectrometry to measure the expressions of genes along the entire genome. However, measuring the expression of all the potentially relevant genes in their various conditions is not a feasible solution that could be performed by a present day lab. For example, let us assume that our two gene pathway structure is driven by an external stimuli. Naturally, expressions of normal genes in response to the presence or absence of the stimuli are measured. In epistasis analysis, additional expressions of genetically perturbed genes under single and double mutant deletions are also measured. Thus, in this case, each gene needs to be measured under 8 different conditions. While it may seem feasible for our substructure, GRNs are usually composed of many more genes. The number of combinations to be tested is exponential depending on the number of genes in the network and the number of external stimuli considered. As a result, in epistasis analysis, to infer the structure of the genetic pathway, we assume that we can only observe the activity of the output trait. Targeted follow up experiments of desired pathways could be conducted to directly observe the activity of every gene. The more accurate the pathway estimates, the more reliable the follow up experiments.

Avery and Wasserman [11] proposed the notion of epistasis analysis as a logical framework for the identification of biological pathways from the data collected. Figure 1.2 [26] displays the logic behind the standard epistasis analysis approach. As seen in the figure, epistasis analysis refers to the use of gene deletions to identify the interactions between genes. More specifically, it refers to situations where the outcome of deleting two genes is “surprising” compared to the results of deleting each gene individually.

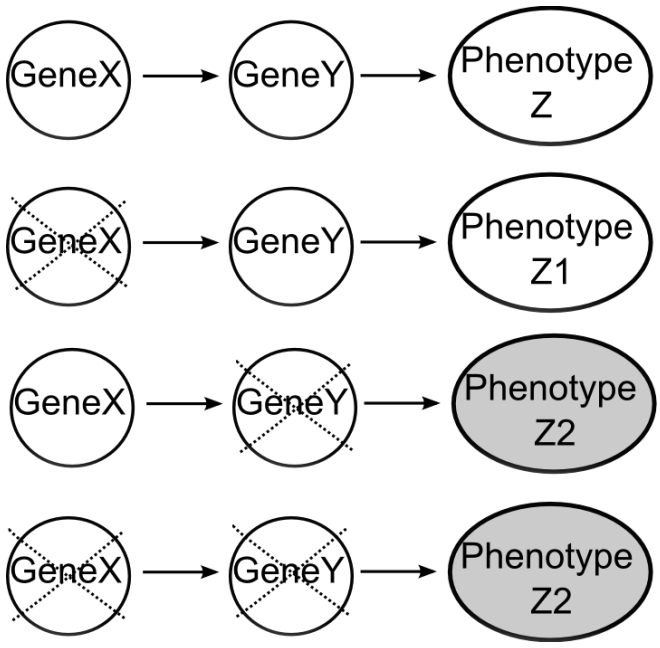

Figure 1.2: Classical epistasis analysis.

In the first row, both genes *X* and *Y* are in their wild-type forms. The middle rows represent the phenotype produced in the case of single knockout of each gene individually. In, the last row both genes are knocked out. One notion of epistasis occurs if the phenotype produced by the double mutant (*Z2*) is similar to that produced by one of the single mutants, but not the other. (Source: [26])

Consider Figure 1.2, for example. With no gene deletions, both genes *X* and *Y* are in their wild-type form, producing an expected phenotype *Z*. Individual gene deletions of *X* and *Y* result in phenotypes *Z1* and *Z2* respectively. Knowing the phenotypic consequences resulting from the single gene deletions, but not knowing the relationship between *X* and *Y*, we would not have any clear expectation of what might result when both *X* and *Y* are deleted. Suppose we observe, however, that the double deletion phenotype is *Z2*. In this case, genes *X* and *Y* are said to be epistatic to each other, as the outcome of deleting both genes is similar to the results of deleting *Y*. This is one notion of surprise in epistasis —

one gene's deletion masks the effect of the other's deletion. (Although as we will see in the next chapter, interpreting masking epistasis is not always as easy as in this example.) Another notion of surprise refers to situations where double deletions aggravate or alleviate the effect of one of the single gene deletions.

While classical epistasis has yielded deeper insights on numerous genetic pathways [11-29], it is not without limitations. One limitation is that it is not always possible to infer the relationship between a pair of genes using epistasis analysis. Similar input-output relationships can be generated by quite different networks, even when subjected to gene deletions.

Let us first consider the three hypothetical networks in Figure 1.3, which illustrate the relationship between an input signal *S*, two intermediate genes *X* and *Y*, and an output trait *Z*. An arrow in the figure represents activation. For example, the link $S \rightarrow X$ means *S* activates *X*. The signal *S* can either be on or off. Activating the signal eventually leads to the signal propagating to the output trait *Z*, and turning it on. Similarly, deactivating the signal results in the output trait being off. The deletion of either one of the intermediate genes (*X* or *Y*) detaches the signal from the output trait. The output trait is off for all the networks regardless of the signal state upon the deletion of the either one or both of the genes. Thus, these three networks cannot be uniquely identified using classical epistasis analysis.

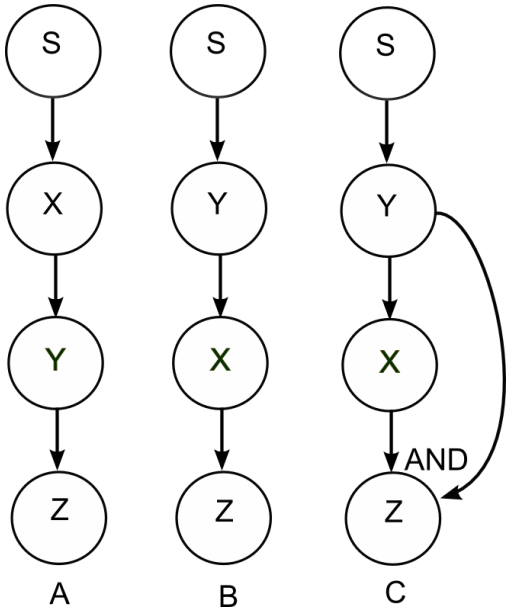

Figure 1.3: Example of three genetic pathway structures.

Three different genetic pathway structures that cannot be discriminated using static epistasis analysis or dynamics alone.

What happens if dynamics is introduced into the analysis? Suppose we drive the genetic pathways in Figure 1.3 with a dynamic signal *S* that is turned off for a long time, activated for the same period of time, then turned off again. Figure 1.4 displays the transitions of the output trait *Z* at each time step for the three network structures. For all the networks when the signal turns on, the output trait *Z* turns on three time steps later (at time step 7). However, there is a difference in when the output signal turns off again. The output trait in the first two networks (Figure 1.3A and B) takes longer to turn off. The signal has to propagate through three links before reaching *Z*. The direct link between gene *Y* and the output trait *Z* in Figure 1.3C, however, allows the signal *S* to propagate more quickly to the trait as it only passes through two links. Thus, dynamics alone

without any genetic perturbations could be used to discriminate the first two networks from the last network in Figure 1.3. However, dynamics alone still cannot be used to discriminate between Figures 1.3A and B, as in both cases, the signal flows through three links before finally turning off *Z*. We can conclude that neither classical epistasis analysis nor dynamics alone can discriminate between these two networks.

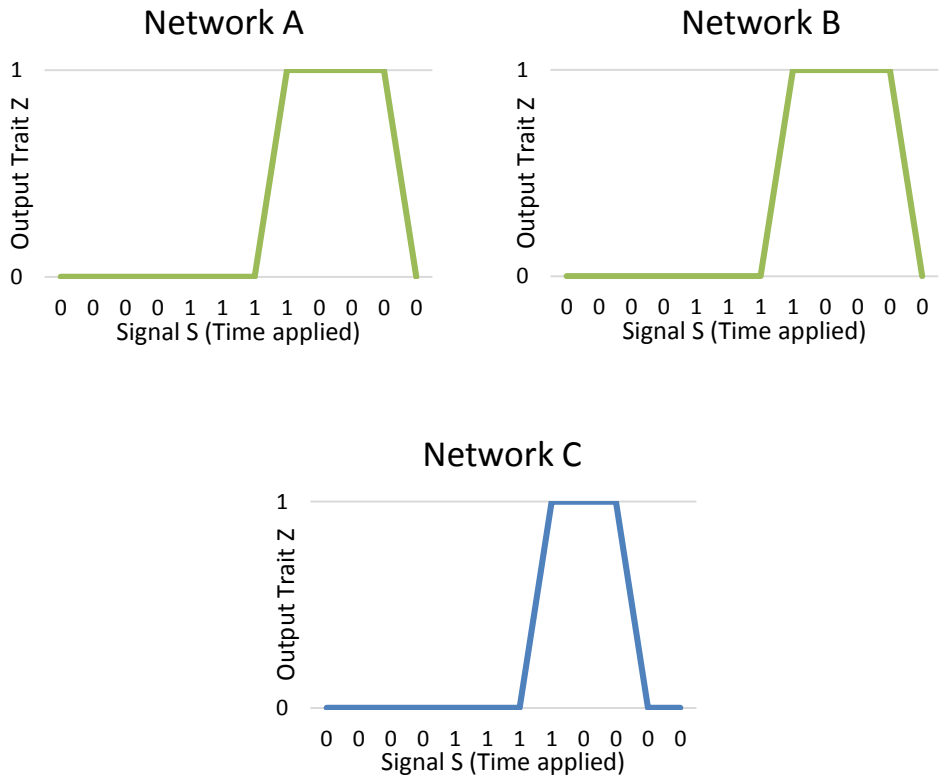

Figure 1.4: Line graphs to describe the effects of a time-varying input signal S on the three networks in Figure 1.3.

The signal is off for four time steps, then turned on for the next four time steps, and finally turned off. Graph A, B and C correspond to Figures 1.3A, B and C respectively.

Epistasis analysis has virtually always been applied to static data. The inference rules of Avery and Wasserman do not acknowledge the importance of dynamics in the input

signal *S*. The signal *S* is assumed to be constant in time, and thus the state of output trait *Z* has a constant steady state for the given input.

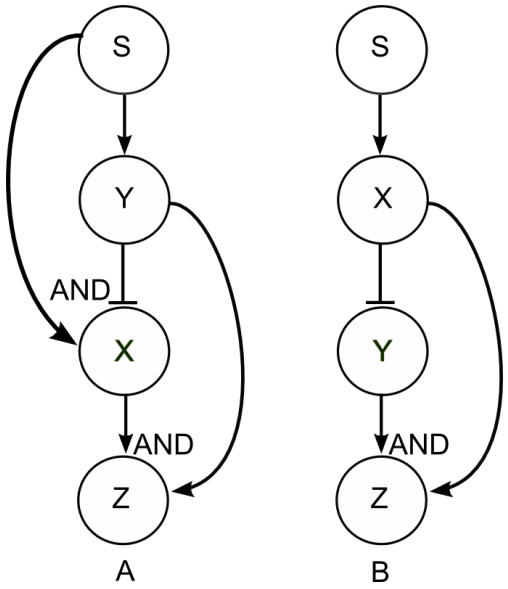

Figure 1.5: Example of a pair of networks.

A pair of networks that cannot be discriminated against using either epistasis analysis or dynamics alone.

Another example of a pair of hypothetical networks that cannot be discriminated by either approach is shown in Figure 1.5. A horizontal bar in the figure represents suppression. In Figure 1.5A, turning on the signal *S* results in turning gene *Y* on, which suppresses gene *X* and then turns off *Z*. Gene *X* is only activated when both the signal *S* is on and gene *Y* is off. Similarly, if the signal is off, *Z* is turned off. Figure 1.5B also results in the same input-output relationship. If the signal is on, gene *X* is activated, which in turn suppresses gene *Y* and turns off *Z*. If the signal is off, the output *Z* is also off. In Figure 1.5A and B, *Z* is on only when both *X* and *Y* are activated due to the presence of

the logical AND at *Z*. Thus, using epistasis analysis does not help us distinguish between the two network pairs. Both networks display the same behavior towards the input *S*. Dynamics alone also does not help, as in both networks, the output trait *Z* behaves in the same way as the signal propagates. In Figure 1.5A, when the signal is on and both genes are off, genes *X* and *Y* are activated at the next time step. (Gene *X* is activated because the signal is on and gene *Y* is off at the previous time step.) Gene *X* turns on briefly for one time step (until *Y* turns on), which causes the output trait *Z* to turn on briefly as well. Similar observations are seen in Figure 1.5B.

To our knowledge, only Azpetia *et al.* [26] have investigated epistasis in the context of dynamical networks with feedback. The authors argue that classical epistasis analysis may lead to wrong inferences when, in fact, there are dynamical feedbacks. The GRN in Figure 1.6 [26] is used to demonstrate this claim. Both genes *X* and *Y* receive input signals which activate them separately and both genes can activate themselves (with feedback loops). For example, after gene *X* activates itself, gene *Y* and the output trait are deactivated. Correspondingly, gene *Y* also activates itself, which leads to the suppression of gene *X* and activation of *Z*. The authors demonstrate that a steady signal for both *X* and *Y* (with the input signal for *X* always being off and the signal for *Y* always on) results in a fixed activated state for *Z*, which does not assist in identifying the network structure. Fluctuating the input signals for both *X* and *Y*, however, leads to variations in the activity of *Z*, which is determined by the state of the genes at the previous time step. Xu *et al.* [27] also describe a notion of dynamic epistasis, but not in the sense of temporal network

dynamics. Rather, their focus is on the distinct behaviour of different mutant alleles of genes, and the evolution epistatic interactions.

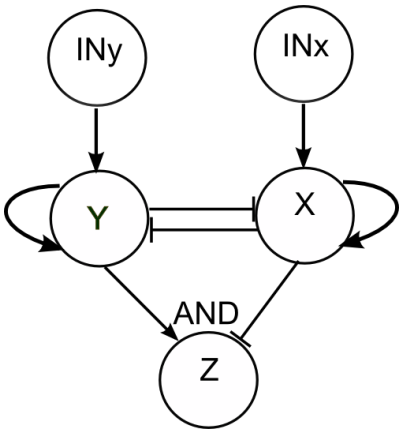

Figure 1.6: Example of genetic network.

A GRN which produces a different *Z* output pattern when either epistasis analysis or dynamic epitasis is used. *INy* and *INx* are input signals for the genes *Y* and *X* respectively. (Source: [26])

We believe that a dynamic input in addition to epistasis analysis is a powerful tool to discriminate between networks. The above observations motivate the main question of this study: How powerful is epistasis analysis if we are allowed to integrate dynamics into the experiment? We extend the classical approach by using a time-varying input signal, where the phenotypic consequence is measured at each time step, and quantify how helpful the addition of dynamics is by studying identifiability of two-gene pathway structures and individual links in those pathways.

The subsequent chapters are divided as follows. We begin with a literature review discussing previous related work on computational methods for GRN inference not in context with epistasis analysis. We then describe the related work on the classical epistasis analysis approach. Next, we define the theoretical framework used for identifiability analysis under different models of epistasis—static or dynamic, allowing or not allowing gene deletions (knockouts), and allowing or not allowing over-expression (knockins). We then use that framework to study identifiability of whole network structures, as well as individual network links, within different classes of networks. In the final chapter, we summarize our primary findings and point out avenues for future work.

2 BACKGROUND

In this chapter, we first refer to related work on GRN inference techniques not in reference to epistasis analysis. Then we describe the related work on epistasis analysis.

2.1 GRN Inference

Identifying the regulatory relationships between genes, including the direction of influence and the type of the relationship, is a fundamental challenge in molecular genetics. There are numerous computational methods for estimating GRN models, depending on the nature of the data available and the modeling formalism chosen [3-9].

Previous studies have investigated the possibility of using the unique input-output relationship of genes in a GRN to completely infer the structure of the network from time series data [3-9]. For instance, the expression levels of genes are experimentally measured as they change over time in response to an external stimuli, then transcriptional networks are reengineered from the data.

In Liang *et al*. [3], the authors conduct a simulation study of GRNS and develop a computational method, called REVerse Engineering ALgorithm (REVEAL), which models the simulated gene networks as Boolean models. A state transition table, which represents the expression of genes at each time step, is constructed for each model. The expression of genes at time t is compared to its expression at time $t+1$, which corresponds to their input and output states respectively. The state transitions of each

gene are analyzed by using the information theoretic principles of mutual information analysis for the gene's input and output states. The activity of a gene at $t+1$ may be affected by more than one input gene. As a result, if the mutual information analysis cannot explain all the state transitions with one input, the number of inputs per unsolved gene is increased iteratively until the state transitions of the gene can be explained with the maximum number of inputs per gene equal to the number of total genes in the network. The performance of the tool is tested by varying the number of genes and the number of inputs per gene in the randomly generated Boolean networks. The authors observe that the algorithm performs well with networks of smaller number of inputs (< 3) per gene. Inferring the network structure with a higher number of inputs is difficult and computationally intensive.

As gene expression is sometimes stochastic in nature, others have proposed learning a more general model system from the data, specifically a dynamic Bayesian network, which can model stochasticity, incorporate prior knowledge and handle hidden variables and missing data [4, 8]. Another approach which aims to construct the first draft of the topology of the entire gene network involves the use of singular value decomposition which generates a set of feasible solutions (i.e. networks) and the selection of the sparsest network using robust regression [5]. The assumption that naturally occurring GRNs in most biological systems are sparse such that each gene interacts with a small percentage of genes is made. However, since small gene networks cannot be regarded as sparse, this approach is not suitable for the fine tuning of small subnetworks which have a biological function. The increased complexity of cellular gene, protein and metabolite networks

motivated Gardner *et al*. [6] to develop a method called the network identification by multiple regression (NIR) for the rapid and scalable identification of the structure of GRNs with the use of no prior information on the network structure. NIR is a form of system identification based on multiple linear regression analysis of steady-state transcriptional profiles [6]. The authors claim that the algorithm is robust to high levels of measurement noise and correctly identifies key regulatory connections in a network.

Margolin *et al*. [7] develop an additional computational method, namely The Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE). The authors claim the available approaches suffer from limitations such as over-fitting, high computational complexity, dependence on non-realistic network models, or the need for additional data, and thus are only able to uncover interactions in simple prokaryotic organisms, such as the yeast *Saccharomyces cerevisiae*. As a result, the authors design ARACNE to specifically recover direct transcriptional interactions in more complex topologies, such as the mammalian GRNs, with high confidence from microarray expression profiles. ARACNE relies on the computation of the mutual information for all the gene pairs in a dataset and the elimination of most indirect interactions using a co-expression method, namely data processing inequality (DPI). DPI states that if two genes X and Y only interact through a third gene W, such that there is no alternative path between X and Y, then the least mutual information between all the gene pairs arises from the indirect interaction between X and Y. ARACNE utilizes this logic by examining each gene triplet and removing the edge with the smallest mutual information value.

Within the context of the DREAM (Dialogue on Reverse Engineering Assessment and Methods) project, Marbach *et al*. [9] conduct a critical performance assessment comparing numerous inference algorithms. The participants of the project were asked to predict the structure of gene networks using only synthetic gene expression data with no knowledge of the multiple benchmark networks (i.e. target networks). Gaussian noise was also added to the expression data. In addition to time series data, data on the knockouts and knockdowns of every gene were provided. The overall accuracy and performance of the methods on individual motifs were then calculated. In their summary, Marbach *et al*. [9] did not go into detail on how the different algorithms were developed. However, their overall results are the following observations. The authors claim that sophisticated techniques are not required to reliably infer the network structure and the outstanding quality of the winning team's predictions is mainly due to the performance of a simple method based on the model of the noise. The network motif analysis also showed that the overall quality of the network predictions are influenced differently by systematic prediction errors, such as the failure to distinguish between direct and indirect regulations or accurately infer the multiple regulatory inputs of genes. The overall results suggest that correctly inferring the structure of a network from gene expression data remains an unsolved issue. According to the authors, however, potential ways of improvement may include the use of a combination of reconstruction network methods.

2.2 Epistasis Analysis

Up to this point, a brief review of the many computational methods used for the reconstruction of GRN models from gene expression data was given. We now examine a classical method from genetics, which in its simplest form can be performed by hand — epistasis analysis [11, 12, 13]. Epistasis is a vital tool in functional genomics to enhance our understanding of the GRN components and their order of action [11-29].

In this subsection, we first describe the different notions of epistasis analysis: as masking or as aggravating/alleviating the effect of a gene. We then describe the different biological elements previous studies have considered in epistasis analysis. Finally, we list computational methods used for epistasis analysis.

2.2.1 Epistasis as Masking

As formalized by Avery and Wasserman [11], and as stated previously, classical epistasis is the identification of the structures of GRNs and the regulatory interactions between genes by examining the relationship between some "signal" and some "trait" or phenotype that we can observe. The trait is observed in different signal states in a wild-type organism and under conditions of single and double knockouts of genes in a pathway controlling the trait. Avery and Wasserman studied how the gene deletions impact the trait in the absence or presence of a signal to infer which gene is upstream of the other and whether it activates or represses the downstream gene. They also showed some cases in which the relationships between genes in a GRN can be inferred.

Consider a GRN with two intermediate genes *X* and *Y*, where the output trait *Z* is observed in the absence or presence of a signal. The two genes and signal can either be on or off, with no intermediate levels of activity. Avery and Wasserman deduced a set of general rules for epistasis analysis. According to their definition, epistasis occurs when the deletion of the genes *X* and *Y* looks identical to the deletion of one of the genes (say, *X*) but not the deletion of the other gene (*Y*). Hence, the notion of "surprise" here is that the single deletion of *X* masks, or obscures, the deletion of *Y*.

Avery and Wasserman illustrated their formulation in two examples (See Figure 2.1). They showed how one can correctly determine the ordering of genes in the sex determination and programmed cell death pathways of *Caenorhabditis elegans* using the inference rules they propose. Because their examples are so informative, and show the subtlety of inferring network structure from gene deletions, we reprise their arguments here.

Figure 2.1A, shows how the gene knockouts of *tra-1* and/or *her-1* produce different phenotypes which can be used to examine the effects of epistasis on a GRN. Sex determination in *C. elegans* is determined by X chromosome dosage. When there is only one X chromosome (XO), *her-1* is activated and *tra-1* suppressed, leading to the male development. Increasing the X dosage with two X chromosomes, however, (XX) results in the suppression of *her-1* and activation of *tra-1*, resulting in hermaphrodite development.

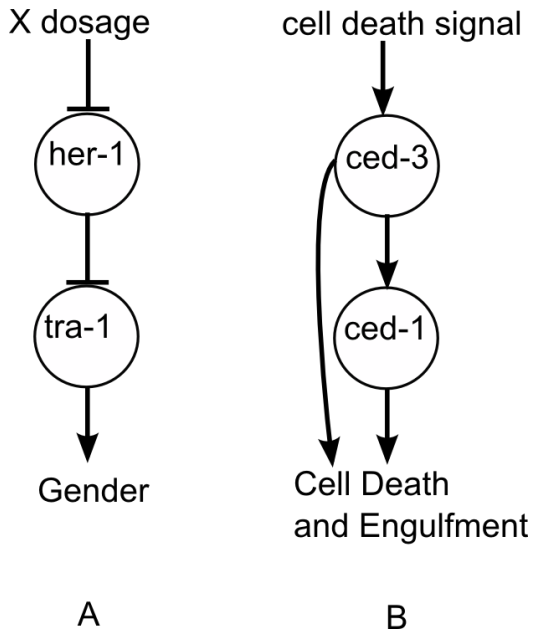

Figure 2.1: Examples of pathways examined by Avery and Wasserman. (Source: [11])

The gene *tra-1* is required for direct hermaphrodite formation. The knockout of *tra-1* leads to the formation of only males (both XO and XX are males), while the knockout of *her-1* produces only hermaphrodites. The knockout of both gene results in phenotypes produced similar to the phenotype of *tra-1* single mutants, i.e. only males are produced. In this case, the phenotypes of the individual mutants are different from the wild-type and from each other. Also, the phenotype of the double knockout is similar to one of the single knockout phenotypes. Examining the phenotypes generated, we can say that *tra-1* is epistatic to *her-1*, meaning that *tra-1* masks the effect of *her-1*. Thus, in this case, the downstream mutation is epistatic to the upstream mutation.

In contrast, Figure 2.1B show a network in which the upstream mutation is epistatic to the downstream mutation. The activation of *ced-3* leads to the activation of *ced-1* which results in cell death and engulfment. The knockout of *ced-3* leads to the cell remaining alive and unengulfed. Alternatively, the knockout of *ced-1* still results in cell death, but the cell is not engulfed as *ced-1* is required for engulfment. Double gene deletion of *ced-1* and *ced-3* results in the same phenotype as knocking out *ced-3*. Since *ced-1* cannot be activated without *ced-3*, we can say that *ced-3* is epistatic to *ced-1*. In the model of programmed cell death in *C. elegans*, the knockout of the *ced-3* gene masks the effect of the downstream gene, *ced-1*, in its pathway, yielding a different output than expected.

Avery and Wasserman illustrate the different epistatic effects of gene deletions on the output of a network. They also demonstrated that genetic knockins, which permanently activate a gene, can provide useful information for determining pathway structure.

2.2.2 Epistasis as Aggravating or Alleviating Double Deletions

A quite different notion of epistasis is when the double deletion produces an effect much stronger than we might expect from the outcomes of the single deletions. An example of this notion of epistasis can be found in the work of Tong *et al.* [14, 15]. Utilizing the gene deletion mutations for each of the known genes in S*accharomyces cerevisiae* created by the yeast genome project [16], Tong *et al.* experimentally study epistasis by using synthetic genetic arrays (SGAs) to identify synthetic lethal interactions (SSL) between individual genes. The SGA methodology involves the crossing of a mutated gene of interest into the entire genome of deletion mutants to produce a double mutant [14]. SSL

occurs when the combination of two mutations lead to cell death, when each single mutation alone does not. They conducted a large scale application of SGA, where 132 different genes mostly involved in actin-based cell polarity, cell wall biosynthesis and DNA synthesis and repair, were crossed into a set 4700 viable gene deletion mutants [14, 15]. The resulting double mutant descendant was then scored for fitness defects. SSL occurs when the fitness of the double mutant shows a significant deviation from the single mutants. The results of their experiment suggests that functionally related genes often tend to interact with each other. Also, genes which constitute the same genetic pathway or biological process tend to have similar patterns of genetic interactions. Similar results were observed when the SGA approach was further extended by Constanzo *et al*. [25] to examine 30% of the yeast genome.

The scope of quantitative epistasis analysis was later widened by the experimental investigation of other types of genetic interactions [17, 18]. SSL analysis is a specific case of a wider epistasis phenomenon. While synthetic lethality represents a qualitative feature (a double deletion results in death or it does not), synthetic sickness is quantitative (for instance, if growth rate is taken as a measure of cellular health). To further sub-classify the types of interactions, epistasis can either be positive (equivalently alleviating, antagonistic or buffering) or negative (equivalently aggravating, synergistic or synthetic). Positive interactions describe cases where a double mutation generates a phenotypic consequence less severe than expected. In this case, the double phenotype is healthier than the sickest single mutant and the genes which act together in a single complex or pathway will often have buffering interactions with each other. Negative interactions, on

the other hand, which include SSL, include cases where the double mutation produces a phenotypic consequence more severe than expected. In the specific case of SSL, as mentioned above, the phenotype of a double mutant results in cell death, an effect more severe than the phenotypes of the single mutants.

2.2.3 Phenotype, Input Signal and Species

The logical modeling of a gene network to infer the structure of individual genes or gene modules is made from purely phenotypic measurements, be it qualitative or quantitative. In addition to cell death [11], qualitative measures may also include the activation or inhibition of cell development [19] or the gender type [11]. Traditional uses of epistasis to order genes within a pathway have become progressively quantitative. Quantitative phenotypic measures involves using a method to quantify and measure genetic interactions for various phenotypic traits, such as growth rate (fitness) [20, 21], gene expression [21, 22] or unfolded protein response in endoplasmic reticulum of yeast [23].

Additionally, in the investigation of epistasis in a pathway, researchers need to decide the type of input signals to manipulate, such as the use of DNA damaging agents like MMS [20], cell starvation or the availability or scarcity of certain nutrients (such as the absence or presence of galactose [24]). While many explicitly integrate the signal in their study, some only exploit gene deletions to investigate the epistatic nature of the network. For example, Jonikas *et al*. [23] used gene mutations as a form of cell stress to characterize their functional dependencies.

Scientists have also embarked on cases of GRNs in different species. Yeast *S. cerevisiae* has been extensively explored [17, 18, 20, 22, 24, 25, 27, 28]. Other studies include growth and development of amoeba *Dictyostelium discoideum* [19, 22] and programmed cell death, sex determination or vulval development in *C. elegans* [11, 13]. In addition, research on epistatic interactions of genes in different conditional environments has been explored, such as the study of genes in yeast in glucose abundant and nutrient limiting conditions [24].

2.2.4 Computational Methods

The issue of increased complexity of data (phenotypes of single and double mutants) and lack of inference rules prompted the development of computational methods, which involve the use of the phenotypic consequence produced by the mutant organisms for the automatic inference of the structure of the gene network. Genepath [19] first implemented the genetic logic proposed by Avery and Wasserman using logic programming to construct gene networks based on qualitative phenotypes. Additionally, many have resorted to the use of the hierarchical clustering approach which exploits quantitative phenotypic measures to group genes with similar phenotypic consequences together and then identify the type of relationships between the groups of related genes [17, 18, 20, 22, 23]. Methods have advanced from roughly grouping the genes based on their phenotype into functional groups to analyzing detailed structures of the relationships between genes and revealing the functional dependencies. For a more detailed analysis of the gene pathways, others adopt a Bayesian learning approach [29, 30] where a large set of genes and their functional dependencies are represented by an ensemble of activity

pathway networks (APNs) and APNs with the highest confidence are selected for further analysis. Many have also adopted the utilization of mathematical and logical models to represent the genotypic-phenotypic relationship, such as flux balance analysis [17, 24, 27], generalized linear model regression [20] and multi linear regression [21].

3 THEORETICAL FRAMEWORK

Here, we describe the theoretical framework used in our comparative study of static and dynamic epistasis. The chapter is divided as follows. We first explain Boolean network models and how they are used to represent GRNs. Next, we give details on the different classes of networks generated for our analysis. After which we describe how to run a simulated experiment on a network. We then explain the set of experiments conducted on each network structure. Finally, we describe the types of identifiability analysis performed to quantify how helpful the addition of dynamics is to epistasis analysis.

3.1 Boolean network model

There are many methods to model GRNs [11-31]. GRN models can either be: stochastic or deterministic, directed or undirected, with discrete or continuous expression, with discrete or continuous time, and with feedback loops or without feedback loops. The main interest for this thesis is to determine the advantage of using a time varying signal in epistasis analysis. Thus, in our theoretical investigation, we model GRNs in one of their simplest forms--as Boolean networks. Kaufmann [31], in the 1970s, investigated the organization and dynamic properties of Boolean networks as random models of GRNs. Many studies then further investigated the use of Boolean models to represent molecular and genetic networks [32-36]. For example, Boolean network models were used to analyze normal and neoplastic cells' cycles in cancer biology [33] and yeast transcriptional networks [35].

The Boolean network is advantageous for many reasons. Being a fully discrete model, the Boolean model makes it possible to enumerate all possible networks of a given size. Additionally, as a deterministic model, our main concern is whether a network is consistent or not with experimental observations. There is no "degree of match" as we would have with a probabilistic and/or continuous model. Moreover, as time is discrete, we can enumerate dynamic signals. With a Boolean model, we also do not need to account for noise or observation accuracy, as we would with a continuous model. We believe that the use of a Boolean model can be the first step to determining relevant global features that may be found in real biological data (gene expression). The simplicity of the Boolean network model to represent GRN may provide a dependable guide to the behaviour of similar systems with more complex behaviour (with continuous, probabilistic functions, etc.) [31-36]. Studies suggest that while the gene expression may be continuous, there is usually a cutoff which can be used to classify genes as on or off [32, 34]. Many studies have also used Boolean variables to represent genes to accurately infer the structure of GRNs [11-36].

A Boolean network [34] $G = (V, F)$ is a directed graph which consists of a set of nodes $V = \{v_1, v_2, \dots, v_n\}$ together with a set of Boolean functions $F = \{f_1, f_2, \dots, f_n\}$, such that for $k \leq n$:

$$f_i : \{0,1\}^k \rightarrow \{0,1\}$$

Each node v_i has a Boolean function f_i associated with it. A function determines the state of each node v_i. If a particular node v_i does not have any k inputs, then the state of the node will remain unchanged. If k > 0, the inputs of the Boolean function f_i for the node

v_i are the set of input nodes directly connected to it and thus the state of v_i is determined by its inputs. The nodes of a Boolean network are deterministic, meaning that for a given input, the same output is always produced. The topology of a network, which refers to the nodes and the connections between them, can be determined from the set of functions of all the nodes in the network. Consider Figure 3.1, the network consists of three nodes A, B and C. The set of functions per node ($f_A(\emptyset) = \emptyset, f_B(A, C) = A\ AND\ C, f_C(A) = A$), give an indication to the number of inputs per node and the type of relationship between them. With these functions, for example, we know that A has no inputs and B receives input from A and C, such that both A and C need to have a value of one, for B to be one.

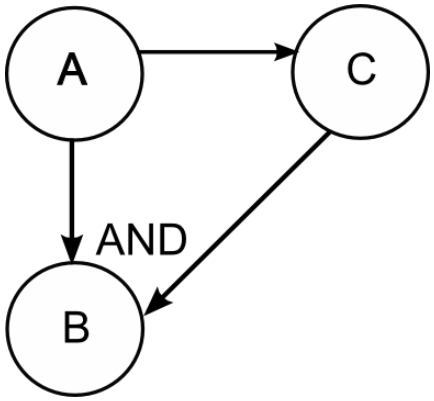

Figure 3.1: Example of a Boolean graph.
This Boolean graph has nodes $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$, where $\boldsymbol{f_A(\emptyset) = \emptyset, f_B(A, C) = A\ AND\ C, f_C(A) = A}$.

The time and state variables of a Boolean network are discrete. For any Boolean network, if the state of a node v_i at time t is given as $x_i\,(t)$. The state of the same node at time $t+1$ is denoted as

$$x_i\,(t+1) = f_i\,(x_{i1}, x_{i2}, \dots, x_{ik})$$

where x_{ij} are the states of the input nodes directly connected to v_i. The equation above implies that the state variable updates are done synchronously, which is the case for classical epistasis.

As mentioned previously, epistasis analysis involves the observation of an output trait Z in response to an input signal S and genetic perturbations of the intermediate genes X and Y. Although these may comprise a subnetwork which is part of a larger network, our study investigates only the elements of the subnetwork: S, X, $Y$and Z. The nodes in our Boolean model are the biological variables: S, X, $Y$and Z, the directed edges demonstrate the relationship between them and the logical functions determine the state values of the variables depending on the states of their inputs.

In our study, we assume that we can control the input signal S in an arbitrary time-dependent manner, so that the input to the network is the time-varying $S(t)$. We also assume that the time takes discrete values, $= 1, 2, 3 \dots$, and at each time step, each of the four state variables takes a Boolean value. In the case of classical epistasis analysis, the input signal S is fixed in any given experiment, meaning that the signal is either always on or always off. The control of the signal S is experimentally possible when the signal

represents an experimental variable (or external stimuli) that is easily manipulated, such as the presence or absence of a drug or nutrient source in the medium in which cells are being grown. To represent cases with limited control over $S(t)$ we also investigate the extent to which limited dynamical changes in S can be beneficial to epistasis analysis.

The four state variables can have the value of 1 or 0. In the case of S, 1 denotes that the signal is on, while 0 means that it's off. For example, in an experiment where the drug dosage represents the signal, in the absence of the drug, the signal is off, and in the presence of the drug, the signal is on. The values of the intermediate genes X and Y represent the activity of the genes, where an active or inactive gene can have the value of 1 or 0 respectively. A knocked in (KI) gene is gene that is over expressed (always on). We assume that a gene can be in one of three forms: inherent (or wild-type), deleted or knocked out (KO), or knocked in. Inherent genes represent the typical forms of the genes as they appear in nature. The activity of each inherent gene in our model can be expressed as follows:

$$X(t+1) = f_x\big(S(t), Y(t)\big),$$

$$Y(t+1) = f_y\big(S(t), X(t)\big),$$

where the expression of a gene is determined by the gene or signal directly affecting it. As stated above, we also allow the genes in some experiments to be altered to take fixed values. The deletion of the genes X or Y eliminates the gene and any function it might have. They are modeled by:

$$X(t) = 0 \; or \; Y(t) = 0,$$

for all times t, such that the gene is always off rather than following its wild-type dynamics. For the case of knockins, the genes are expressed as:

$$X(t) = 1 \; or \; Y(t) = \; 1,$$

such that the gene is always on. Performing such genetic manipulations is not always trivial, but there has been a great deal of genetics research for centuries, and there are many well-established means for doing so.

The output trait Z is assumed to follow the Boolean dynamics at all times and in all experiments. A value of 1 denotes that the phenotypic consequence of a genetic pathway is observed or measured to be above a specified threshold, while 0 means that the trait is not observed or measured to be below a specified threshold. (The threshold is specified by the scientist conducting the study.) The activity of Z may depend on any or all of the other three biological variables S, X and Y, so that:

$$Z(t+1) = \; f_z\big(S(t), X(t), Y(t)\big).$$

The indegree of a gene or the output trait is the number of variables directly affecting it. Each input can appear directly or negated, in which case it is considered activating or repressing respectively. Let us first consider when the indegree of a gene/output trait is one. For example, when $X(t+1) = \; S(t)$, the gene X is activated when the signal S is on at the previous time step. Conversely, if $X(t+1) = \; NOT \; S(t)$, the gene X is activated only if the signal S is off at the previous time step. In this case, we say S represses X. In situations when there is more than one input to a gene/output trait, we restrict f_x, f_y and

f_z to be the logical AND or logical OR. For example, if the rule for Z's dynamics is $Z(t+1) = AND(NOT\ S(t), X(t), NOT\ Y(t))$, then we would say Z is repressed by S, activated by X and repressed by Y, and that Z turns on only when X is on at the previous time step, and neither S nor Y are on.

We follow the assumptions used by most of the previous work on epistasis analysis regarding the type of networks used. We assume that all the networks generated are acyclic. For example, there is no cyclic dependency between X and Y meaning that a connection between the two genes, if any, is only one direction. Auto-regulation is also not allowed. In classical epistasis analysis [11], these assumptions are made so that the output trait Z comes to a fixed, steady state value for any given fixed value of the input S, and that the steady state value is independent of the initial state of the network. For our project, we remove the restriction of observing only steady state traits. However, we retain the acyclicity assumption to avoid initial state dependence.

3.2 Network Classes

In this section, we describe the classes of networks used in our simulation study. We simulate more than one class of networks, from the most general which includes all the acyclic networks, to the simplest which consists of only 16 networks. The general network class represents situations in which scientists have less knowledge about the GRN under study. The more specialized class networks are for situations where scientists have an idea about either the type of interaction between the genes X and Y (in the case of Linear or LinearPlus) or the effect of knocking out individual genes in the pathway (in

the case of Single Knockout Visible). Classes of networks of size greater than one are generated to observe what type of information (static or dynamic epistasis and with or without genetic perturbations) would be helpful to discriminate between the different network structures. Simulated experiments are conducted on all networks in each of the network classes.

For a detailed analysis of GRNs, we generate all the possible network topologies of two genes X and Y which regulate the state of an output trait Z when driven with a signal S. In our diagrams, an arrow and a horizontal tangent to the node denote activation and suppression respectively. We use all of six rules below to describe all the possible interactions in a network:

1. Influence of the signal S on gene X.
2. Influence of the signal S on gene Y.
3. Influence of the signal S on output trait Z.
4. Interaction between genes X and Y.
5. The effect of gene X on Z.
6. The effect of gene Y on Z.

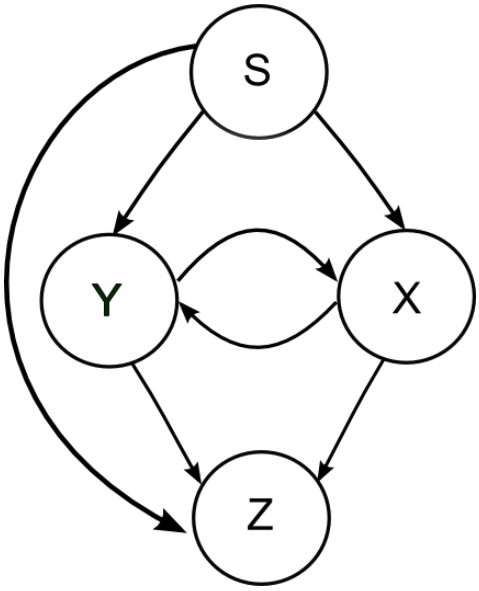

Figure 3.2: Network structure with all the possible links allowed. In this figure, only activating links are shown. However, repressing links are also allowed.

Figure 3.2 displays all the possible links, not all of which are present in the same network. There can be an activating link between S and X (shown in Figure 3.2), as well repressing and absent links. The remaining links, excluding the $X - Y$ link, can have the same three choices. In addition to having no link, the $X - Y$ link can have activating or repressing links in either direction. Additionally, for all the network structures created, we assume that we have full control over the signal S and any output from Z cannot be used as input for any of the other variables.

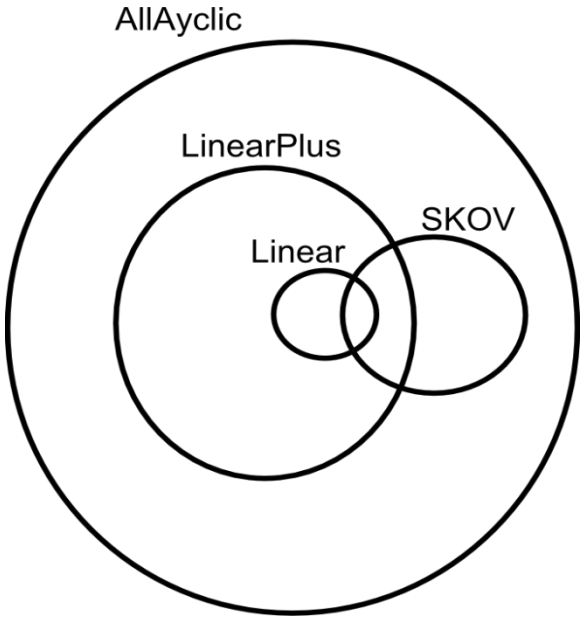

Figure 3.3: A Venn Diagram to represent the relationship between the different network classes. The AllAcyclic class is a superset of the other classes.

We categorize the GRNs into four different classes of networks. The large number of networks in the classes are due to all the possible links we allow. Figure 3.3 shows the relationship between the different network classes using a Venn diagram. Below, we describe the characteristics and assumptions used in each class:

- Linear Networks

This is the simplest and smallest group of networks. It consists of all networks with only three links connected as $S \to X \to Y \to Z$ or $S \to Y \to X \to Z$. Each link has two options, it can either be activating or repressing. With three links per network, two choices per link and two choices for the order of X and Y, there is a total of $2(2^3) = 16$ networks in

the Linear class. Figure 3.4 enumerates all Linear class networks, where the top row A differs from the bottom row B in the order of X and Y.

- LinearPlus Networks

The networks of the Linear class are a subset of this class. The LinearPlus class allows additional feedforward links. Consider the Linear chain $S \rightarrow X \rightarrow Y \rightarrow Z$ in Figure 3.4A, for example. Additional links would include links from any upstream variable to any downstream variable such as $S \rightarrow Y$, $S \rightarrow Z$ and $X \rightarrow Z$. Each of the links may be activating, repressing or absent. For every Linear network structure, there are 85 possible LinearPlus networks. Two logical possibilities (logical AND or logical OR) are defined for each node with multiple inputs. By straightforward enumeration, we found the LinearPlus class contains 1360 distinct networks.

LinearPlus is a more complicated class than the Linear class of networks. It allows more complex behaviour for the output trait Z. In both the Linear and the LinearPlus network classes, X and Y are both on a genetic pathway that leads from S to Z, but their ordering and the nature of their direct relationship, activating or repressing, is unknown. Thus, these network classes model situations in which we already have some strong evidence that two genes, X and Y, collaborate in a pathway.

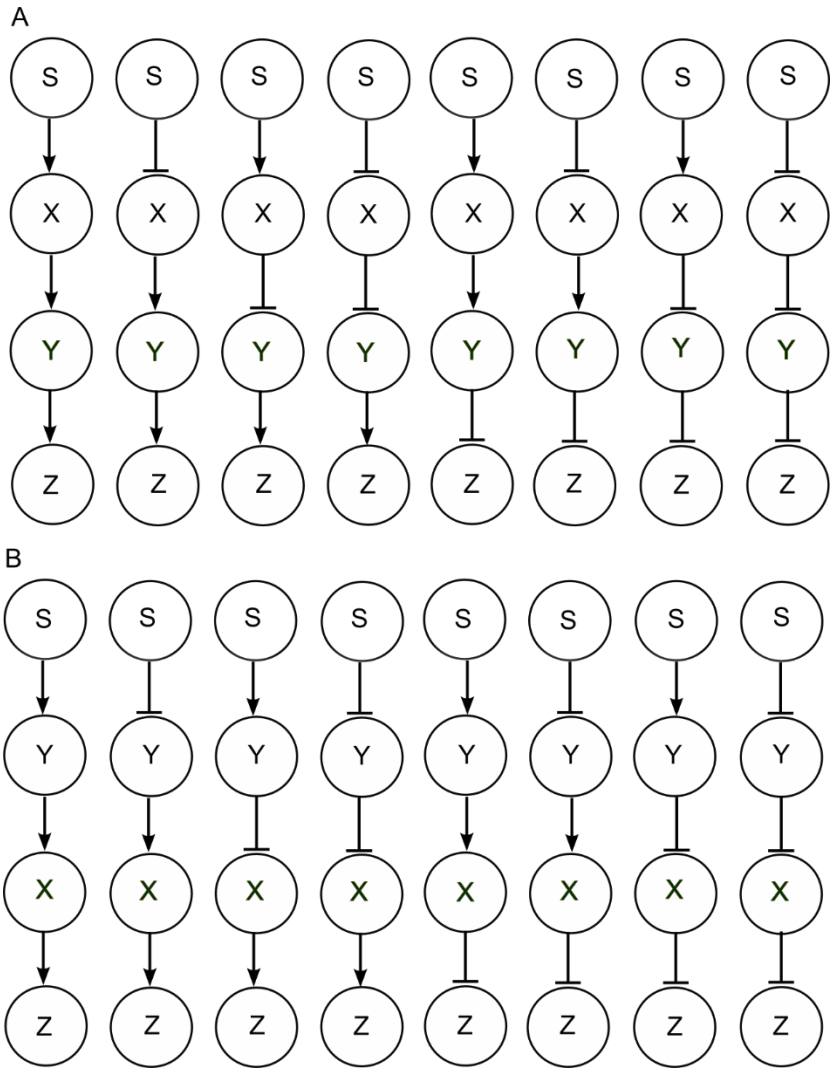

Figure 3.4: All the networks that make up the Linear network class. There is a total of 16 different network structures**.**

- AllAcyclic Networks

This is the largest network class (a superset of all the other network classes), and comprises all acyclic networks with the four variables S, X, Y and Z. This class has only the two basic restrictions found in every class of networks considered: S is controlled by the experimenter (does not take any inputs) and Z does not provide input to any other

variable. The relationship between X and Y is unknown such that the link $X - Y$ could either be absent or present. Whether S influences the trait Z is also unknown. Thus, the AllAcyclic class models situations where the experimenter has very limited knowledge about the network structure. This class may be used to represent cases of high-throughput screens, where one is simply interested in genes or stimuli that might influence a trait. By straightforward enumeration, we found the AllAcyclic class contains 3243 distinct networks.

- Single Knockout Visible (SKOV) Networks

This group of networks is more restricted than the LinearPlus class and is a subset of the AllAcyclic network class. It is motivated by a common approach to detecting pathways that are affected by single gene deletions of X or Y by identifying the candidate genes that may be in a pathway. For example, assume that the signal is always on. The output produced when both genes are in their wild-type form is compared to the output produced when gene X is individually knocked out. Suppose you also compare the outputs produced when the signal is always off. If for either situation, there is a difference in the outputs produced, we can say that gene X is "single knockout visible". Additionally, suppose you conduct the same comparison with individually knocking out gene Y instead of X. If there is a difference in the outputs produced in the individual knockout of both genes X and Y, then we can say that the network is "single knockout visible".

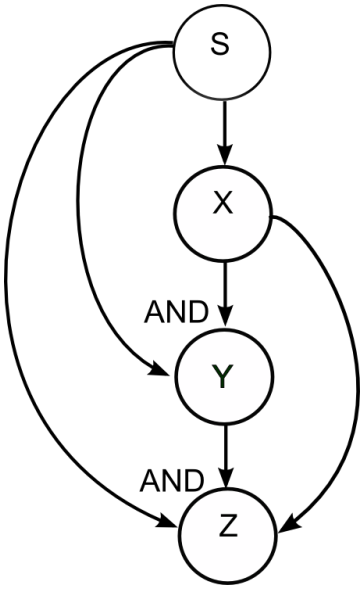

Figure 3.5: An example of a SKOV network structure.

Consider the network in Figure 3.5. The signal S activates X, Y and Z, where gene Y receives input from S AND X and the output trait Z receives input from S, X AND Y. If S is on and both genes X and Y are in their wild-type form, Z is on. However, the deletion of either gene X or Y individually, while the signal is on, results in the turning off of Z due to the logical AND present at Z. The contradicting effects on the output trait with and without the gene deletions of either X or Y makes the pathway "single knockout visible". That is, a single gene deletion of either gene helps in the understanding of the order of genes in the pathway, and thus the genes would be flagged in a single knockout screen. A follow up to such a screen might be to do a more detailed epistasis analysis, to try to understand how the different flagged genes are related. By straightforward enumeration, we found the SKOV class contains 220 distinct networks, the second to smallest network class.

3.3 Running a simulated experiment on a network

As stated above, we are considering four different classes of networks. To determine the type of data we can obtain from a network using different input signals and genetic perturbations, we run a set of experiments on each network structure in each of the four network classes. Here, we describe how we run a simulated experiment on a network structure.

Consider the network structure in Figure 3.6A. An *experiment* here is defined as driving this network with an input signal S, such that

$$S(t) \in \{0,1\}\, for\, t\ \in \{1, \dots, t_{max}\},$$

where t corresponds to the time steps, $t_{max} \geq 4$ is the final time step in the experiment and the signal S at each time step could either be 1 or 0. The output trait Z is observed at each time step t to study the effect of the signal on the genetic pathway. In the first three time steps, the value of the output trait depends on the initial conditions for the genes X and Y. The networks have a depth of at most three links to Z, thus it takes at most three time steps for the input signal to reach Z. In our study, to avoid complexity, we assume that we cannot control or observe the values of X and Y and thus cannot consider any dependence of their unknown states in the data. Hence, since at most three time steps are needed to ensure that the value of the output trait is fixed in the networks we study, each input is held for four time steps to ensure that Z is at a steady state value for a constant signal S.

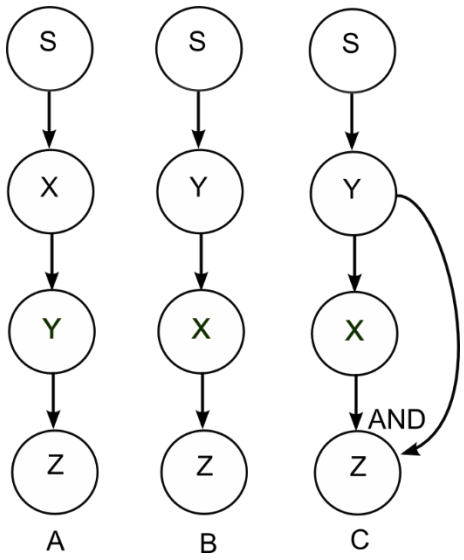

Figure 3.6: An example of three hypothetical GRN structures.

For the input signal $S(t)$, we consider three options:

- Static Signal: The signal S is constant at zero, $S(t) = 0$, or constant at one, $S(t) = 1$, in every experiment conducted.

- Step Dynamics Signal: We drive the network with a time-varying input $S(t) = (0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1)$, where the signal steps from zero to one to zero, and then back down to one (See Figure 3.7). We observe how the step dynamic signal propagates through the network and alters the trait dynamically from the fourth time step onwards. The outcome of an experiment is $Z(t)$ for $t \in \{4, \dots, t_{max}\}$.

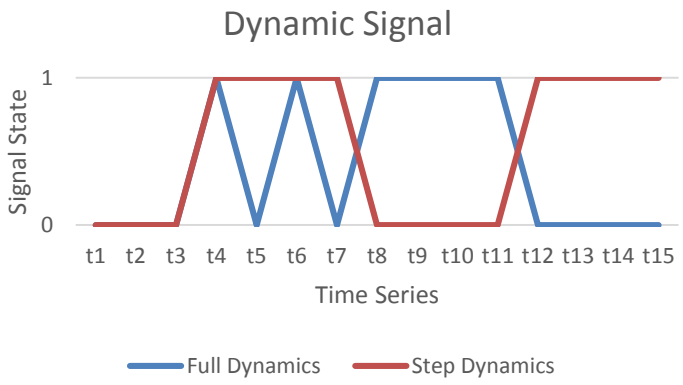

Figure 3.7: Line graph to represent how both signals propagate through time.

- Full Dynamics Signal: Here, we drive the network with the time-varying input $S(t) = (0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0)$, as seen in Figure 3.7. The reason for this particular dynamic sequence is that the networks have a depth of at most three links. Thus, it takes at most three time steps for an input signal to reach the trait. The output trait $Z(t)$ can be written in term of $S(t-1)$, $X(t-1)$ and $Y(t-1)$. The intermediate genes $X(t-1)$ and $Y(t-1)$ can then be further written in terms of $S(t-2)$ and/or $S(t-3)$ depending on their order and which gene is upstream the other. Thus, the output trait can always be written in terms of the signal of S, with $Z(t) = S(t-1)$, $S(t-2)$ and $S(t-3)$. The mentioned sequence for the input signal contains all the possible 8 triplets of values: (000, 001, 010, 011, 100, 101, 110, 111). Thus, although we could consider other dynamical signals, none could provide any more information than the one we have chosen. We test how the signal can influence Z in every possible

way. Again here, we only observe the output trait Z at the fourth time step to avoid dependence on the initial values of the intermediate genes X and Y.

In addition to the type of input signal we choose to drive the genetic pathway, there are three options for each of the genes X and Y. As stated previously, they will either follow wild-type (WT), knock-out (KO) or knock-in (KI) dynamics. Thus, an experiment can be fully defined as:

$$E = (S(1), \dots, S(t_{max}), X_{status}, Y_{status})$$

where $X_{status}, Y_{status} \in \{WT, KO, KI\}$.

3.4 Set of Simulated Experiments Per Network

In the above section, we describe what is meant by an experiment and the types of input signals and genetic perturbations we allow. Here, we explain the set of experiments conducted per network. For each network, there is a set of $27 = (3 \times 3 \times 3)$ different experiments, with 3 choices for the signal S, 3 choices for the genetic perturbations of X and 3 choices for the genetic perturbations of Y. We organize them such that we consider nine modes of analysis for each network structure which we describe below. Static epistasis refers to when the signal is fixed, while dynamic epistasis refers to when the signal is time varying for both step or full dynamic signal inputs.

- Static Epistasis with Wild-Type: We observe the trait under two constant signal conditions: signal on or off, gene X wild-type and gene Y wild-type. Thus, there a two

different effects on the output trait Z. A pair of networks can be discriminated from each other if they have different output traits under these two conditions.

- Static Epistasis with Knockouts: Both genes X and Y in this case can either be wild-type or deleted. Thus, these two choices for each gene and the two choices for the static signal (either always on or always off), generate 8 different conditions which can influence the output trait Z differently. We observe the trait under each of these conditions. The 8 resulting Boolean values of Z define the output behaviour of a network. Under this analysis, two networks are considered distinguishable if they have different output trait values under any of the 8 conditions. An example of a complete truth table for the graph network in Figure 3.6A under classical epistasis analysis is shown in Table 3.1, where the output trait Z is observed for all different combinations of the genes' states and signal states.

Table 3.1: Truth table for the network structure in Figure 3.6A

S	WT or KO		Variable Values		
	X	Y	X	Y	Z
Off	KO	KO	0	0	**0**
Off	KO	WT	0	0	**0**
Off	WT	KO	0	0	**0**
Off	WT	WT	0	0	**0**
On	KO	KO	0	0	**0**
On	KO	WT	0	0	**0**
On	WT	KO	1	0	**0**
On	WT	WT	1	1	**1**

- Static with Knockouts and Knockins: Here, there is an additional option for each of the genes X and Y, the gene can be knocked in (always on). For each gene X and Y, there are three choices: wild-type, knocked-out or knocked-in, and the signal S again has 2 choices: on or off. Thus, this results in 18 different conditions under which the trait Z is observed.

- Dynamics with Wild-Type: Neither gene X or Y are knocked out or knocked in; they are both wild-type. However, the signal S changes dynamically, and the output trait is observed at each time step. Since the length of the signal under which we observe the output trait equals 12, there are 12 output trait values which we can use to infer the network structure.

- Dynamics with Knockouts: The dynamics is similar to what was described previously. However, in this case, both genes X and Y may either be knocked out or in their wild-type form. With two choices for each gene and 12 time steps in the signal S, the value of the output trait Z is observed under 48 different conditions.

- Dynamics with Knockouts and Knockins: Again, here an additional option of knocking in the gene is considered. Each gene X and Y, can either be knocked out, knocked in or wild-type. Thus, there are 9(12) = 108 output trait values for network inference.

3.5 Identifiability Analysis

Up to this point, we have described four classes of networks. We have also considered how for each network structure in each network class we will observe the value of the output trait under different experimental conditions. Here, we describe what is meant by the identifiability of network structures and individual links.

3.5.1 Network Identifiability

Consider the pair of network structures B and C in Figure 3.6. To analyze network identifiability for this example, we observe the effects of the signal on the output trait for both networks under the different experimental conditions discussed above. This pair of networks is *distinguishable* by an experiment if the output trait of B is different from the output trait of C at any time $t \geq 4$ during the experiment. We say that this pair of networks is *distinguishable* by a set of experiments $E = \{E_1, \ldots, E_m\}$ if the output trait of B is different from the output trait of C for any of the experiments. The pair of networks are *equivalent* if the networks are not distinguishable by any of the experiments. For example, the networks B and C generate the same output trait value under static epistasis analysis with wild-type and single and double gene deletions. Thus, we can say that networks B and C are equivalent under this experimental condition. On the other hand, under step dynamics analysis with wild-type and single and double gene deletions, these networks produce different output traits, and are thus distinguishable.

After the simulation of the output trait values for each network in each network class under the different experimental conditions, we study the ability of a set of experiments

to discriminate between not only a pair of networks, but a whole set of networks $N = \{N_1, \ldots, N_n\}$, as we are not only interested in the unique identification of a network. Thus for a set of networks, we introduce the notion of equivalence classes. An equivalence class is group of networks which produce the same output, that is have the same effect on the output trait. The use of any experimental condition divides the set of networks N into equivalences classes, such that:

$$N = Q_1 \cup Q_2 \cup \ldots \cup Q_k,$$

where N is the collection of all networks, as described above, Q_i represents an equivalence class, and $Q_i \cap Q_j = \emptyset$ for $i \neq j$. Equivalence classes here may include single or multiple networks. In an ideal situation, all the networks are identifiable, and thus fall in their own equivalence classes. An example of an equivalence class with multiple networks is shown in Figure 3.8, which shows a group of six LinearPlus networks that produce the same output under static epistasis analysis with knockouts.

3.5.2 Link Identifiability

Uniquely identifying an entire genetic pathway is the best possible outcome. However, since we may not always be able to identify a network uniquely based on the output trait observations, the identification of individual links in a network is a more tractable goal. Let Q be the equivalence class for the group of networks in Figure 3.8. If Q contained a single network, then all links in that network could be identified. In this case, where Q contains multiple networks, a link can be identified if all networks in the class contain the same type of link—activating, repressing or absent. Consider the link from S to X in the networks in Figure 3.8. Figures 3.8A, B, C and E have an activating link from S to X,

while Figure D has a repressing link and Figure F has no link from S to X. Thus, the link is not identifiable for the networks in Q. However, consider the link from S to Z in all the networks. We notice that the link is identifiable as for all the networks there is always an activating link from S to Z. For a given network class N and experimental condition E, we will report the fraction of times an activating/repressing/absent link can be identified as the total size of all equivalence classes where all networks have that link, divided by the total number of networks that have that link.

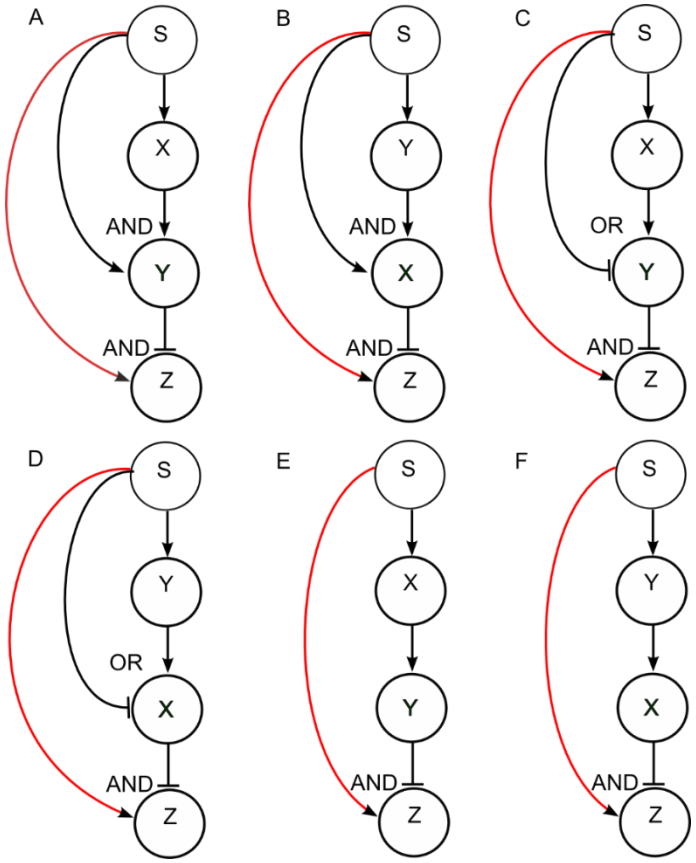

Figure 3.8: An equivalence class of LinearPlus networks under the static epistasis analysis with knockouts.

The link from S-Z (red) can always be identified for this equivalence class, as all the networks have an activating link from S to Z.

4 RESULTS

We have thus far described the theoretical framework we use for our analysis. To demonstrate how helpful the addition of dynamic analysis is to epistasis analysis, we conduct an identifiability analysis of the two gene network structures in each of the network classes (Linear, LinearPlus, AllAcyclic, SKOV) under the different experimental conditions described in the previous chapter (Static, Static with KO, Static with KO and KI, Step Dynamic, Step Dynamic with KO, Step Dynamic with KO and KI, Full Dynamic, Full Dynamic with KO, Full Dynamic with KO and KI). We then conduct an identifiability analysis of the individual links.

4.1 Equivalence Classes and Network Identifiability

We start with the simplest class of networks. Figure 4.1 provides a visual representation of all the network structures in the Linear network class reorganized according to the outputs generated by each experimental condition. With experimental conditions which do not include the addition of genetic perturbations, be it static analysis, step dynamic analysis or full dynamic analysis none of the network structures are uniquely identified. The networks are divided into two equivalence classes, each of size eight. Red rectangles are drawn around the two groups of networks in Figure 4.1. For all the networks, it takes exactly three time steps for any change in the input signal to propagate to the output trait and $Z(t)$ can always be written in terms of $S(t-3)$. There are two equivalence classes as the net effect of the networks is either $Z = S$ or $Z = NOT\ S$. For example, consider the 8 networks in the equivalence class on the left in Figure 4.1. For all the networks, if the

signal *S* is on, the output trait *Z* eventually turns on and if the signal is off, *Z* turns off. However, for the equivalence class on the right, the opposite effect on the output trait is seen. If *S* is on, *Z* turns off and if *S* is off, *Z* turns on.

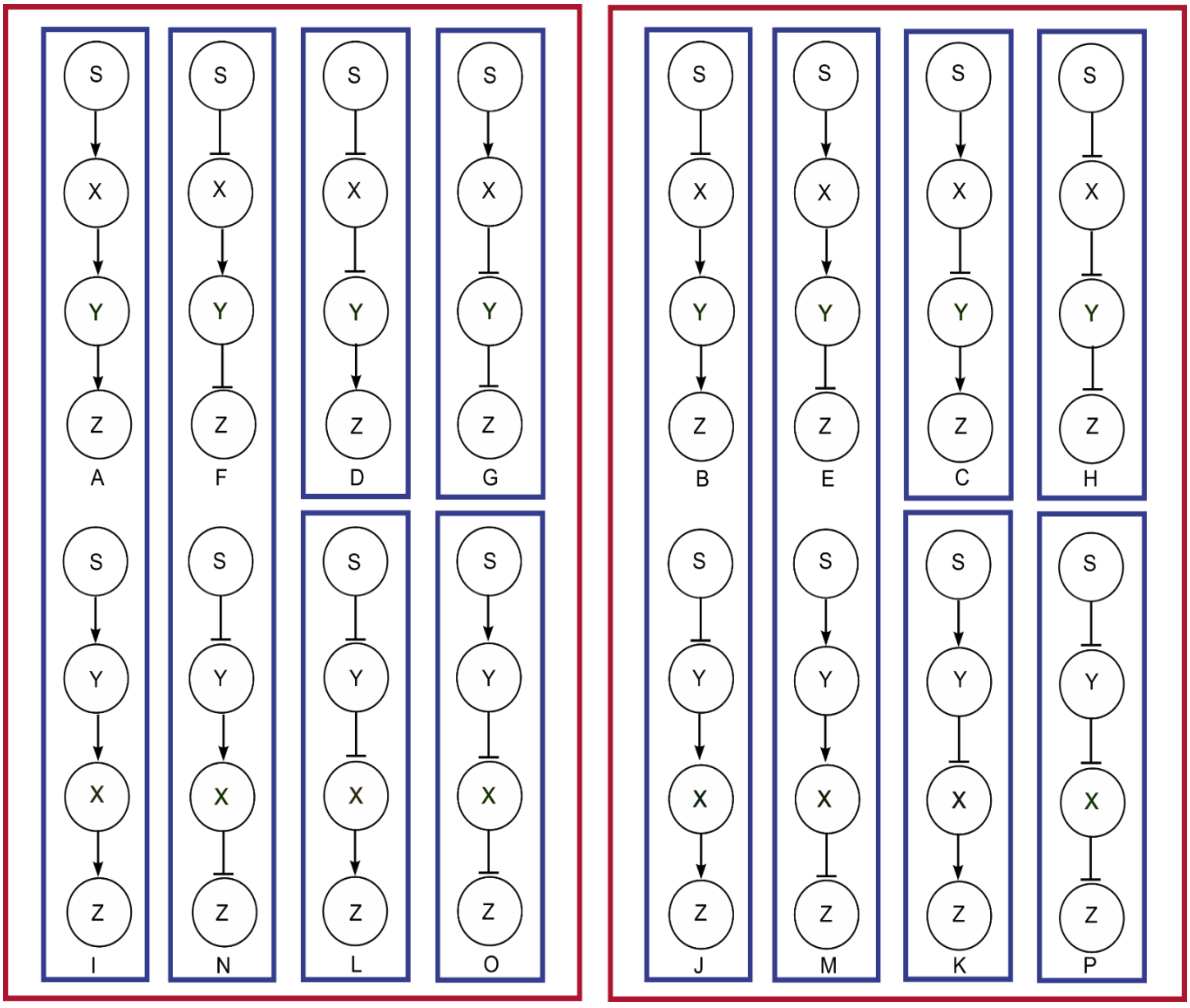

Figure 4.1: Linear networks are categorized into different categories depending on the experimental condition.

Red rectangles are drawn around two equivalence classes generated with experimental conditions which do not include the addition of genetic perturbations, be it static analysis, step dynamic analysis or full dynamic analysis. Blue rectangles surround the equivalence classes generated from the addition of single and double gene deletions to all types of signal dynamics. This results in the unique identification of half of the networks (D, G, C, H, L, O, K, P).

The addition of single and double gene deletions for all types of signal dynamics results in the unique identification of half of the networks (D, G, C, H, L, O, K, P). The rest of the eight networks are divided into four equivalence classes, where each class contains a pair of networks. Blue rectangles surround all the equivalence classes generated. In the equivalence classes of size two, we observe that the genes *X* and *Y* are in the two possible orders, with an activating link between them. Also, for these classes, the signal regulates the upstream gene and the downstream gene regulates the output trait in the same way. For example, in the equivalence class with networks Figure 4.1A and I, the signal activates the upstream gene, the upstream gene activates the downstream gene, and the downstream gene activates the output trait. Dynamics aside, the deletion of either or both genes *X* and *Y* in these networks results in the turning off of *Z*. Finally, with the use of both knockouts and knockins, regardless of the type of input signal, all the networks in the Linear network class are uniquely identified.

While the use of a dynamic signal in the Linear class does not influence the number of identifiable graphs or equivalence classes, as all the network structures can be uniquely identified using only static analysis with knockouts and knockins, the LinearPlus, AllAcyclic and SKOV network classes show different trends. Figure 4.2 provides a visual depiction of the identifiable and non-identifiable but equivalent networks induced by different experimental conditions, while Table 4.1 provides some summary statistics. In Table 4.1, we report the number of similar pairs (4.1A), the number of uniquely identifiable networks (4.1B), the number of non-singleton equivalence classes (4.1C) and the average size of the non-singleton equivalence classes (4.1D), under each experimental

condition for each network class. In our study, a similar pair of networks, for example, are networks that generate the same output, meaning that they look identical under a specified experimental condition. For each network class, this helps to observe the effect of the experimental conditions relative to each other. We notice that within each type of signal dynamics (static, step or full dynamics), as you move progressively to the right with the addition knockouts then knockins and knockouts, the number of similar pairs decreases, the number of identifiable graphs increases, the number of non-singleton equivalence classes increases, and the average size per non-singleton equivalence class decreases.

Unsurprisingly, static analysis without the privilege of gene perturbations provides the least information for discriminating network structures. No network can be identified definitively, and for the three network classes (LinearPlus, AllAcyclic and SKOV), the networks are simply divided into four equivalence classes. These correspond to networks always outputting $Z = 0$, those with $Z = 1$, those with $Z = S$, and those with $Z = \text{NOT } S$. The addition of dynamics in the three classes of networks, LinearPlus, AllAcyclic and SKOV, results in a substantial increase in the number of equivalence classes. While dynamics alone does not uniquely identify any graph structure (Figure 4.2B, C, E, F, H, I outer ring, or Table 4.1), the integration of knockouts or knockins and knockouts with dynamics shows a large advantage over static analysis. For example, in the LinearPlus network class, only 0.588% of the networks can be uniquely identified using static analysis with knockouts. The use of full dynamics with knockouts increases the percentage of identifiable networks to 23.5%. The combination of knockouts and

knockins with dynamic analysis produces an even higher percentage of identifiability. For example, approximately half of the LinearPlus networks are uniquely identified with the use of full dynamics with knockouts and knockins, while 43.5% of the networks can be uniquely identified using step dynamics for *S* with knockouts and knockins.

Table 4.1: Statistics on similar pairs, network identifiability and equivalence classes.

	Static Dynamics			Step Dynamics			Full Dynamics		
	WT	WT + KO	WT + KOKI	WT	WT + KO	WT + KOKI	WT	WT + KO	WT + KOKI
A) Number of Similar Pairs:									
Linear	56	4	0	56	4	0	56	4	0
LinearPlus	242072	51844	3648	46104	6412	592	23704	3204	336
AllAcyclic	1348916	467686	30014	400500	177736	14046	312532	156584	13470
SKOV	7622	2164	396	940	420	68	828	340	68
B) Number of Identifiable Networks:									
Linear	0	8	16	0	8	16	0	8	16
LinearPlus	0	8	80	0	200	592	0	320	688
AllAcyclic	0	8	56	0	184	600	0	304	792
SKOV	0	8	40	0	56	120	0	56	120
C) Number of Non- Singleton Equivalence Classes:									
Linear	2	4	0	2	4	0	2	4	0
LinearPlus	4	76	256	36	268	320	62	278	336
AllAcyclic	4	76	327	36	288	583	64	316	567
SKOV	4	20	44	30	36	44	42	44	44
D) Average Size of Non- Singleton Equivalence Classes:									
Linear	8	2	0	8	2	0	8	2	0
LinearPlus	340	17.8	5	37.8	4.3	2.4	21.9	3.7	2
AllAcyclic	810.8	42.6	9.7	90.1	10.6	4.5	50.7	9.3	4.3
SKOV	55	10.6	4.1	7.3	4.6	2.3	5.2	3.7	2.3

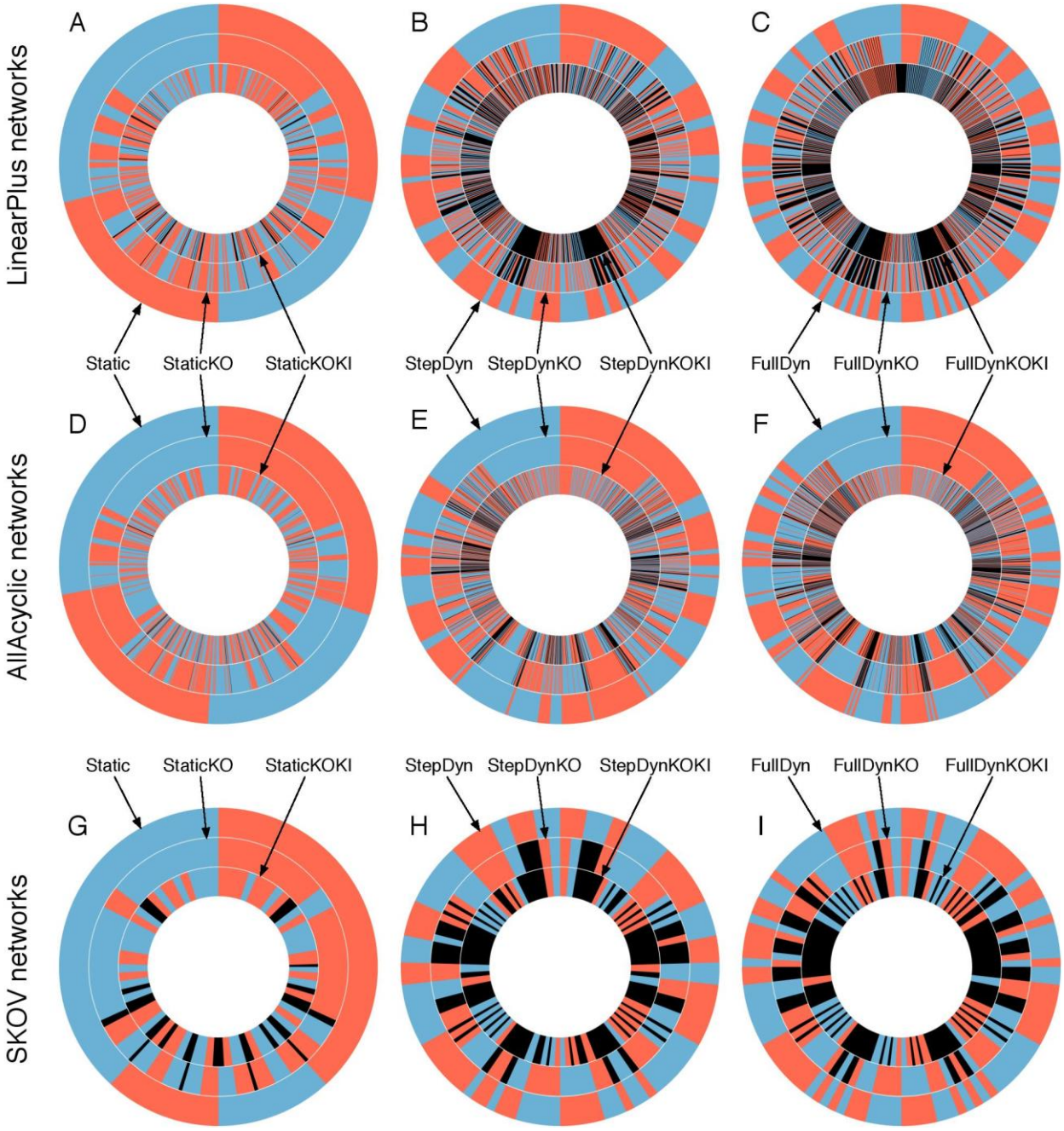

Figure 4.2: Circos graphs visualizing the equivalence classes and identifiable networks for different network classes and under different experimental conditions

A similar trend is observed for the AllAcyclic network class. Full dynamics with

Each graph represents a specific network class (A-C: LinearPlus, D-F: AllAcyclic, G-I: SKOV) and the type of dynamics assumed for the input signal (A, D, G: static, B, E, H: step dynamics, C, F, I: full dynamics). Within each ring of each graph, the alternating red and blue wedges correspond to equivalence classes with more than one network, with the size of the wedge being proportional to the number of networks in the equivalence class. Black wedges indicate networks that fall in their own equivalence class, and are thus identifiable. The three rings within each graph correspond to wild-type observations only (outer ring), wild-type plus single and double deletions (middle ring), and wild-type plus single and double deletions and knockins (inner ring).

knockouts uniquely identifies 9.37% of networks compared to only 0.247% using static analysis. The addition of knockins also increases the percentage of identifiability for both types of analyses. For example, 24.4% of the networks can be distinguished using full dynamic analysis with knockins and knockouts.

Also, for LinearPlus and AllAcyclic class networks, while both full and step dynamics display improved results over static analysis, full dynamics with genetic perturbations can uniquely identify more graph structures than those with step dynamics and genetic perturbations. However, the same cannot be said about the SKOV class network. For SKOV networks, both step and full dynamic analysis identify the same percentage of graph structures. A time varying signal with knockouts and both knockouts and knockins identify 25.5% and 54.5% graph structures respectively. Thus, while the addition of dynamics greatly increases the percentage of identifiable graphs, the use of full dynamics in the SKOV network class has no advantage over step dynamic analysis in the identifiability of graphs. If we consider the number of equivalence classes (Figure 4.2H and I), full dynamics with knockouts results in an increase in the number of equivalence classes in comparison to step dynamics with knockouts. However, knockouts and knockins with step and full dynamics result in the same number of equivalence classes.

As one would expect, experimental conditions that result in greater numbers of identifiable networks also tend to result in greater numbers of smaller equivalence classes of non-identifiable networks (Figure 4.2, Table 4.1). That is, even when a network cannot be uniquely identified, there tend to be fewer other networks with which it can be

confused. For instance, static analysis with knockouts on LinearPlus networks results in 76 equivalence classes with an average size of roughly 18 networks. Full dynamic analysis with knockouts increases the number of equivalence classes to 278 with average size of roughly 4. Similar results are observed for step dynamic analysis with knockouts. Knockouts and knockins with full dynamics on LinearPlus networks produces 336 equivalence classes of size 2, where each pair of networks differs by a single link. Consider Figure 4.3 where an example of two equivalence classes (A and B) generated by full dynamics with knockins and knockouts are shown. In Figure 4.3A, the two networks only differ in the order of the *X* and *Y* genes. Similarly, in 4.3B, we observe that the all links are identical, except for the link between S and *X*.

Thus, while dynamics with knockouts or knockouts and knockins may not uniquely identify all the network structures, it greatly reduces the number of possible networks that are consistent with given experimental outcomes. Further, and boding well for experimental utility, manipulating the input signal through simple step dynamics is nearly as powerful as a more sophisticated dynamical scheme that obtains the maximum possible information. To summarize the results, Figure 4.4 illustrates the percentage of identifiable graphs in each network class under the different experimental conditions in a bar chart.

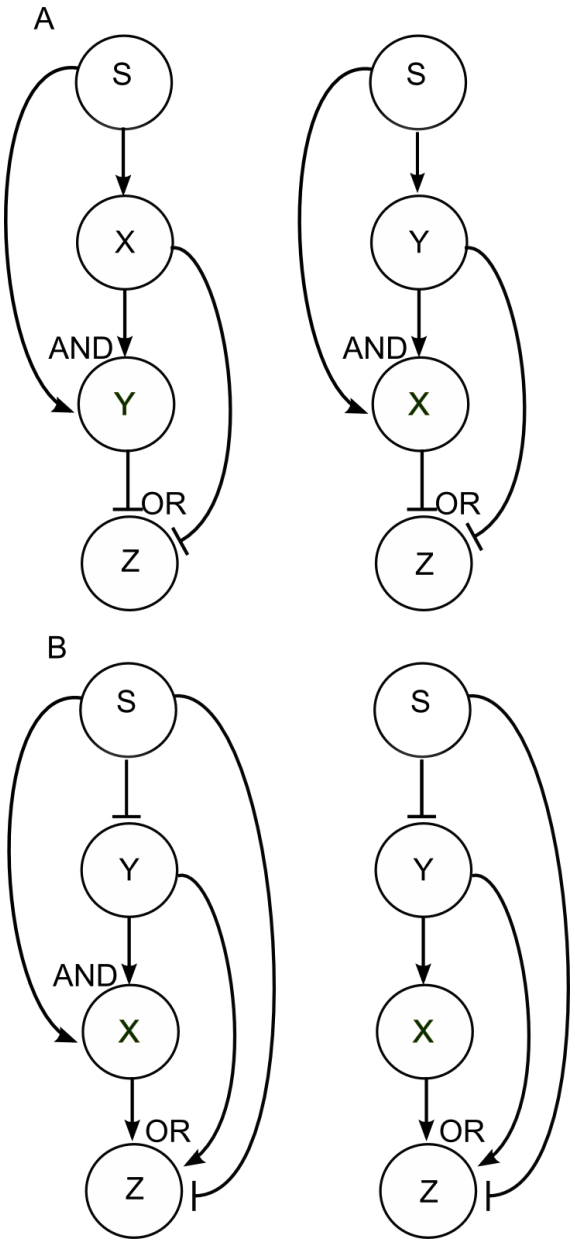

Figure 4.3: An example of two equivalence classes (A and B) of LinearPlus networks generated by full dynamics with knockins and knockouts.

Each equivalence class consists of a pair of LinearPlus networks. These pair of networks only differ by a link. For 4.3A, they only differ in the *X*-*Y* link, as for 4.3B, they only differ in the *S*-*X* link.

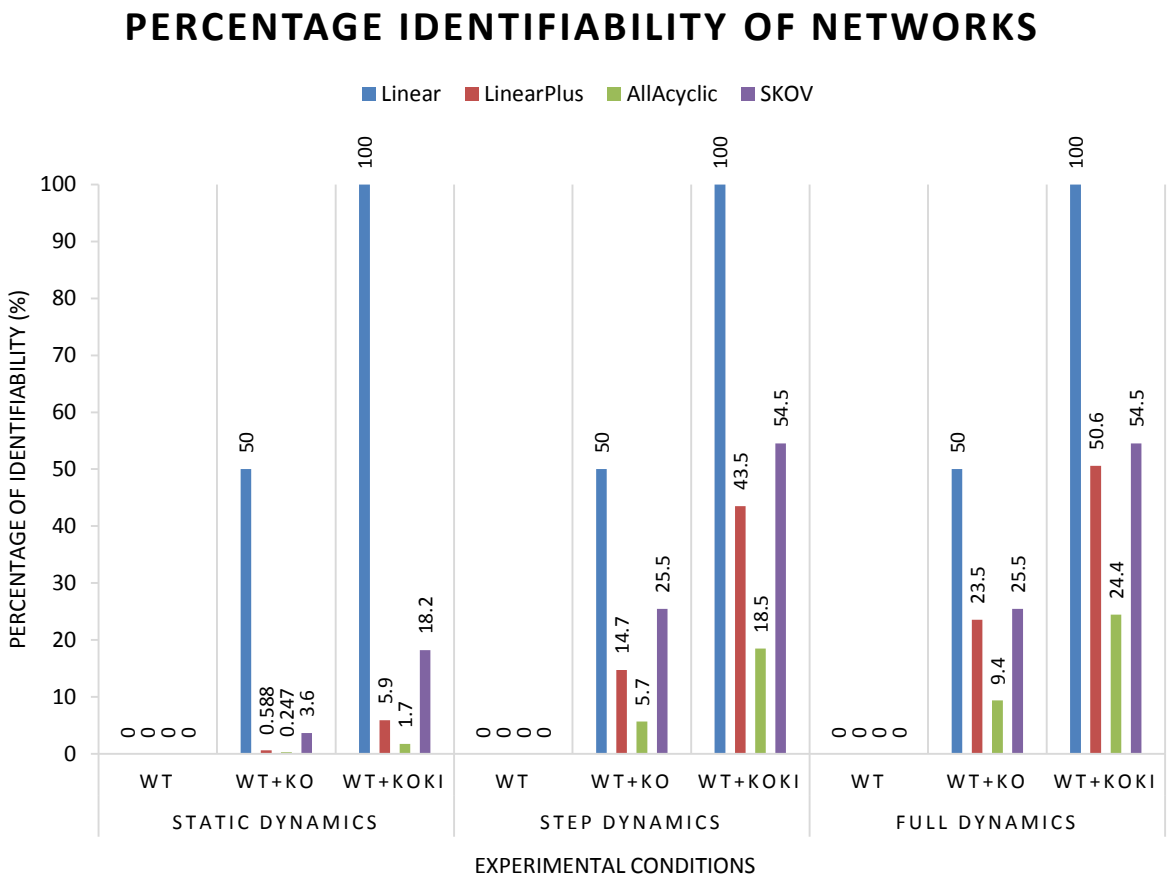

Figure 4.4: Percentage of identifiable networks in each network class: Linear, LinearPlus, AllAcyclic and SKOV under the different experimental conditions: Static, StaticKO, StaticKOKI, StepDynamic, StepDynamicKO, StepDynamicKOKI, FullDynamic, FullDynamicKO and FullDynamicKOKI.

4.2 Individual Link Identification

Here, we consider the percentage of identifiability of all the links (see Figure 4.5 for Linear, LinearPlus, AllAcyclic and SKOV link identifiability under static signal with knockouts and under full dynamics with knockouts; Table 4.2 provides comprehensive statistics).

Again, we start with the simplest class. For an analysis with knockouts regardless of the input signal *S* in the Linear network class (Figure 4.5A and B), the links between *S* and the genes *X* and *Y* and those between *X* and *Y* and the output trait *Z*, can be positively identified for half of the networks. Similar results are shown by specifically considering the activating, repressing or absent links. The addition of knockins regardless of the input signal increases the percentage identifiability to 100% for all the links (Table 4.2A).

For static analysis with knockouts of the LinearPlus network class (Figure 4.5C and D), the $S - X$ and $S - Y$ links can be confidently identified in just 6.2% of the networks. In more detail, when such links are activating or repressing, we can be sure of this for 6.9% of networks. However, when there is no link from $S - X$ or from $S – Y$, we can never be sure of this fact, under the condition with static signal and gene knockouts. The $S - Z$ link has a greater percentage of identifiability at 31.8% across all types of interactions, with 33.3% for both the activation and repression links and 28% for links with no interaction. The identifiability of links between the genes (X and Y) and the trait Z is similar to that between the signal and the genes. Across all types of interactions, 31.2% of the links can be identified, with 34.8% for both activation and repression links and 10%

when there is no interaction. In the comparison of static analysis with knockouts to full dynamics with knockouts (Figure 4.5D), we observe a great increase in the percentage of identifiability of all the links. Perhaps most dramatically, the links between the signal and genes (*X* and *Y*) go from 6.2% identifiable to 57.6% identifiable when we switch from

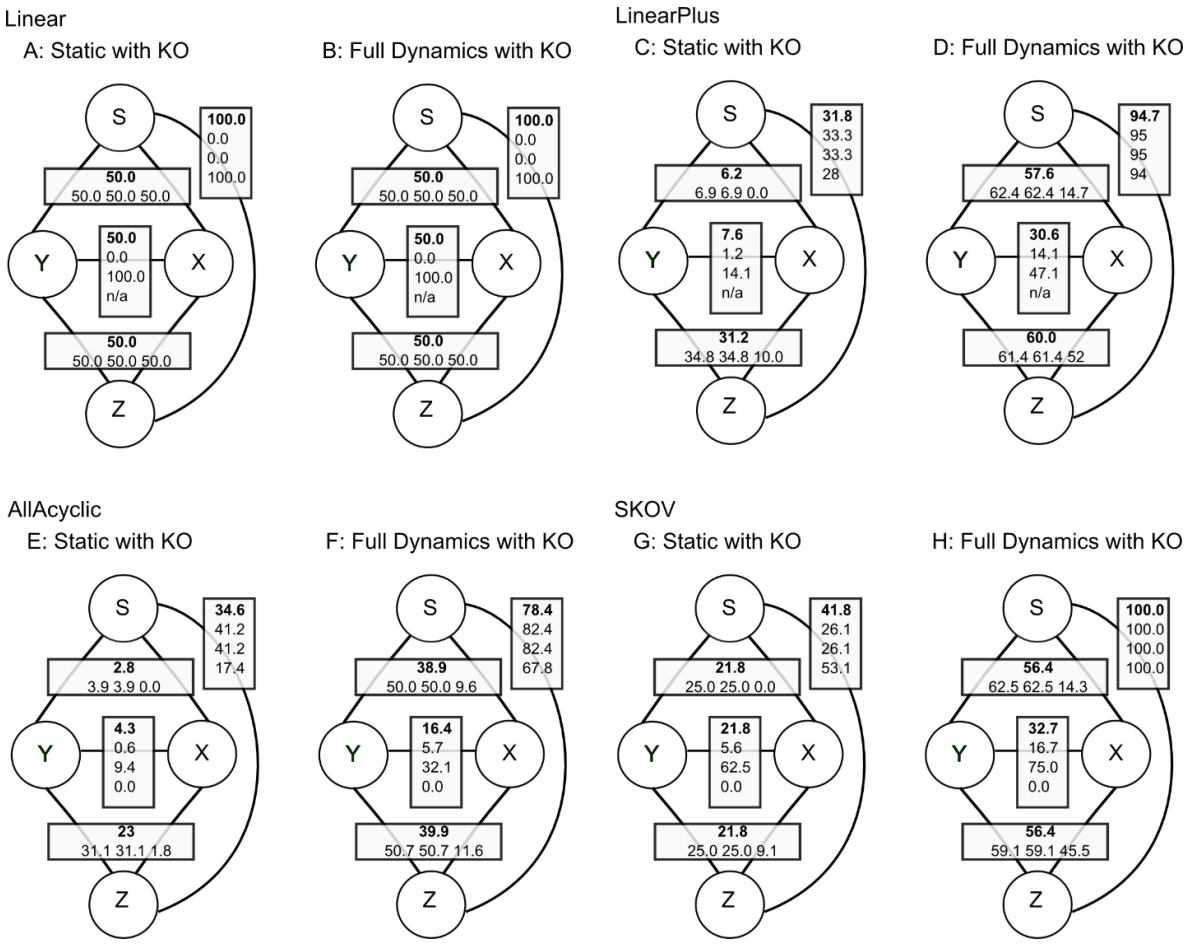

Figure 4.5: Percentages of networks in all network classes (A-B: Linear, C-D: LinearPlus, E-F: AllAcyclic, G-H: SKOV) for which different links can be identified under the experimental condition combining static signal with knockouts (A, C, E, G) and the condition combining a fully dynamic signal with knockouts (B, D, F, H).

Due to the symmetry of *X* and *Y*, identifiability of $S - X$ and $S - Y$ links is identical, and similarly for $X - Z$ and $Y - Z$ links. Within each box, the bold number gives the total percentage of networks for which the link can be definitely identified as being activating, repressing or absent. The remaining three numbers give the total percentage of networks where the link can be identified, in comparison with the total number of networks that have an activating, repressing or absent link respectively.

static to dynamic analysis. There is near perfect identifiability for the $S - Z$ link, and substantially better identifiability of the relationship between X and Y. Similar trends are seen for the AllAcyclic (Figure 4.5E and F) and the SKOV network classes (Figure 4.5G and H).

Dynamics alone for the LinearPlus and AllAcyclic network structures does not assist in identifying any of the links (they are 0% identifiable) except for the $S - Z$ link. In comparison with these two network classes, however, when dynamics alone is used in the SKOV class, the links $S - X$ or Y, X or $Y - Z$ and $S - Z$ always show some percentage of identifiability, with the $S - Z$ link being the most identifiable. Also, while a significant increase is observed with the use of either types of time varying input signal compared to static in the SKOV network class, the links in both step and full dynamics analysis have identical percentages of identifiability. Thus, there is no advantage of using the full dynamics analysis over the step dynamic analysis.

In all network classes, for all links excluding the interaction between genes X and Y, in all the experimental conditions that do not include knockins, and for all network classes, activating and repressing links have equal identifiability. A repressing link between X and Y, however, has greater identifiability than an activating one. For example, in the case of static analysis with knockouts for LinearPlus networks, activating links are 1.2% identifiable whereas repressing links are 14.1% identifiable. When knockouts and knockins are allowed, this disparity vanishes, so that activating and repressing links are equally identifiable.

Naturally, the use of full dynamics with both knockouts and knockins generates the best results in the unique identification (with the highest percentages) of each link in the network structure for LinearPlus, AllAcyclic and SKOV networks (Table 4.2). All the links to the output trait are 100% identifiable and the links from the input signal to the genes are 85.9%, 62.3% and 89.1% for LinearPlus, AllAcyclic and SKOV respectively. Additionally, the interactions between the genes X and Y have the highest percentage of identifiability compared to other types of analysis. For example, in the LinearPlus network class, 78.8% of the $X - Y$ links can be identified.

Table 4.2: Percentage of networks for which links can be identified.

	Static Dynamics			Step Dynamics			Full Dynamics		
	WT	WT + KO	WT + KOKI	WT	WT + KO	WT + KOKI	WT	WT + KO	WT + KOKI
A) Linear									
S — X or Y ,overall	0	50	100	0	50	100	0	50	100
activating	0	50	100	0	50	100	0	50	100
repressing	0	50	100	0	50	100	0	50	100
absent	0	50	100	0	50	100	0	50	100
S — Z , overall	100	100	100	100	100	100	100	100	100
activating	0	0	0	0	0	0	0	0	0
repressing	0	0	0	0	0	0	0	0	0
absent	100	100	100	100	100	100	100	100	100
X — Y , overall	0	50	100	0	50	100	0	50	100
activating	0	0	100	0	0	100	0	0	100
repressing	0	100	100	0	100	100	0	100	100
absent	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
X — Z or Y ,overall	0	50	100	0	50	100	0	50	100
activating	0	50	100	0	50	100	0	50	100
repressing	0	50	100	0	50	100	0	50	100
absent	0	50	100	0	50	100	0	50	100
B) LinearPlus									
S — X or Y ,overall	0	6.2	34.1	0	50	83.5	0	57.6	85.9
activating	0	6.9	37.3	0	54.9	89.5	0	62.4	92.2
repressing	0	6.9	37.3	0	54.9	89.5	0	62.4	92.2
absent	0	0	5.9	0	5.9	29.4	0	14.7	29.4
S — Z , overall	0	31.8	100	57.6	84.1	100	85.9	94.7	100
activating	0	33.3	100	60	85	100	86.7	95	100
repressing	0	33.3	100	60	85	100	86.7	95	100
Absent	0	28	100	52	82	100	84	94	100
X — Y , overall	0	7.6	34.1	0	27.1	76.5	0	30.6	78.8
activating	0	1.2	34.1	0	9.4	76.5	0	14.1	78.8
repressing	0	14.1	34.1	0	44.7	76.5	0	47.1	78.8
Absent	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
X — Z or Y ,overall	0	31.2	100	0	53.5	100	0	60	100
activating	0	34.8	100	0	55.2	100	0	61.4	100
repressing	0	34.8	100	0	55.2	100	0	61.4	100
absent	0	10	100	0	44	100	0	52	100

Table 4.2: Percentage of networks for which links can be identified (Continued).

	Static Dynamics			Step Dynamics			Full Dynamics		
	WT	WT + KO	WT + KOKI	WT	WT + KO	WT + KOKI	WT	WT + KO	WT + KOKI
C) AllAcyclic									
S — X or Y ,overall	0	2.8	21.4	0	34.5	59.9	0	38.9	62.3
activating	0	3.9	26.7	0	46.5	68.6	0	50	70.6
repressing	0	3.9	26.7	0	46.5	68.6	0	50	70.6
absent	0	0	7.4	0	2.9	37	0	9.6	40.5
S — Z , overall	0	34.6	100	59.1	73.9	100	71	78.4	100
activating	0	41.2	100	65	78.3	100	76	82.4	100
repressing	0	41.2	100	65	78.3	100	76	82.4	100
absent	0	17.4	100	43.7	62.4	100	58	67.8	100
X — Y , overall	0	4.3	17	0	15.4	37.7	0	16.4	40.7
activating	0	0.6	19.6	0	4.5	43.4	0	5.7	46.8
repressing	0	9.4	19.6	0	30.9	43.4	0	32.1	46.8
absent	0	0	0	0	0	0	0	0	0
X — Z or Y ,overall	0	23	100	0	37.7	100	0	39.9	100
activating	0	31.1	100	0	48.7	100	0	50.7	100
repressing	0	31.1	100	0	48.7	100	0	50.7	100
absent	0	1.8	100	0	8.9	100	0	11.6	100
D) SKOV									
S — X or Y ,overall	0	21.8	58.2	9.1	52.7	89.1	12.7	56.4	89.1
activating	0	25	62.5	10.4	58.3	93.8	14.6	62.5	93.8
repressing	0	25	62.5	10.4	58.3	93.8	14.6	62.5	93.8
absent	0	0	28.6	0	14.3	57.1	0	14.3	57.1
S — Z , overall	0	41.8	100	89.1	100	100	100	100	100
activating	0	26.1	100	91.3	100	100	100	100	100
repressing	0	26.1	100	91.3	100	100	100	100	100
absent	0	53.1	100	87.5	100	100	100	100	100
X — Y , overall	0	21.8	61.8	0	32.7	76.4	0	32.7	76.4
activating	0	5.6	66.7	0	16.7	83.3	0	16.7	83.3
repressing	0	62.5	62.5	0	75	75	0	75	75
absent	0	0	0	0	0	0	0	0	0
X — Z or Y ,overall	0	21.8	100	14.5	56.4	100	14.5	56.4	100
activating	0	25	100	18.2	59.1	100	18.2	59.1	100
repressing	0	25	100	18.2	59.1	100	18.2	59.1	100
absent	0	9.1	100	0	45.5	100	0	45.5	100

5 CONCLUSIONS AND FUTURE WORK

In our theoretical investigation, we explored the potential of dynamics in epistasis analysis. We simulated using a dynamic input signal to drive Boolean models of genetic pathways, in addition to genetic perturbations, to obtain a better understanding of the organization of genes in the pathway and uniquely identify a greater number of gene networks. To quantify how helpful the addition of dynamics to epistasis analysis is, we conduct an identifiability analysis of network structures and individual links under different experimental conditions. We show the advantage of dynamic epitasis analysis using the three different network classes (LinearPlus, AllAcyclic, and SKOV) and different experimental conditions (with or without genetic perturbations).

Our primary findings suggest that the use of a dynamic signal alone does not uniquely identify any of the network structures. It even appeared to be weaker in comparison with traditional genetic approaches based on genetic perturbations. However, the combination of dynamical input with gene perturbations proved to be far more powerful than the classical static epistasis analysis approach. Dynamic epistasis could be used to discriminate between GRNs that were previously indistinguishable and identify a higher percentage of links. For instance, when we enumerated all acyclic Boolean pathway models, we found that only 0.247% of them could be uniquely identified by a classical, static epistasis analysis based on gene knockouts. However, driving the same pathways with a dynamical input, in combination with gene knockouts, allows 9.37% of them to be

identified uniquely. Our positive results show the potential value of dynamics in epistasis analysis.

While a better understanding of the pathway architectures is obtained with dynamics and genetic perturbations, there are still some architectures that could not be entirely identified. It is not possible to perfectly discriminate between all the GRN structures in the three network classes: LinearPlus, AllAcyclic and SKOV, using either static or dynamic epistasis analysis. However, dynamics with genetic perturbations greatly reduces the number of alternatives, and often results in the unique identification of certain links within the pathway. Also, we observed that the addition of dynamics to gene deletions has greater discriminatory power than the addition of knockouts and knockins to static epistasis analysis. This is true for overall network identifiability as well as for links interior to the pathway, although direct links from pathway members to the output phenotype/trait are best identified by combining knockouts with knockins. Of course, all three can be combined for even better pathway inference.

An important question for further research is what more can we add to attain full pathway identifiability? One possibility is dynamic gene perturbations, where genes are perturbed during the course of an experiment. Here, we have explored the value of dynamics only in the input signal driving a pathway.

Our study is not without limitations. Many simplifying assumptions that may not always be true in practice are made. We use Boolean models, with dynamics that are

deterministic, proceed in discrete time, and are not subject to time delays, to represent GRNs. While Boolean models are useful in simplifying complex issues and thus have allowed us to observe the overall advantage of a dynamical approach, real biological systems are different from Boolean networks. Nodes in a Boolean network take only binary values which are updated synchronously, whereas experiments involving external stimuli, gene expression and an observable output trait/phenotype in real genetic pathways are not binary and change continuously [3]. For example, our analysis assumes that deleted mutant genes completely lack normal gene functionality, meaning a deleted gene has no activity. However, mutants generated in genetic screens may exhibit partial loss or partial gain of function in case of knockdowns or knockins respectively, which may generate more complex phenotypes. In the case of the observable output trait, most recent studies on epistasis analysis have been exploring the use of quantitative phenotypic measurements [20-23]. Moreover, with Boolean models, deterministic GRNs are assumed. However, according to recent studies, most biological systems are stochastic in nature [4, 8]. For future work, more general model systems, such as dynamic Bayesian models, which can incorporate prior knowledge and handle hidden variables and missing data, could be considered to represent GRNs. In other words, a more realistic model which captures properties of GRNs not present in a simple Boolean model could be selected.

Other assumptions made in our study include the number of genes in the genetic pathway, the number of inputs per gene/output trait and the logical functions used for the genes/output trait with multiple inputs. With two intermediate genes and the use of only

logical AND or logical OR, only specific subnetworks of biological systems are considered. Additionally, we assume that the genetic pathway is driven by one external stimuli, which may not always be the case. Future work may include conducting a theoretical investigation of epistasis analysis with the addition of dynamics by enumerating all the possible networks of an increased size, with more logical functions and external stimuli possibilities. We also restrict the type of GRNs to only acyclic networks. Although most studies on epistasis analysis assume the acyclicity of gene networks [11-28], many biological processes rely on feedback mechanisms for regulation [26]. They may appear as loops in which one gene regulates itself or two or more genes regulate each other. Future work may also include modeling gene networks as those with feedback loops and multidirectional influence between the genes.

As seen above, relaxing some of these assumptions is an important direction for future work. It would allow the continuation to assess the theoretical value of dynamics in epistasis analysis and the development of algorithms that will be directly applicable to real-world data. The combination of quantitative statistical, regression or probabilistic-based approaches [21, 28, 29, 30] with dynamical models in epistasis analysis could lead to methods with great practical utility in determining genetic pathway structures.

The next step would be to translate the theoretical advantage of dynamic epistasis analysis into practice. The applied side of this research project is currently being conducted by our collaborating lab. It involves experimental flow cytometry data analysis, where to identify the DNA damage genetic pathway in yeast, a DNA damaging

drug is used as the external stimuli. Initially, single gene knockouts of all the yeast genes are performed to identify candidate genes that may be involved in the DNA damage pathway. Pairs of candidate genes are then knocked out to further identify the networks of genes influencing the DNA damage response.

Will the assumptions made act as a practical barrier to the understanding of gene networks? It is practically unreasonable to relax all the assumptions. Naturally, scientists choose assumptions depending on the data set and the features to be extracted from the network under study. With the assumptions made in this project, one could get a general understanding of the structure of the network. To determine which model (and assumptions) best represent GRNs, we could model a known GRN using various model based approaches [21, 28, 29, 30] and then conduct an assessment which compares the predicted networks generated by the models from real quantitative phenotypic measurements to the actual known GRN.

REFERENCES

1. Charlotte K. Omoto and Paul F. Lurquin. (2004). Genes and DNA: A Beginner's Guide to Genetics and Its Applications. New York, NY: Columbia University Press.
2. Peter Raven, George Johnson, Susan Singer, Jonathon Losos, and Kenneth Mason. (2008). Biology – 8th Edition. New York, NY: McGraw-Hill Publishing Company.
3. Shoudan Liang, Stefanie Fuhrman, and Roland Somogyi. REVEAL, a general reverse-engineering algorithm for inference of genetic network architectures. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 18–29, 1998.
4. Kevin Murphy and Saira Mian. Modelling gene expression data using dynamic bayesian networks. *Vol. 104. Technical report*, Computer Science Division, University of California, Berkeley, CA. 1999.
5. M. K. Stephen Yeung, Jesper Tegner, and James J. Collins. Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences of the USA*, 99:6163–6168, 2002.
6. Timothy S. Gardner, Diego di Bernardo, David Lorenz, and James J. Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301:102–105, 2003.
7. Adam A. Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo D Favera, and Andrea Califano. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7(Suppl 1):S7, 2006.
8. Andrew Golightly and Darren J. Wilkinson. Bayesian sequential inference for stochastic kinetic biochemical network models. *Journal of Computational Biology*, 13(3):838–851, 2006.
9. Daniel Marbach, Robert J. Prill, Thomas Schaffter, Claudio Mattiussi, Dario Floreano, and Gustavo Stolovitzky. Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences*, 107(14):6286, 2010.
10. Eric Davidson and Michael Levin. Gene regulatory networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(14):4935–4935, 2005.
11. Leon Avery and Steven Wasserman. Ordering gene function: the interpretation of epistasis in regulatory hierarchies. *Trends in Genetics*, 8(9):312–316, 1992.
12. Heather J. Cordell. Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. *Human Molecular Genetics*, 11(20):2463–2468, 2002.
13. Patrick C. Phillips. Epistasis: the essential role of gene interactions in the structure and evolution of genetic systems. *Nature Reviews Genetics*, 9(11):855–867, 2008.
14. Amy Hin Yan Tong, Marie Evangelista, Ainslie B. Parsons, Hong Xu, Gary D. Bader, Nicholas Pag´e, Mark Robinson, Sasan Raghibizadeh, Christopher WV. Hogue, Howard Bussey, et al. Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science*, 294(5550):2364–2368, 2001.
15. Amy Hin Yan Tong, Guillaume Lesage, Gary D. Bader, Huiming Ding, Hong Xu, Xiaofeng Xin, James Young, Gabriel F Berriz, Renee L Brost, Michael Chang, et al. Global mapping of the yeast genetic interaction network. *Science*, 303(5659):808–813, 2004.
16. Selina S. Dwight, Midori A. Harris, Kara Dolinski, Catherine A. Ball, Gail Binkley, Karen R. Christie, Dianna G. Fisk, et al. *Saccharomyces* Genome Database (SGD), accessed 19 August 2003. Available at www.yeastgenome.org.
17. Daniel Segre, Alexander DeLuna, George M. Church, and Roy Kishony. Modular epistasis in yeast metabolism. *Nature Genetics*, 37(1):77–83, 2005.
18. Maya Schuldiner, Sean R. Collins, Natalie J. Thompson, Vladimir Denic, Arunashree Bhamidipati, Thanuja Punna, Jan Ihmels, Brenda Andrews, Charles Boone, Jack F. Greenblatt, et

al. Exploration of the function and organization of the yeast early secretory pathway through an epistatic miniarray profile. *Cell*, 123(3):507–519, 2005.

19. Blaz Zupan, Janez Demsar, Ivan Bratko, Peter Juvan, John A. Halter, Adam Kuspa, and Gad Shaulsky. GenePath: a system for automated construction of genetic networks from mutant data. *Bioinformatics*, *19*(3), 383-389, 2003.
20. Robert P. St Onge, Ramamurthy Mani, Julia Oh, Michael Proctor, Eula Fung, Ronald W. Davis, Corey Nislow, Frederick P. Roth, and Guri Giaever. Systematic pathway analysis using high-resolution fitness profiling of combinatorial gene deletions. *Nature Genetics*, 39(2):199–206, 2007.
21. Hilary Phenix, Katy Morin, Cory Batenchuk, Jacob Parker, Vida Abedi, Liu Yang, Lioudmila Tepliakova, Theodore J. Perkins, and Mads Kaern. Quantitative epistasis analysis and pathway inference from genetic interaction data. *PLoS Computational Biology*, 7(5):e1002048, 2011.
22. Nancy Van Driessche, Janez Demsar, Ezgi O. Booth, Paul Hill, Peter Juvan, Blaz Zupan, Adam Kuspa, and Gad Shaulsky. Epistasis analysis with global transcriptional phenotypes. *Nature Genetics*, 37(5):471–477, 2005.
23. Martin C. Jonikas, Sean R. Collins, Vladimir Denic, Eugene Oh, Erin M. Quan, Volker Schmid, Jimena Weibezahn, Blanche Schwappach, Peter Walter, Jonathan S. Weissman, et al. Comprehensive characterization of genes required for protein folding in the endoplasmic reticulum. *Science*, 323(5922):1693–1697, 2009.
24. Brandon Barker, Lin Xu, and Zhenglong Gu. Dynamic Epistasis under Varying Environmental Perturbations. *PloS One*, 10(1), e0114911, 2015.
25. Michael Costanzo, Anastasia Baryshnikova, Jeremy Bellay, Yungil Kim, Eric D. Spear, Carolyn S. Sevier, Huiming Ding, et al. The genetic landscape of a cell. *Science*, 327(5964), 425-431, 2010.
26. Eugenio Azpeitia, Mariana Ben´ıtez, Pablo Padilla-Longoria, Carlos Espinosa-Soto, and Elena R. Alvarez-Buylla. Dynamic network-based epistasis analysis: boolean examples. *Frontiers in Plant Science*, 2, 2011.
27. Lin Xu, Brandon Barker, and Zhenglong Gu. Dynamic epistasis for different alleles of the same gene. *Proceedings of the National Academy of Sciences*, 109(26):10420–10425, 2012.
28. Hilary Phenix, Theodore J. Perkins and Mads Kaern. Identifiability and inference of pathway motifs by epistasis analysis. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23(2):025103–025103, 2013.
29. Alexis Battle, Martin C. Jonikas, Peter Walter, Jonathan S. Weissman, and Daphne Koller. Automated identification of pathways from quantitative genetic interaction data. *Molecular Systems Biology*, 6(1), 2010.
30. Blaz Zupan and Marinka Zitnik. Gene network inference by probabilistic scoring of relationships from a factorized model of interactions. *Bioinformatics, 30*(12), i246-i254, 2014.
31. Stuart A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, *22*(3), 437-467, 1969.
32. Tatsuya Akutsu, Satoru Kuhara, Osamu Maruyama, and Satoru Miyano. Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. In *Proceedings of the Ninth ACM-SIAM Symposium on Discrete Algorithms*, pages 695–702, 1998.
33. Zoltan Szallasi and Shoudan Liang. Modeling the normal and neoplastic cell cycle with "realistic Boolean genetic networks": their application for understanding carcinogenesis and assessing therapeutic strategies. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 66–76, 1998.
34. Tatsuya Akutsu, Satoru Miyano, and Satoru Kuhara. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 17–28, 1999.

35. Stuart A. Kauffman, Carsten Peterson, Björn Samuelsson, and Carl Troein. Random Boolean network models and the yeast transcriptional network. *Proceedings of the National Academy of Sciences of the USA*, 100(25):14796–14799, 2003.
36. Joshua ES. Socolar and Stuart A. Kauffman. Scaling in ordered and critical random boolean networks. *Physical Review Letters*, 90(6), 2003.