

Evaluating Machine Learning Algorithms Used to Infer Gene Regulatory Network Structure

by

Dan Handley

Submitted to the Department of Philosophy in partial
fulfillment of the requirements for the degree of

Master of Science

in

Logic and Computation

Carnegie Mellon University
November 2002

Acknowledgments

I would like to express my sincere gratitude to Clark Glymour for his support and guidance. I am greatly indebted to Peter Spirtes, Richard Scheines, Larry Wasserman, Joe Ramsey, and David Peters for their kindness, patience, and invaluable instruction. I would also like to thank Trevor Tompkins for his suggestions and assistance.

I. INTRODUCTION	1
II. WHY IS IT IMPORTANT TO UNDERSTAND GENE REGULATION?	3
III. BRIEF OVERVIEW OF GENE EXPRESSION AND REGULATION	7
How genes are expressed	7
Gene regulation in prokaryotes and eukaryotes	12
Prokaryotes	13
Eukaryotes	18
Eukaryotic RNA	20
Non-regulatory factors in determining gene expression activity	21
Eukaryotic gene regulatory mechanisms	23
IV. MODEL REPRESENTATION OF GENE REGULATION	27
Causal models and DAGs	28
Terminology	31
Variables	32
Inferring independence relationships from data	34
Multivariate systems	39
Inferring causal relationships from data	41
Causal Markov Condition	41
Faithfulness	43
Causal Sufficiency	45
Conditioning on colliders	47
Automated causal inference	48
Transmission functions	53
V. MEASUREMENT OF GENE EXPRESSION	55
What makes understanding gene regulation difficult?	55
Brief overview of microarray technology	58
cDNA arrays	59
Affymetrix arrays	64

VI. EXPERIMENTAL MANIPULATION OF GENE EXPRESSION	67
Gene expression manipulation	71
Knockout and knockin experiments	73
Gene underexpression and overexpression	76
VII. SEARCH REPRESENTATIONS AND STRATEGIES	78
Experimental data	78
Causation versus association in passive observation	81
Scoring searches	84
Constraint-based searches	86
Hybrid approaches	87
Clustering/PCA	87
VIII. EVALUATING A SEARCH METHOD BASED ON ACTUAL DATA	89
Model discovery versus model validation	89
Other considerations	90
“Gold Standards”	92
IX. CONCLUSION	101
REFERENCES	104
APPENDIX A – THE PC ALGORITHM	107

I. Introduction

DNA—in the form of genes—carries the information necessary for making proteins in all known life. Which proteins are created, and in what quantity and at what time, determines the structure, function, and behavior of all cells. Which genes are expressed and when they are expressed is in turn regulated by other genes. Understanding the function and interaction of these genes promises to improve disease diagnosis, disease treatment, and drug discovery as well as contributing tremendously to our fundamental understanding of biological processes.

The human genome contains between 30,000 and 35,000 genes encoded by 3.1 billion total nucleotide base pairs (A,C,T,G) [29]. Performing individual molecular biology experiments to uncover how each of these genes interacts with every other gene separately is expensive and prohibitively time-consuming. A recent technological development promises to help enormously, however. The *DNA microarray* can be used to measure expression levels of thousands of genes simultaneously and relatively cheaply. Interpreting this enormous amount of information is, however, a daunting task. Some computational biologists wish to employ artificial intelligence techniques, or more precisely, machine learning, to infer the causal relationships between the thousands of genes in a complex regulatory network. Using these machine learning techniques, we hope to uncover the complex relationships between how the expression of each gene affects the expression of every other gene in both normal and disease states—in a fraction of the time and at a fraction of the cost of ordinary molecular biology experiments.

A number of different machine learning strategies have already been proposed for inferring regulatory network structure from microarray data. Some examples of these are

multivariate linear regression [17], neural network pruning [17], Boolean networks search with mutual information measures [25], linear programming [2], simple correlation analysis [5], Bayes net search [20], and Constraint/Scoring Bayes net search [13,40]. How do we know which if any of these approaches are viable and worthy of further pursuit?

In this paper I present two approaches for evaluating machine learning algorithms used in inferring gene regulatory networks. The first is based on looking at the design criteria used in a machine learning model to infer the causal network involved in gene regulation. The second is based on evaluating the performance of a gene regulatory network model derived through machine learning with respect to some standard derived from actual biological data. This requires a “gold standard” of sorts, that is, data and a known causal structure obtained independently through biological experimentation.

II. Why is it important to understand gene regulation?

At the time of this writing, 2595 human genes have so far been identified as either causing, predisposing, or protecting people from disease [43]. Examples of such genetically-linked afflictions are Crohn's disease, cardiomyopathy, Amyotrophic Lateral Sclerosis (Lou Gehrig's Disease), many if not all cancers, and both juvenile and adult-onset diabetes. Understanding how these genes are expressed and regulated therefore promises many benefits to humankind.

When considering medical science, we might put all somatic pathology into three broad categories: afflictions arising from trauma, those arising from exogenous pathogens, or those that are somehow the result of how one's own genes are expressed.¹ Modern medicine has made enormous strides over the past century in treating trauma and infectious diseases, but the treatments developed for genetically-linked afflictions such as diabetes, cardiovascular disease, and cancer are still relatively crude—and are often merely symptomatic or palliative treatments at that. Understanding and ultimately deliberately manipulating gene expression at the cellular level will be a true revolution in the practice of medicine.

The first such benefit, one society is just beginning to reap, is to improve diagnostics and identify genetic predispositions to disease. Once those at higher risk for diseases are identified, closer medical surveillance of this group can improve the odds of early detection and thus reduce both morbidity and mortality. Already gene-based tests

¹We can also propose a fourth category, *toxicology*, which one could argue might be subsumed under the first and third categories mentioned.

are available for dozens of diseases including Huntington's disease, thalassemias, and cystic fibrosis [29].

A better understanding of gene regulation also holds promise for drug discovery and delivery, a rapidly-expanding area of research known as *pharmacogenomics*. There is significant variation in genetically-determined enzyme expression patterns among human populations. These enzymes greatly affect the rates of absorption, distribution, metabolism, and excretion of drugs and their metabolites. Drug effects therefore may vary enormously between any two individuals in a population. A drug dosage that is therapeutic to one individual might be dangerously toxic to another. Conventional drugs also often have wide-ranging and unintended metabolic effects, resulting in numerous side-effects. By tailoring drugs, dosages, and delivery regimens to an individual's particular genetic makeup, drugs can be used to target only the specific gene and metabolic pathways responsible for the disease. This will result in drugs with much greater specificity, efficacy, and with far fewer side-effects than those of conventional treatments.

Understanding gene regulatory networks and gene expression will also allow us to better understand the cell's response to toxic substances. Understanding how gene expression patterns responds to toxicants will produce much better methods for toxicological assessment of chemical agents, and therefore promises to produce much more sensitive and specific toxicological tests than those currently available. Further, understanding how gene expression responds to DNA damage will allow us to better understand the mechanisms involved in cellular response to various genotoxic agents, such as DNA-damaging mutagens. Uncovering the underlying mechanisms of the cell's

response to physical or chemical injury will ultimately allow us to design more effective and specific drug therapies and chemoprotective agents.

A better understanding of gene regulation may also allow us to control stem cell differentiation and tissue development [30]. This may someday allow generation of replacement tissues and even whole organs to be effectively “engineered” out of an individual’s own genetic material. For instance, a person whose heart cannot pump blood adequately due to chronic disease or injury may have that heart repaired with new muscle cells grown from other cells containing his own genetic material. To the person’s immune system, these new cells would therefore be not be considered foreign. Producing replacement tissues and organs from a person’s own cells would obviate the need for organ donors in addition to eliminating difficulties with immune rejection foreign tissues.

Not all interest in gene regulation and expression is aimed at medical advance, however. Understanding gene regulation in greater detail may allow us to better engineer new organisms or better predict the characteristics of newly engineered organisms. Current genetic engineering programs are aimed at producing such organisms as insect-resistant crops that require no application of pesticides and micro-organisms that degrade petrochemicals. Understanding and controlling how these organism operate on the genetic level could vastly improve our ability to produce organisms with specific desired characteristics, as well as give us much more confidence that these organisms will not have unwanted side effects on the environment.

A comprehensive understanding of how genes are expressed and how they interact may also solve some of the key fundamental questions remaining in biology. For instance, understanding gene regulation promises to shed light on which characteristics of

an organism are genetically determined, which are caused by the influence of environmental conditions, and which might be the result of some complex inter-relation between the two. This may help settle the ongoing nature-versus-nature debate in biology. It also remains a mystery just how mere *information* coded in a sequence of DNA can be responsible for producing a living cell. Further, understanding gene expression and regulation may help us to understand just how living cells came to exist in the first place, as well as how and why life evolved from simple unicellular organisms to complex, multicellular plants and animals. Understanding how genes are expressed and regulated is perhaps the last major frontier remaining in the life sciences.

III. Brief overview of gene expression and regulation

How genes are expressed

Proteins are considered the most basic building blocks of life. The term “protein” denotes a general category of unbranched amino acid polymer whose function is determined by its particular physical structure and chemical properties. These properties in turn are determined by the particular sequence of 20 possible biologically-active amino acids as well as the exact manner the amino acid chain is folded into a three-dimensional structure. Proteins perform many different kinds of functions for the cell. We can place the different types of structure and function of protein into broad categories:

- catalytic proteins, such as enzymes
- defense proteins, such as antibodies
- structural proteins, such as fibronectin
- transport proteins, such as albumin
- contractile proteins, such as actin/myosin
- gene regulatory proteins

The very existence of life is made possible by thousands of different proteins acting at the right times and right places in a cell, as if they were performing a highly-choreographed molecular ballet. The total number of different proteins, the “cast members,” that can be expressed in each cell is enormous. There are an estimated 150,000 different proteins expressed in all the cells of the human body alone, although

only a fraction of this total is expressed in any one cell at any one time. The diversity of life is also made possible by the many different possible combinations of proteins, these combinations being responsible for producing a virtually endless number of different cell types and organism structures. Each form of life, from algae to dandelions to zebras, is fundamentally a function of the particular assortment of proteins produced in the cells that make up the organism [3,7,8].

What determines which protein is made by the cell and when? The specific amino acid sequence of each protein is encoded by the genetic code, consisting of linear chains of four nucleotide bases: adenine (A), cytosine (C), thymine (T), and guanine (G). The linear chain is called deoxyribonucleic acid, or DNA, and can stretch for thousands or millions of nucleotide bases. Each set of three consecutive bases, for example CGA, comprise a “word” known as a codon, and each codon specifies a particular amino acid (in this example, “CGA” would code for the amino acid arginine). The linear DNA molecule is directional in its structure, with one end designated the 5’ end the other the 3’ end (this particular naming is due to the standard numbering scheme of carbons on the ribose sugar molecule) (see Figure 1).

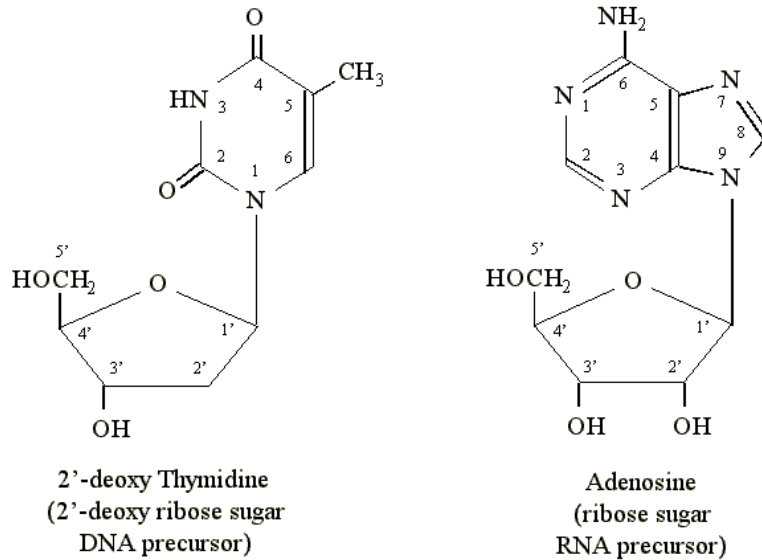


Figure 1.

DNA exists in the cell normally in double stranded form, meaning that two linear DNA strands form a pair of strands in which the sequence of one strand is exactly complementary to the sequence in the other. The strands join to each other because of the specific pairing between the bases cytosine (C) and guanine (G), and between adenine (A) and thymine (T) respectively². As an example of a short stretch of double-stranded DNA, one strand might be 5'-CCGAT-3', while the complementary strand would be 3'-GGCTA-5'.³ When two such complementary strands join together to form double stranded DNA, the process is called *hybridization*. Whenever the genome is discussed then, it is assumed that both strands are present, but the code is considered only on one strand from the 5' to 3' end.

²This pairing is due to the molecular shape of each molecule. Cytosine's shape fits that of guanine as would a hand to a glove, but not to adenine or thymine. Conversely, adenine's shape fits that of thymine's, but neither that of cytosine or guanine.

³ Because the genetic code is present in two complementary DNA strands, there is inherent redundancy that allows for repair of one strand if the other is damaged.

How the genetic code is ultimately expressed is in very general terms the same in all organisms. However, there are profound differences in the actual gene expression mechanisms found in simpler life forms such as single-celled bacteria and archaea (prokaryotes) and those in higher cells that contain organized nuclei (eukaryotes), such as those found in plants and animals.

In both the prokaryotic and eukaryotic cases, DNA is *transcribed* into messenger RNA (mRNA) by an enzyme known as an RNA polymerase. During transcription the RNA polymerase first binds to a specified initiation site on the linear DNA coding region, and then begins to move along it from the 5' end to the 3' end. As the RNA polymerase moves along the DNA molecule from beginning to end of a coding region, it produces a corresponding linear strand of mRNA in a sequence complementary to the original DNA by sequentially adding the appropriate nucleotide to the new RNA chain (with uracil in lieu of the thymine found in DNA) one base at a time. At the end of the coding region the RNA polymerase reaches a stop or termination signal, in which case the completed mRNA strand and the RNA polymerase molecule are both released.

The mRNA is then *translated* into the amino acid sequences that ultimately make up proteins. Amino acid chains, known as polypeptides, are produced by translation of the mRNA described above. This is accomplished by a spool-shaped structure that rolls along the mRNA, in conjunction with individual “adapters” which “plug” the appropriate individual amino acids into the end of the sequence extending the chain one amino acid at a time. The spool-shaped structure is called a *ribosome*, and the “adapters” are known as *transfer RNA* (tRNA).

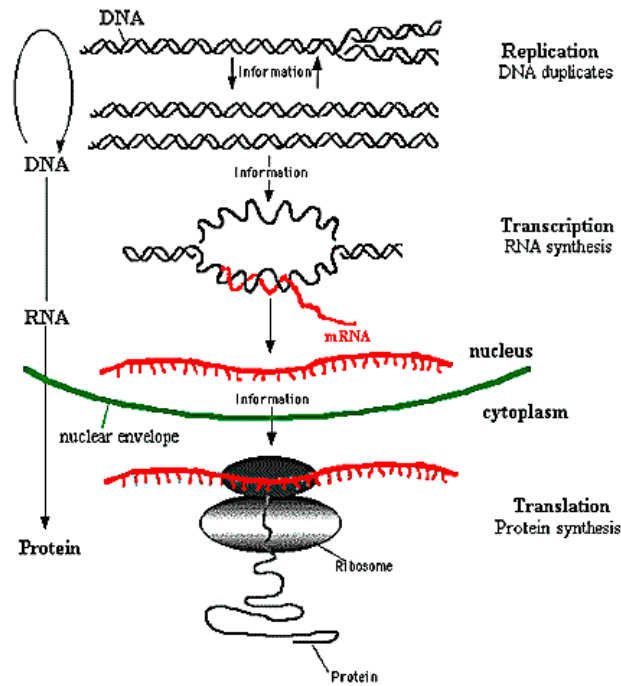


Figure 2.

The polypeptide chain that comes off the ribosome is then sometimes processed by certain enzymes to give it a specified shape and functionality⁴. Some of these enzymes fold and shape the amino acid chains, others snip specified portions off here and there, and yet others add non-amino acid groups such as the iron-containing heme molecule that binds oxygen in hemoglobin. Once folded and configured, some polypeptide chains attain full functionality as free-standing proteins, while others might serve to go on as subunits in a larger complex of protein molecules.

While many of these proteins go off to serve the cellular functions described above, a number of these proteins, the *regulatory* proteins, go on to serve as activators,

⁴ This is known as *post-translational modification*.

enhancers, or inhibitors of further gene transcription. In this way, proteins serve as the main intermediary between each and every gene expressed in a cell.

By knowing how much of each type of mRNA is being produced in a cell, we also have an indication of the amount of each protein product is being produced. mRNA concentration can therefore be regarded as an indicator of gene expression activity. The next section will describe one way to measure a particular pattern of gene expression activity via quantifying individual mRNA concentrations.

Gene regulation in prokaryotes and eukaryotes

Prokaryotes include *bacteria* as well as the less well-characterized organisms called *archaea*. Nearly all prokaryotes are motile, free-living organisms, but in some instances they may live in colonies or organized communities. Eukaryotes may be single-celled as in protozoa and yeast, but are also the constituents of all the multicellular organisms with which we are familiar, such as animals, plants, and fungi. Prokaryotes lack a nucleus; genetic material is loosely concentrated near the middle region of the cell. Eukaryotes, in contrast, have a well defined nucleus surrounded by a nuclear membrane which contains the cell's genetic material. Prokaryotes also divide by simple binary fission, while eukaryotes typically divide in a highly organized fashion known as mitosis [25]. The manner in which the genome is organized and regulated in prokaryotes differs markedly from that in eukaryotes, so it might be reasonable to discuss them separately.

Prokaryotes

Prokaryotic organisms are almost always unicellular and are well-known for their metabolic diversity. They are thus often classified in terms of being *autotrophs* or *heterotrophs*. Autotrophs need only CO₂ as a source of carbon, while heterotrophs require an organic compound such as a sugar for its carbon source. Prokaryotes can also be classified as being either *phototrophic* or *chemotrophic*, i.e., using either photosynthesis or chemical bonds as an energy source [8]. The type of prokaryote known as archaea can also be classified in terms of being methanogens (creating methane from hydrogen sulfide), extreme halophiles (preferring salt environments), or extreme thermophiles (preferring high temperatures)⁵. The point here is to highlight just how diverse and extreme the environments are in which prokaryotes can live, and also how quickly and extensively their gene expression patterns can change to adapt to those conditions. This is in remarkable contrast to eukaryotes which often live in comparatively tranquil physiological environments. As we shall see shortly, the gene regulatory mechanisms in prokaryotes are often much simpler and contain fewer intermediate biochemical steps than those found in eukaryotes.

The prokaryotic genome consists of a single main molecule of circular DNA near the center of the organism.⁶ This main genomic DNA is often supplemented with smaller circular DNA molecules, known as *plasmids*, that code for only a few proteins. A bacterial cell doesn't always contain plasmids, but the presence of particular plasmids

⁵ The heat-stable DNA polymerase that makes PCR amplification possible at the necessary high temperatures used is derived from the extreme thermoacidophile organism *Thermus aquaticus* found in geothermal hot springs.

⁶ This solitary DNA molecule is often loosely referred to as a chromosome, although often the term *chromosome* refers only to the DNA molecules found in the eukaryotic genome.

may produce specific proteins that might confer antibiotic resistance or some other such property upon the bacterium. A bacterial cell can therefore contain anywhere from zero to perhaps several plasmids. Plasmids replicate independently of the bacterial genome, and once replicated may freely exit the bacterium to enter another bacterium. Plasmids therefore represent one mechanism of genetic exchange between individual bacteria, providing a means by which certain characteristics such as antibiotic resistance may spread among a bacterial population. Plasmid protein expression is not under the control of the main genomic regulatory mechanisms.

Prokaryotic DNA is generally compact in terms of gene arrangement, containing few if any non-coding regions. Sometimes a coding genetic sequence actually overlaps another coding sequence, producing a different protein depending on exactly where translation starts. Sometimes this overlap is simply dependent on a frameshift of a single nucleotide. The prokaryotic genome is therefore considered to be much more dense and compact in terms of information coding than that of eukaryotes [3,7].

Prokaryotes are also constantly replicating their genomes. Unlike most eukaryotes, prokaryotes propagate strictly through simple binary fission. There is no general scheme for an orderly partitioning of genetic material from parent to progeny. A prokaryote simply replicates its genome, and then divides into two daughter organisms that each contain one copy of the replicated parent genome.

With respect to actual gene expression, prokaryotes follow the basic pattern of DNA transcription and translation described earlier. The DNA is transcribed starting at an initiation site for the particular gene by an RNA polymerase. The resultant messenger RNA (mRNA) is then translated by ribosomes using tRNA attached to individual amino

acids to produce a polypeptide chain that will ultimately become the desired protein molecule. Unlike in eukaryotes, prokaryotic mRNA is not modified or transported into the cytoplasm prior to translation—in fact mRNA translation to a polypeptide chain often begins even while the mRNA is still in the process of being transcribed.

In prokaryotes, the site of initiation of transcription is preceded by *promoter* site. The promoter site has two components, one approximate ten nucleotides upstream (-10) and the other approximately thirty-five nucleotides upstream (-35). The -10 site usually has a string of nucleotides having the sequence “TATAAT.” During the start of transcription, the RNA polymerase recognizes the specific sequences at the -10 and -35 positions, and so attaches to the DNA there. Once attached, the RNA polymerase will immediately begin transcribing in the 5' to 3' direction until it reaches a termination sequence which signals it to stop and release itself from the DNA. Without any sort of regulation, prokaryotic RNA polymerase will attach to DNA at promoter sites at random, resulting in a natural low-level transcription rate. Down-regulation can stop this low-level transcription completely, while up-regulation can increase this basal transcription rate up to a thousand fold [7,44].

Prokaryotes contain only a single type of RNA polymerase that produces all of the mRNA in the cell. The active RNA polymerase is known as a *holoenzyme*, whose main constituent is known as a *sigma* protein (σ). The sigma protein subunit often requires the presence of other factors to make it a whole, fully active holoenzyme.⁷ Thus these extra factors are involved in regulating the RNA polymerase activity (and thus gene

⁷ When cells are shocked by toxic chemicals or heat, they respond by producing protective *heat shock proteins*. These proteins are produced from a specified set of special heat shock genes through a particular variety of the sigma subunit, called σ^{32} . Most genes are transcribed with a σ^{70} subunit, however.

expression activity) [35,44]. Regulation of gene expression in prokaryotes therefore can occur by one of four very general strategies:

First, certain proteins may bind to a particular gene's promoter site on the DNA molecule and actively attract the RNA polymerase to it. This is known as *regulated recruitment*. Thus, proteins that attract the RNA polymerase to the initiation site are *activators*. Where the basal transcription rate depends on random RNA polymerase attachment to the promoter site, these activators actually actively recruit the RNA polymerase to that site resulting in an increase in transcription activity at that gene [30].

Second, certain proteins may simply bind to the inactive sigma subunit to produce a fully-functioning holoenzyme. This is called *polymerase activation*. An example might be a protein that binds to a metabolic precursor, intermediate, or end-product, such as a particular protein known as catabolite activator protein (CAP). These proteins again serve as activators of gene expression that respond to the presence of particular molecules. In this way polymerase activation serves as a kind of "sensor," increasing gene expression in response to the presence of nutrients, toxic molecules, or catabolic intermediates [35,44].

Third, in some instances a promoter sequence will bind the RNA polymerase holoenzyme tightly, not allowing transcription to proceed at all. The promoter site will then require activation by a regulator protein that will allow transcription to proceed. This is called *promoter activation*, and is seen to occur with genes responsible for producing proteins that help the cell respond to toxic metals such as mercury [35].

Finally, certain regulatory proteins may bind to a portion of the DNA on the gene and physically block the RNA polymerase, preventing it from transcribing at all. By

binding to the appropriate site on the DNA molecule, these proteins may completely block all transcription of a particular gene. These proteins are called *repressors*.

Thus, gene activity can be up-regulated by activators in the three manners described above, or down-regulated through a repressor. Activators and repressors may be solely the product of genes themselves, or as mentioned above, metabolic precursors, intermediates, or end-products that bind to an activating or repressing protein. In this way adaptive gene regulation can help the prokaryotic organism respond to the presence or lack of a wide variety of nutrients or potentially toxic compounds in the environment. Additionally, in prokaryotes, when several proteins are useful in a single metabolic pathway, these proteins are expressed simultaneously using a single initiation site for several genes. This contiguous cluster of genes is known as an *operon*. Expressing several genes at once that are always need simultaneously via an operon is more efficient and ostensibly quicker than regulating them via separate mechanisms. This is an example of how the prokaryotic genome is regulated more simply (and controlled more quickly) than in eukaryotes.

The specific manner in which particular activators and repressors operate can vary widely depending on the particular gene in question and its role in the organism's metabolism. Thus, the description given here is a simplified version of the general kinds of mechanisms seen in prokaryotic gene regulation. When considering a model of prokaryotic gene regulation, the types of parameters we may wish to consider include:

- transcription rate
- mRNA degradation rate

- translation rate
- polypeptide post-translation modification rate
- concentration and kinetics of precursors or catabolites
- concentration and kinetics of activator(s)
- concentration and kinetics of repressor(s)

Eukaryotes

The eukaryotic genome differs markedly from that of the prokaryote. While the prokaryotic genome exists as a single circular molecule of DNA, eukaryotic genome is highly organized in the form of several individual chromosomes.⁸ Each chromosome consists of one very long double-stranded DNA molecule. The DNA is wrapped compactly around cylindrical proteins called *histones*. Each histone contains almost two full turns of DNA. Each DNA-histone complex is called a nucleosome. In the chromosome, the DNA appears as a very long string of highly organized beads, each bead being a nucleosome. Chromosomes therefore contain a large amount of protein in addition to the DNA with which we are familiar. Chromosomal DNA is also highly structured, supported within a vast scaffolding consisting of millions of histones.

For gene expression to occur, the DNA must be temporarily unbound from the histones that support it. Not a lot is known about the details of how this is controlled, but there are enzymes that temporarily detach the DNA while it is being transcribed. There is

⁸ Strictly speaking, the term *chromosome* denotes the DNA-containing entities that become visible through a light microscope as a cell prepares for division. Most of the time the genomic DNA is maintained in a less-well defined mass (though still highly structured) called *chromatin*.

also evidence that adjacent regions of DNA in a chromosome tend to be expressed or not expressed to a similar degree at the same time. One interpretation of this is that the enzymes that control DNA-histone binding and unbinding serve in some way as overall gene regulators for a large region of DNA, irrespective of the regulators that control individual genes. Adjacent genes might therefore also have related functions, but this has not yet been shown to be generally the case [9].

Another major difference between the genome of prokaryotes and that of eukaryotes is the presence of large non-coding regions of DNA in eukaryotes. In prokaryotes the genome is nearly a continuous coding sequence, whereas in eukaryotes the genetic code is divided into two main types of regions: *exons* and *introns*. Exons are sequences that are ultimately expressed as proteins, while introns are intervening stretches of code that are not expressed at all.⁹

Introns contain a variety of sequences with peculiar characteristics. Some stretches appear to be merely random sequences, while others contain long stretches of repeating nucleotide sequences. Introns also contain strange sequences known as *transposons* that can jump out of one place in a chromosome and insert themselves in another place, or even to a different chromosome altogether. In humans, non-coding sequences account for 97% of the genome, while only 3% of the genome actually codes for proteins [7].

The manner in which genes are expressed in eukaryotes is also much different from that in prokaryotes. In prokaryotes, the cell's machinery transcribes the DNA directly into messenger RNA (mRNA). In eukaryotes, the procedure is slightly more

⁹ The way to remember this is *exon* is *expressed*, *intron* is *intervening*.

complicated. First, a complete stretch of DNA sequence containing both exons and introns is transcribed into pre-messenger RNA. Then, the non-coding intron regions are cut out of the pre-messenger RNA sequence, and the remaining pieces are spliced into a coding mRNA consisting entirely of concatenated exons. This spliced mRNA is then given a “cap” of 7-methylguanosine at the 5’ end and poly-adenylated at the 3’ end. Polyadenylation consists of adding 50-300 consecutive adenosine nucleotides to the 3’ end. The capping and polyadenylation is believed to protect the mRNA from degradation as it is transported from the cell’s nucleus into the cytoplasm, where it is used as a template for production of proteins [7].

The finished mRNA is typically transported to the rough endoplasmic reticulum (known as *rough ER*) in the cell. The “rough” in the name denotes the granular appearance of that part of the cell in the microscope due to the presence of ribosomes. In the rough ER, proteins are produced through translation of mRNA as just described for prokaryotes. The finished proteins are then distributed throughout the cell where they are needed.¹⁰

Eukaryotic RNA

It should be mentioned that in eukaryotes all coding DNA does not necessarily code for proteins. Transfer RNA and ribosomal RNA sequences are also encoded as DNA. In eukaryotes, roughly 70% of the expressed genome (exons) code for large ribosomal RNA, 20% code for mRNA and small nuclear RNA (snRNA) involved in

¹⁰ At this point one area of confusion (or perhaps carelessness) should be mentioned. It should be remembered that a *gene* is not a *protein*. Genes do not have functions of their own, even though we often discuss genes as having this or that function. Rather, it is the proteins encoded by genes that have function. [36]

splicing, and 10% codes for transfer RNA and small ribosomal RNA. The DNA encoding RNA simply undergoes transcription, but no translation. In other words, the desired end product is RNA only, so there is no need to go beyond the transcription step from DNA to RNA.

Eukaryotes also do not have a single type of RNA polymerase as in the case of prokaryotes. Eukaryotes have three, which are designated RNA polymerase I (Pol I), RNA polymerase II (Pol II), and RNA polymerase III (Pol III). Pol I transcribes DNA that codes for most of the RNA that, together with certain proteins, make up the ribosomes in cells.¹¹ Pol II transcribes DNA into mRNA similar to the single type of RNA polymerase we saw in prokaryotes.¹² Pol III transcribes transfer RNA (tRNA).¹³ We can refer to the types of genes transcribed by each of these polymerases as type I genes, type II genes, and type III genes respectively [35,44].

The point here is that when we talk about a genome, we often just think of genes that encode proteins. However, genes encode for much more than just proteins. If we wish to speak about only that portion of the genome that is ultimately expressed as mRNA, then the proper term is *transcriptome*.

Non-regulatory factors in determining gene expression activity

At this point, before we even get into the discussion of gene regulatory mechanisms, we should highlight the relationship between genes and the proteins they

¹¹ Ribosomes are RNA-protein complexes.

¹² Pol II also transcribes a particular type of RNA called small nuclear RNA, or snRNA whose function is not well understood.

¹³ Pol III also transcribes the remaining ribosomal RNA not transcribed by Pol I, in addition to a number of miscellaneous types of RNA called U6-snRNA, snoRNA, and scRNA.

encode. First, a single gene does not necessarily encode a single protein. Alternative ways of splicing exons into mRNA can produce different proteins from the same pre-mRNA. Also, different manners of folding the nascent polypeptide as well as different types of post-translational modification can produce different proteins from the same starting mRNA. Thus, from one gene we may potentially get several different types of protein.

Further, the expression level of any one protein does not necessarily arise from a single gene. The enzymes responsible for folding nascent polypeptides and performing post-translational modification of polypeptides are encoded as genes themselves. In some cases then, the expression level of a particular protein may depend on the simultaneous expression of several genes at once. This would mean that many genes are responsible for producing one protein. Thus, in such an instance, mRNA expression levels for a particular gene would not necessarily reflect the true expression levels of the protein for which it might encode. We would have to take into account the mRNA expression levels for the ancillary enzymes that are responsible for post-translational modification as well [4].

We see then that understanding gene regulatory mechanisms are not quite so simple as we might have assumed—genes do not necessarily have a one-to-one relation with proteins. Rather, they can have a many-many relationship. Also, contrary to what we might have liked, explicit gene regulatory mechanisms are not the only mechanisms involved in the control of gene expression in terms of their protein end product. There are many factors involved in determining the expression levels of protein end-products. Further, when measuring mRNA levels, we should remember that these levels do not

always necessarily reflect protein levels in the cell. Finally, it is not mRNA but the protein end-products that exhibit the actual biological activity within the cell.

Eukaryotic gene regulatory mechanisms

The types of mechanisms involved in regulating gene expression activity in eukaryotes is far more complicated than in prokaryotes. For one thing, the different RNA polymerases, Pol I, Pol II, and Pol III, each involve different mechanisms of polymerase activation. Since we are mostly interested in mRNA expression levels here, we will discuss only the mechanisms involving Pol II and type II genes here, but it should be remembered that the mechanisms involved in the other two types of RNA polymerase are different but nearly as sophisticated.

In eukaryotes, there are promoter sequences for type II genes similar to that found in prokaryotes. One sequence recognizable by an RNA polymerase occurs as a “TATA” sequence around the -25 position in the gene. There may be one or more upstream promoter elements (UPE) as well, sometimes hundreds of bases away.

In eukaryotic transcription of type II genes, the RNA polymerase first forms a pre-initiation complex. This complex may consist of over a dozen different proteins that, together, form a fully active RNA polymerase unit. The proteins are known as transcription factors (TF). One transcription factor is distributed throughout the nucleus and will work with every gene. This is called a general transcription factor (GTF), but is more specifically known as transcription factor IID or TFIID. Others have activity only when associated with specific genes or types of genes. These are called specific transcription factors.

The protein responsible for recognizing the promoter of the gene is called TATA-binding protein (TBP). There are also up to twelve TBP-associated transcription factors (TAF) that may be specific for the particular gene being transcribed. To form a pre-initiation complex, TFIID and TBP first recognize the TATA promoter element. Then the necessary TAFs join the complex. At this point, the Pol II complex becomes an active RNA polymerase and begins transcription. Because the formation of an active RNA polymerase complex requires so many elements joined at once, the types of random transcript initiation and basal transcription rates we saw in prokaryotes do not exist in eukaryotes [3,7,35]. Transcription in eukaryotes is therefore more of a “deliberate” activity.

The various specific transcription factors just mentioned are not only those that recognize a promoter element at the -25 position. Some recognize the various upstream promoter sequences, if present. Some also recognize *enhancer* sequences that might be anywhere on the DNA molecule. These enhancers differ from promoters because they can be anywhere on the DNA molecule, and their presence may increase the transcription rate of several genes at once. Some transcription factors affect nucleosome positioning, releasing the DNA from its DNA-histone complex so that it is exposed and able to be transcribed. Finally, some transcription factors are involved with forming the pre-initiation complex as discussed previously.¹⁴

How do transcription factors control gene expression in eukaryotes then? Every gene seems to require a different scheme. Some genes' expression is controlled only by

¹⁴ The transcription factors that recognize upstream promoter elements, enhancers, and that affect nucleosome positioning are DNA-binding transcription factors. The transcription factors that activate the pre-initiation complex only undergo protein-protein interactions.

the level of general and specific transcription factors synthesis. When the requisite transcription factors increase in concentration and are all present simultaneously, the genes are expressed. This type of regulation is slow to respond to changes, however. Up-regulation depends on building up concentrations of the transcription factors and could take some time. Similarly, down-regulation relies on the decay of the transcription factors through normal molecular degradation that happens over time (i.e., “wearing out”). Because this type of gene regulation responds so slowly, it is used in situations where the gene expression levels are more or less stable, such with genes involved in cell differentiation and development.

In other instances gene expression is regulated by direct activation caused by the presence of transcription factors within the cell. Gene expression can be up-regulated by transcription factors that increase the rate of transcription, or it can be repressed by certain transcription factors that inhibit transcription. Genes that express the proteins for transcription factors therefore are responsible for gene-to-gene regulatory interactions.

Gene expression can also be affected by molecules outside the cell boundary, however. Molecules of a particular shape, such a steroid hormone molecule, may fit into a specific receptor on the cell’s surface or the surface of the nuclear membrane. The shape of the receptor is specific for a particular shape of molecule. When the receptor is occupied by a molecule that fits into it correctly, the receptor initiates a cascade of signaling inside the cell. This is known as a *second messenger* system. Second messenger systems can control a number of molecular pathways inside the cell, including the activity of transcription factors. Thus, hormones or drugs applied externally to the cell can, via the activity of transcription factors, up- or down-regulate particular genes.

Finally, one way in which gene expression activity can be suppressed completely is by *methylation* of cytosine. In some instance, an enzyme will attach a methyl group to each of the cytosine nucleotides within entire stretches of DNA. This will prevent transcription of that DNA entirely [7]. When the DNA is to be transcribed, another enzyme will remove the methyl groups where necessary.

Some gene expression regulatory mechanism are therefore amenable to relatively rapid changes, while others are essentially permanent or semi-permanent set into place. Permanent changes in gene expression result from DNA rearrangements, the particular structure of the chromosome/chromatin that enables or disables transcription, and in some cases through the transition of gene expression pathways into steady-state, self-maintaining feedback loops.

Thus, gene expression activity in eukaryotes may depend on these factors:

- chromatin accessibility
- nucleosome binding and exposure of DNA to RNA polymerase
- transcription factor synthesis rate
- transcription factor degradation rate
- transcription rate
- mRNA degradation rate
- mRNA processing (splicing, editing)
- translation rate
- methylation and de-methylation
- post-translational modification

IV. Model representation of gene regulation

If we wish to understand more fully how genes regulate each other, it stands to reason we would like to construct some sort of model or representation of the mechanism. This model might serve two functions. One is to allow us to predict outputs from a given set of specified inputs. The other function is to give us a better overall understanding of the causal connections between genes.

Our interest in creating a model of gene expression should be contrasted with researchers whose goal it is to identify *markers* of disease. For instance, as mentioned previously, many researchers are interested in identifying certain types of genes that predispose someone to a particular disease. In some instances, the presence of certain gene products, i.e., proteins, might indicate a pathological state. An example of this would be PSA, or prostate-specific antigen, that tends to be highly expressed in prostate tumor cells. Researchers who are looking for markers of disease tend to look for certain genes or gene products in tissue samples from a diseased individual that are not found in normal controls. When large amounts of gene expression data are involved, these researchers tend to look for clusters of genes that are abnormally expressed in order to identify robust markers of disease. This goal is reflected in the statistical tools and methods they bring to bear on this problem. This approach, then, is aimed at being more of a diagnostic tool than an investigatory one. Much of the “gene expression” literature is of this nature.

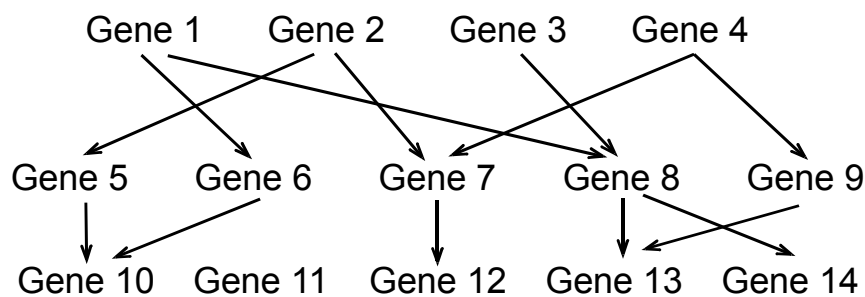
In contrast to this is gene expression analysis aimed at elucidating the mechanisms involved in gene regulation. This requires the construction of some sort of model as mentioned previously. A natural representation of gene regulatory networks

would be a causal representation, in which one would identify which gene(s) regulate which other gene(s).

Causal models and DAGs

In discussing gene regulation, we naturally view one gene as controlling whether another gene is expressed or not. In this instance, the expression of the *regulator* gene would be a *cause*, while the expression (or lack of expression) of the *regulated* gene would be considered the *effect*. Therefore, inherent in our understanding of gene regulation is the idea of a causal model. In the simplest case, in which we represent a cause by a variable A and the effect by a variable B, we can say that A causes B.

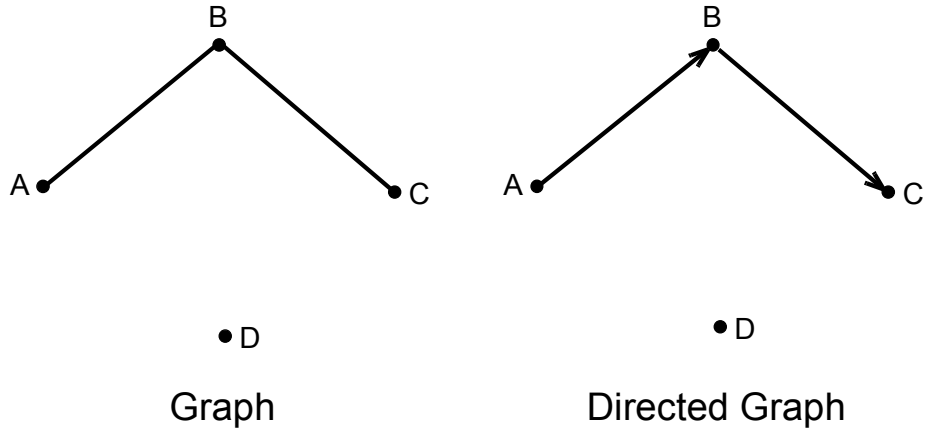
As already discussed, genomes consist of multitudes of genes; further, rarely does a single gene solely control only one other gene. In some cases, one gene may single-handedly control the expression many others, while in another case several genes may control the expression of one gene. Also, in some instances one gene may be regulated by another gene, but may itself serve as the regulator of one or more others. Hence, instead of a simple “A causes B,” we have a complex network of multiple causes and multiple effects.



If we wish to represent this sort of causal network mathematically, a natural place to start would be to consider a standard mathematical representation known as a *graph*. A graph consists of one or more vertices (represented as points) along with a representation of the relationship between any two vertices in the form of edges (represented as lines).¹⁵ Thus, a graph might consist of a number of points, with some or all of those points connected by lines. The points (vertices) designate the variable of interest, while the lines between them (edges) signify whether some relation exists between those points. By itself, a standard graph such as this will not adequately represent the sort of causal relationship we want to represent because we need some way to distinguish between what variable is the cause and what is an effect. For this, we need to add some sort of provision for directionality to the edges that connect the vertices. For this we use a *directed graph* [10].¹⁶

¹⁵ The technical definition is: A graph G is a finite nonempty set V together with an irreflexive, symmetric relation R on V . [10] Reflexivity simply means that something can have the specified relation to itself, e.g., if John loves John (himself) then in this case the relation “loves” is reflexive. This provision of denying reflexivity in our definition of a graph will be necessary in our case because we want to rule out representations of something causing itself.

¹⁶ For the technical definition we simply need to remove the condition requiring symmetry: A directed graph G is a finite nonempty set V together with an irreflexive relation R on V . [9]

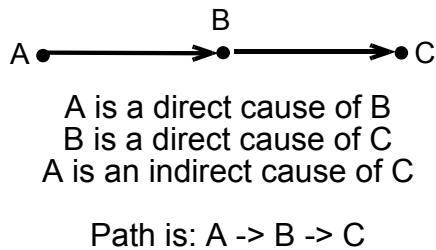


In a directed graph, we simply use a *directed edge* instead of just an edge. The directed edge is (not surprisingly) represented by a single-headed arrow. Hence, a directed graph may have a number of vertices, each possibly connected to another by an arrow. The vertex at the tail of the arrow represents the variable signifying the cause, while the vertex at the head of the arrow represents the variable signifying the effect [10].

We are not quite finished, however. So far we do have an adequate representation of a causal network, but a directed graph as described above does not rule out the representation of something causing an effect which in turn serves as the cause for the original cause (A causes B and B causes A). In such a situation we would have a causal cycle. Similarly, we could have any number of cyclic representations involving any number of variables, such as A causes B causes C causes A again. Such cycles set up feedback loops that require a complex and difficult mathematical treatment. For simplicity we will restrict ourselves to directed graphs that do not have any cycles, known as *directed acyclic graphs* (DAGs) [12,39].

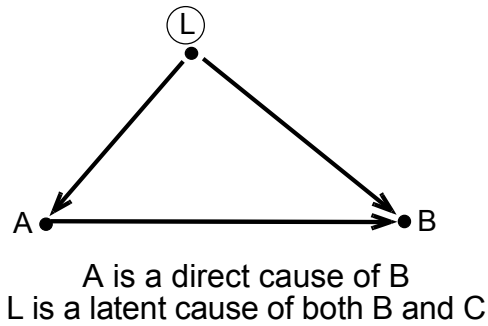
Terminology

There is some terminology associated with DAGs. First, we can make a distinction between a *direct* cause and an *indirect* cause. A direct cause is simply a cause with no represented intermediate variables, whereas an indirect cause is a cause which is mediated by one or more other represented variables. A sequence of connected variables in which there is no reversal of directed edges is called a *directed path* [2,39].

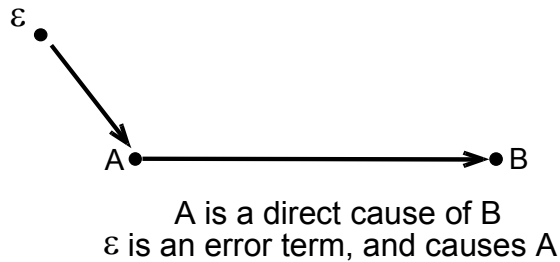


A variable that is a direct cause of an effect is called a *parent* (in this instance, A is a parent of B). The effect is similarly denoted a *child* (B is a child of A). A variable that is an indirect cause of an effect is called an *ancestor* (A is an ancestor of C), while a *descendent* is an effect of an indirect cause (C is a descendent of A). If an arrow connects two variables B and C in either direction, we say that B and C are *adjacent*. If two arrows run into a variable from two other variables, then the middle variable is called a *collider* [12,39].

When representing a causal system we sometimes have a cause or number of causes that are not readily identified or measured. We call this a *latent* cause, and represent it by a *latent variable* [12,39].



Further, we might also have random unmeasured causes that contribute to variation in a specific measured variable. This is often called noise or error, and is typically represented in a DAG with the letter ϵ .



Variables

Now that we have a DAG as a representation of the causal network, we need to specify the nature of the variables we will be using. The simplest case is a *boolean* variable which takes one of two states, such as “on” or “off,” or “yes” or “no.” For instance, if we wish to represent a causal model for the relationship between smoking and

We also have some probability distribution over the vectors of values our represented variables can take. For instance, in the case of a fair die, there is 1/6 chance of getting “1,” a 1/6 chance of getting “2,” and so on. In the case of lung cancer, we would have to determine the probabilities of having lung cancer as well as the probabilities of smoking.¹⁷ The probabilities assigned to each of the vectors of values our variables can take gives us a *probability distribution function*, or pdf.¹⁸ By “marginalizing” the pdf, each particular variable has some probability distribution function associated with it that will tell us what the probability is of getting any particular value from among the set of possible values for that variable.

Inferring independence relationships from data

Once we have a suitable manner for representing a causal model, and have established the necessary conditions for inferring the relationship between DAGs and the probability distribution over the variables represented in those DAGs, our next task is to actually derive the appropriate causal model from data. We can attempt to do this through two different general strategies: passive data collection or *observation*, or through deliberate variable manipulation in the form of *intervention*.

¹⁷ Probabilities such as this are often assumed to be equal to the observed frequencies or proportions of populations. For instance, in this case we would count the number of people with lung cancer among those who smoke and those who do not smoke. The value of the observed frequency or proportion of a population is known as a maximum likelihood estimate (MLE) for the probability.

¹⁸ This is often just called a “probability distribution.” There is usually an assumption of the particular shape of the pdf based on the data collected, such as a binomial distribution, normal (gaussian) distribution, or uniform distribution, for instance. This assumption is believed to be warranted because there appears to be a finite family of well-characterized probability distribution functions such as these that seem to occur with some regularity in nature. The probabilities are assumed to be limiting relative frequencies. The probabilities are estimated by the observed frequencies.

When we attempt to infer causal information from observation, we collect data on what we hypothesize to be a cause, and also data on that we believe to be the effect as mentioned previously. For instance, if want to know if cigarette smoking causes lung cancer, we count the number of people who smoke who have lung cancer as well as the number of people who smoke who do not have lung cancer. This then gives us four proportions:

- smoking, no lung cancer
- smoking, lung cancer
- no smoking, no lung cancer
- no smoking, lung cancer

If there were no connection between smoking and lung cancer, we would expect the proportions of people with lung cancer to be the same among non-smokers as it is among smokers. We say the proportion of those who both smoke and have lung cancer (joint probability) equals the proportion of those with lung cancer multiplied by the proportion who smoke:

$$\Pr(\text{LC} = \text{yes and S} = \text{yes}) = \Pr(\text{LC} = \text{yes}) \times \Pr(\text{S} = \text{yes})$$

In other words, if the above equality holds, then we have no evidence of a causal connection between the two. We say that the proportion of lung cancer cases is independent of the proportion of smokers. However, if the equality does not hold, then we have demonstrated an *association* between the two variables. An association does not

necessarily mean we understand the causal relationships between the variables, although it is common for people to erroneously infer causation from a mere association. An association *may* occur because there is a causal relationship between the two variables, but it can alternatively mean that there is some other cause responsible for the association between the variables, i.e., a latent cause (as well as other mechanisms). An observational study by itself cannot necessarily distinguish which of these two is the case.

Example 1:

	Smoker	Non-smoker	row sum
Lung Cancer	0.06	0.14	0.20
No Lung Cancer	0.24	0.56	0.80
column sum	0.30	0.70	1.00

In this case, the proportion of those who are simultaneously Smokers and have Lung Cancer is 0.06, while the total proportion of Smokers is 0.30 and the total proportion of those with Lung Cancer is 0.20. Since $0.06 = 0.30 \times 0.20$ we have demonstrated that in this case Smoking is *independent* of Lung Cancer.¹⁹

Example 2:

	Smoker	Non-smoker	row sum
Lung Cancer	0.15	0.05	0.20
No Lung Cancer	0.15	0.65	0.80
column sum	0.30	0.70	1.00

In this case, the proportion of those who are simultaneously Smokers and have Lung Cancer is 0.015, while the total proportion of Smokers is still 0.30 and the total proportion of those with Lung Cancer is 0.20. Since $0.15 \neq 0.30 \times 0.20$ we have

¹⁹ We denote this with the symbol \perp . In the example presented we would denote the independence between Smoking and Lung Cancer as $S \perp LC$.

demonstrated that in this case Smoking and Lung Cancer are *dependent*, or associated in some way.

The example presented thus far exhibit two simplifications that are not always necessarily the case when looking at independence relations. The first is that the example uses boolean variables which have only two states. We have only looked at the dependence relationship between the event Smoking = yes and Lung Cancer = yes. Strictly speaking, to say that the variables are independent, we would have to do this analysis for every combination of variable values.²⁰ This means we would have to look at the other three events as well: Smoking = no and Lung Cancer = yes, Smoking = no and Lung Cancer = no, and Smoking = yes and Lung Cancer = no. In the boolean case, our task is overly simple because we can immediately deduce the independence relations of the others from the one event we analyzed. This is because if the probability of Smoking = yes is 0.15, then it immediately follows that the probability of Smoking = no is $1.0 - 0.15$, or 0.85. It is not quite this simple when we have more than two values for a variable. In that case, to truly prove independence we would actually need to do the independence test for every possible combination of values taken by the pair of variables.²¹

In the smoking and lung cancer example given above, there is a second simplification. In that example we assumed a discrete probability distribution. An example of a discrete probability function would be a function describing the chance of

²⁰ However, if we demonstrate dependence between any two values taken by the variables, then we have demonstrated that the variables are dependent.

²¹ We could get away with $n-1$ tests for each variable because all the probabilities for values of the variable have to sum to 1. Once we know the independence relations for $n-1$ values of the variable, the last one follows immediately.

getting any particular face on a thrown die. If the die is fair, we would assign a $1/6$ chance of getting a “1,” a $1/6$ chance of getting a “2,” and so forth as explained earlier. If we were to plot this distribution, we would have only six separate values on the x -axis.

We can just as well have a continuous probability distribution function, though. The most familiar example is the normal distribution, also known as a gaussian or bell curve, although there are many other common continuous distributions. If we were to plot our continuous normal distribution, instead of having discrete probabilities, we now have a continuous curve instead of discrete points. What does this curve tell us about the probability of getting any particular value we might specify? Since we have a continuous distribution, the probability at any particular point is zero. Therefore, when discussing continuous probability distributions we need to look at intervals, not individual values. For instance, if we have some normal distribution, we might ask what the probability is of getting a value between 0 and 0.5, or between 0.1 and 0.2, or between 0.005 and 0.010? In probability theory, we refer to these intervals as *events*, just as in the discrete case. The probability is then given as the area under the curve bounded by the endpoints of the interval, i.e., the integral evaluated from one endpoint of the interval to the other endpoint.

Now, how do we prove independence between events in the continuous case? Mathematically it is a bit more cumbersome, but involves the same intuition as the discrete case. The same independence equation holds as before, although now we might be a little more explicit that we are talking about independence of the probabilities of events:

$$\Pr(\text{event } A \text{ and event } B) = \Pr(\text{event } A) \times \Pr(\text{event } B)$$

As just mentioned, in the continuous case our events are actually intervals. Each event, A and B, might belong to two separate probability functions. Independent events entail that the probability of the two events jointly equals the product of the probabilities of the two events separately. To say that the two *variables* are truly independent then, not just two events, we would have to look at the entire probability distributions of each. Therefore, if the joint probability distribution (the two probability distributions together) can be factored into the produce of each of the two marginal probability distributions separately, then we can say that the two distributions are independent, and therefore that the two variables are independent [39]. For normal distributions, independence of two variables is equivalent to their vanishing correlation or vanishing covariance.²²

Multivariate systems

Very rarely are we interested in a looking at a causal system involving only two variables. Often, we are interested in systems involving several variables, and in the case of gene regulatory networks the number of variables can run into the thousands. Therefore we need a mathematical treatment for looking at the relationship between more than two variables, i.e., a multivariate system. The simplest such case is a system involving three variables, so we will start there.

Suppose that in addition to smoking, we wanted to see if eating vegetables rich in beta carotene were involved in either preventing or causing lung cancer. In this case we could assign three boolean variables, Lung Cancer [no, yes], Smoking [no, yes], and

²² Another way of explaining this is by saying that if we regress one variable on the other, if the variables are independent then we will get zero correlation between the two.

Vegetables [no, yes]. Now we can look at two variables in relation to the third variable. When looking at independence relations this way, we call it *conditional independence*. That is, we want to know if lung cancer is associated with smoking given that someone eats vegetables, for instance.

First then we have to introduce the idea of conditional probability. We denote conditional probability as $P(A=a|B=b)$, meaning the probability that A will have the value a given that it is the case that B has the value b . In our example this becomes:

$$P(\text{LC} = \text{yes and S} = \text{yes} | \text{V} = \text{yes}) = P(\text{LC} = \text{yes} | \text{V} = \text{yes}) \times P(\text{S} = \text{yes} | \text{V} = \text{yes})$$

This states that the probability of being a smoker with lung cancer given that one eats vegetables is equal to the probability of having lung cancer given that one eats vegetables multiplied by the probability of smoking given that one eats vegetables. If this equality holds, then we say that lung cancer and smoking is independent conditional on eating vegetables. In this instance, this would mean that regardless of whether not one eats vegetables, if that person is a smoker he or she runs the same risk of getting lung cancer as they would otherwise.

With conditional independence relations then, we have a way of looking at two variables in relation to a third. We are not required to condition on only a single variable at one time. We could just as well condition on two, three, or any number of variables while we look for an independence relationship between any two other variables.

Inferring causal relationships from data

Inferring causal information (as opposed to association) from observational data requires us to examine some assumptions concerning the relationship between DAGs and the conditional independence relations derived from data [39]. The two primary assumptions required to bridge DAGs and probability distributions are the Causal Markov Condition and Causal Faithfulness. In addition, an assumption known as Causal Sufficiency is relevant to inferring gene regulatory causal network structure (and representing it in the form of a DAG) from gene expression data.

Causal Markov Condition

Previously we saw two ways of representing probability distributions over variables. We can diagram the probability distributions and the joint probability distributions in the form of DAGs, and we can also write down the independence relations we know entailed by the joint probability distribution over the variables. If we want to be able to infer causal relationship information in the form of DAGs from data, we need to know how DAGs might be related to independence relations discovered in the data.

First, we can make a very important and useful assumption known as the Causal Markov Condition developed by Spirtes, Glymour and Scheines (1993) that tells us what independence relations are entailed by a particular DAG. Let us first state this condition formally:

Let G be a causal graph with vertex set V and P be a probability distribution over the vertices in V generated by the causal structure represented by G . G and P satisfy the Causal Markov Condition if and only if for every W in V , W is independent of $V \setminus (\text{Descendants}(W) \cup \text{Parents}(W))$ given $\text{Parents}(W)$. [39]

Informally, we can say that any variable W is independent of all other variables except the parents and descendants of W , given the parents of W . Direct application of this condition to a DAG gives us certain independence facts about the distributions generated by that DAG.

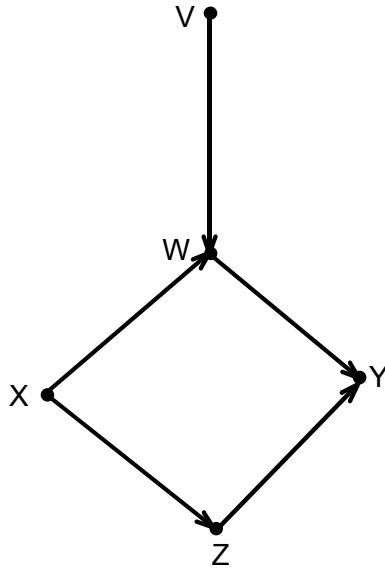
The significance of this is that given an hypothesis about what a particular casual structure looks like, we can write down the independence relations that should be evident from analysis of data. This is important because application of the Causal Markov Condition is what allows us to do a search for an equivalence class of causal structure (i.e., DAG) based upon observational data.²³

Consider as an example the following DAG:

²³ A useful theorem that follows is that given the Causal Markov Condition, we may factor the joint probability distribution of the variables into the product of conditional probabilities as:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | pa(X_i))$$

This makes calculation of probability distributions from DAGs much easier. Instead of having to calculate every the probability distribution of every variable conditional on every combination (and subset of combinations) of every other variable, we only need to know the joint probability distribution over all the variables and the conditional probabilities of each variable and its parent.



The Causal Markov Condition applied to this DAG implies the following conditional independence relations:

$$X \perp\!\!\!\perp V$$

$$V \perp\!\!\!\perp \{X, Z\}$$

$$W \perp\!\!\!\perp Z \mid \{X, V\}$$

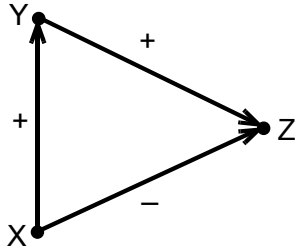
$$Z \perp\!\!\!\perp \{V, W\} \mid X$$

$$Y \perp\!\!\!\perp \{X, V\} \mid \{W, Z\}$$

Faithfulness

While we have seen that by applying the Causal Markov Condition to a DAG we can infer certain independence facts about the distributions generated by that DAG; however, might there be other independence relations given by the distributions as well that are not given to us by the Causal Markov Condition? We can imagine one situation where this could be the case. Consider the case where two variables X and Y are each direct causes of a third Z (i.e., a collider), but where the effect of X through Y produces an

increase in Z while the influence of X directly on Z causes an equal and opposite decrease in Z :



The effect of X through the two causal pathways therefore exactly cancel any combined effect on Z . In this case, the DAG would correctly represent the causal pathways between each of the variables X , Y , and Z . However, in this situation there would exist an apparent independence relation between X and Z not entailed by application of the Causal Markov Condition to the DAG. This however is an extraordinary and highly coincidental case—competing causes must exist in exactly equal and opposite way for such a complete cancellation effect to occur.

When all and only the independence relations of distribution P are entailed by the Markov condition applied to G , however, we say that P and G are *faithful* to one another. We may also say that that a distribution P is faithful provided there is some DAG to which it is faithful. Formally, we can state this as the Faithfulness Condition:

Let G be a causal graph and P a probability distribution generated by G . $\langle G, P \rangle$ satisfies the Faithfulness Condition if and only if every conditional independence relation true in P is entailed by the Causal Markov Condition applied to G .

We normally feel justified in accepting the Faithfulness Condition because situations that would violate it, such as the example above, are argued to be very rare. In the context of gene regulation the particular scenario violating the Faithfulness Condition described above seems to be even more implausible—it would be quite surprising that the effect of a gene might have two opposite effects, equal in magnitude such as to cancel exactly. One could argue that the inefficient bioenergetics involved would be extraordinarily maladaptive. Therefore, there would be a constant selection pressure aimed at purging such gene expression schemes from a population.

Causal Sufficiency

So far we have implicitly assumed that the DAG we are using to represent gene regulatory networks is both complete and exhaustive of all the necessary variables. In other words, we have been making an implicit assumption that all and only the variables we need to correctly represent causal pathways in gene regulatory networks consist of individual genes and their expression levels. Biologically, we know this is not necessarily the case. We may have exogenous factors, for instance, such as nutrient or second messenger molecules that directly affect particular gene expression levels that are not represented in our DAG. Similarly, there may be intermediary mechanisms, such as complex interactions between transcription factor proteins that mediate gene regulation that are not represented in our DAG as well. These may have interactions with each other and/or the environment in ways not represented by the explicit variables used in the construction of our DAG. Realistically then, there might exist unmeasured causes—latent

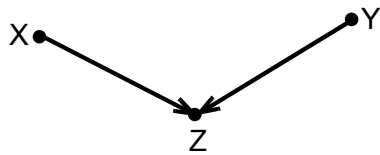
variables—not represented in our DAG that might have a significant influence on the true causal structure.

We may never know exactly all the latent variables present. Even if we are aware of certain latent variables, we may not be able to measure them and therefore we will not be able to assign data-driven probability distributions to them. Intuitively, we may see that we may never achieve an exhaustive list of all the latent variables. As we look with finer and finer detail at the complex biochemical interactions in the cell, we may discover more and more molecular intermediaries or exogenous variables. How then can we know that the DAG we derive from measured gene expression levels is the true model of the gene regulatory network, or whether the model is distorted by the presence of numerous unrecognized and/or unmeasured latent variables? Because of the presence of an indefinite number of latent variables, any number of DAGs may correctly represent the independence relations derived from any particular set of data. We are therefore faced with the question of whether the choice of variables represented in our DAG is *causally sufficient*. In general, it may not be possible to assume causal sufficiency; however in the context of gene regulatory mechanisms, it may be reasonable to assume the variables used in our DAG are causal sufficient. We are already aware that genes are used to regulate other genes, and therefore any intermediaries such as transcription factors can be reasonably subsumed under the directed edges already present in the DAG. Further, since the experimental data used to derive our DAGs consist of controlled experiments, exogenous factors such as nutrient concentration or environmental conditions are intentionally held constant. Therefore, while in general, inferring causal models from data

requires an explicit assurance of causal sufficiency, one may argue that in the context of genetic regulatory networks, the assumption of causal sufficiency is a reasonable one.

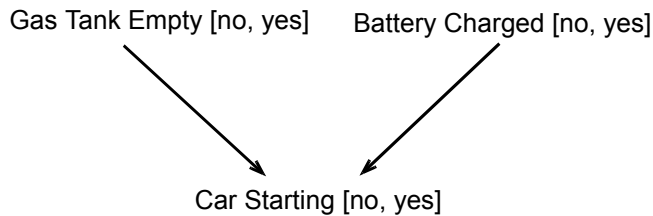
Conditioning on colliders

When we have faithful distributions, we may be able to infer the presence of dependencies between two variables when we condition on the appropriate third variable. Specifically, this situation occurs when we condition on a *collider*.



To get the intuition involved, consider if we have three variables: Car Starting [no, yes], Gas Tank Empty [no, yes], and Battery Charged [no,yes]. Both Gas Tank Empty and Battery Charged are direct causes of Car Starting; the variable Car Starting in this case is a collider. In general, does knowing if the gas tank is empty have anything to do with the battery being charged? No, they are independent. However, what would we be able to infer if we observe the car to start (i.e., Car Starting= “yes”)? In this case, we would know immediately the states of both the variables Gas Tank Empty and Battery Charged (i.e., Gas Tank Empty= “no” and Battery Charged= “yes”). Knowing the state of the variable Car Starting therefore makes the other two variables dependent. Restating

this, we would say that conditioning on a collider produces dependence between the adjacent variables.



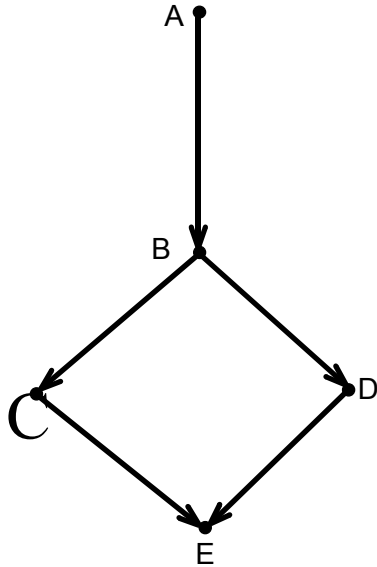
Therefore, if the faithfulness condition is met, if we have variables X, Y , and Z comprising a collider ($X \rightarrow Z \leftarrow Y$), then $X \perp\!\!\!\perp Y$, but X is not independent of Y when conditioned on Z . This is a very useful inference.

Automated causal inference

The significance of the connection between probability distributions and causal graphs is that through analysis of conditional independence relations derived from a set of data, application of the Causal Markov Condition and Faithfulness Conditions allow us to start to get a picture of the true causal graph that initially produced the data. There have been many methods proposed to actually go about doing this, and each has its own advantages and disadvantages. For instance, some methods are computationally more efficient than others, and therefore as a practical consequence we expect them to be carried out in a more reasonable time frame when implemented on a computer. We will

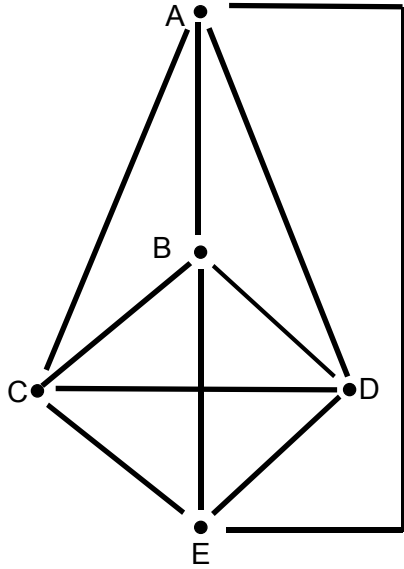
use the PC²⁴ algorithm to illustrate how such a causal search process can be automated in terms of a stepwise procedure that can be implemented on a computer [39]. The algorithm is stated in Appendix A. To give the reader an intuitive idea of how the procedure performs causal search from data we will go through a simple example here.

Imagine this is the true graph (the actual causal structure that generated the data):



Our task, then, is to see how much of this graph we can recover through a causal search algorithm (in this case the PC algorithm). First, we start with an undirected graph in which each variable has an edge to every other variable. This is known as a *saturated* graph.

²⁴ “PC” comes from the first names of the originators of the algorithm, Peter Spirtes and Clark Glymour.



Next, we search for any independence relations between every two variables. We call this a search for *zero order independencies* (we can say that we are conditioning on the *empty set*). Whenever we find independence between two variables, we remove the edge between those two variables. As more independencies are discovered and more edges are removed, the graph becomes more sparse. In the case of this example, we find no zero order independencies so we do not remove any edges.

Next, we begin looking for conditional independencies on combination of subsets of remaining variables. First, we search for any independencies between any two variables conditional on a single remaining adjacent variable. We call this a search for *first order independencies* ($n=1$). In our example, the conditional independencies we find are:

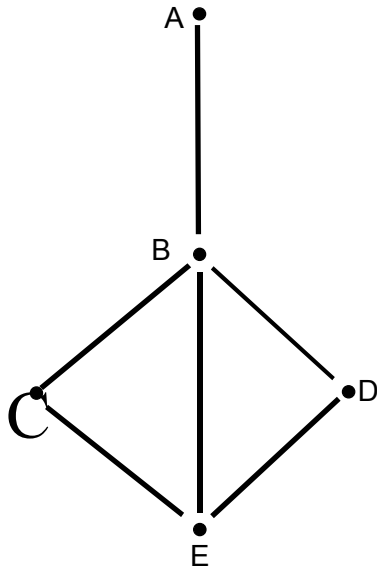
$$A \perp\!\!\!\perp C \mid B$$

$$A \perp\!\!\!\perp E \mid B$$

$$A \perp\!\!\!\perp D \mid B$$

$$C \perp\!\!\!\perp D \mid B$$

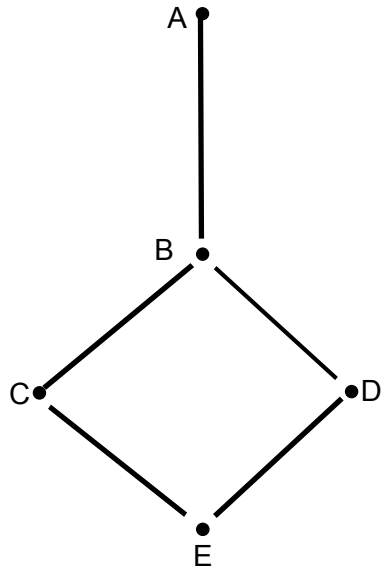
We therefore remove the edge between A and C , A and E , and A and D , and C and D to give us:



Once the search on first order independencies is exhausted, we look for independencies between any two remaining adjacent variables conditional on every subset containing *two* variables having a remaining adjacency to each variable. We call this a search for *second order independencies* ($n=2$). In this example, we find the following conditional independence relation:

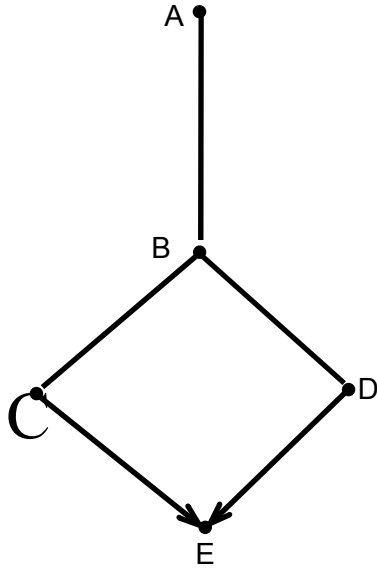
$$B \perp\!\!\!\perp E \mid \{C, D\}$$

And so we remove the edge between B and E to give us:



The algorithm continues until no more independencies can be found in this manner. In this example, the testing stops at $n=2$.

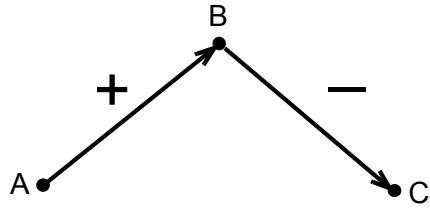
Once the undirected graph is discovered, the orientation part of the algorithm begins. We look at each triple of vertices: A, B, C ; A, B, D ; B, C, E ; B, D, E ; C, B, D ; C, E, D . E is not in the conditioning set that produced conditional independence between C and D (i.e., the Sepset) so $C-E$ and $E-D$ form a collider at E . None of the other triples form colliders. The resultant partially directed graph then is:



In this fashion described above, from observational data we can infer not only the undirected graph based upon conditional independencies between variables, but by applying the Causal Markov Condition we can also sometimes infer the presence of directed edges based upon conditional independencies as well.

Transmission functions

While we have specified the nature of the variables that label our vertices in a DAG, we still have a relationship between the variables signified by the edges. If we wish to represent not only the fact that one variable has a causal relationship with another variable, but actually wish to represent the form of that relationship as well, then we need to assign some sort of function to the edges into a variable. In the simplest case, we can just label each edge with a + or – depending on whether cause A produces an increase, or decrease, in the value of effect B, respectively.



A causes an increase in B,
B causes a decrease in C

In more sophisticated cases we can use any sort of function we would like to use; the difficulty in treating the mathematics obviously become more of a burden, however. In practice, the relationship between variables is often modeled with just a linear equation²⁵, or less commonly, a polynomial. In relatively rare instances the function used will be an ordinary differential equation. In the context of gene regulatory networks modeled by DAGs, we will refer to this function whatever the form as a *transmission function*. In gene regulation, the best established transmission function appears to be very non-linear. Ideally then, in modeling gene regulatory networks (or any such causal representation of a biological system) we would like to have both: 1) the causal model in terms of a DAG, and 2) a transmission function for each edge into a variable in the DAG.

²⁵ A DAG translated into a system of linear equations is called a Structural Equation Model (SEM).

V. Measurement of gene expression

What makes understanding gene regulation difficult?

As just described previously, there are two general strategies for inferring causal information from data. One is to infer a causal graph (or possibly a partial causal graph) from passive observation of data. The other is to intervene, or manipulate the value of a variable, and then observe the effect. To do this in practice, researchers typically conduct gene disruption experiments such as gene knockouts or gene overexpression studies.

Gene knockout experiments compare deletion mutants with wild-type organisms, an approach known as Difference-Based Regulation Finding (DBRF). [32]. Thus, as a first rough estimate using the (dubious) assumption that only single genes regulate other genes, using DBRF on 30,000 genes, one would require 30,000 separate experiments.

Akutsu et. al. have actually performed a much more detailed estimate of the upper and lower bounds for the number of gene disruption experiments needed [2]. They looked at *Saccharomyces cerevisiae* (yeast) with only 6200 genes, and considered a gene regulatory system as being merely boolean (i.e., a gene is either expressed or not).

Assuming the easiest case, in which each regulator gene affects only two other genes, they have shown that in computer science terms the number of separate experiments required to produce the entire regulatory network would be on the order of magnitude $O(n^2)$. Even in under such a best case scenario, which is obviously overly optimistic, the

number of separate experiments required would be enormous. This estimate totally ignores anything beyond pairwise interactions between genes, assumes that genes are always regulated by only a single regulator gene and not some combination of genes, and

assumes that genes are either “on” or “off”—three very bad assumptions in light of our biological knowledge. Therefore, in reality things appear to be much worse than even this pessimistic rough first estimate might even suggest.

An initial thought might also be to glean the entire global gene regulatory network structure using a series of standard molecular biology experiments. Individual molecular biology “bench” experiments, however, are extremely involved and expensive. Aside from the substantial operating costs involved in simply maintaining a laboratory’s infrastructure, individual experiments often require great quantities of supplies and dozens if not hundreds of worker-hours. Molecular biology experiments are very complex, tedious, and usually frustratingly temperamental. Introducing any new variation in experimental design more often than not results in a string of failures before one can start to expect modestly successful experiments. It often takes months to fine-tune the exact parameters, reagents, and apparatus settings involved in any new technique. Often, to avoid cross-contamination in experiments (the most obvious single source of experiment failure), laboratories typically go through vast quantities of single-use, disposable supplies—hundreds of microcentrifuge tubes, pipettes, gloves, reagent vials, etc. Unfortunately, even when an experiment does work, the data generated are often equivocal, or at least difficult to interpret, and therefore experiments are commonly repeated.

Further, many techniques are so involved and specialized that if done in-house are expensive and require experienced technicians, such as techniques involving cell culture and transgenic animals. Animal or human studies involve oversight from an institution’s animal use or human subject committee to ensure compliance with regulations and ethical

standards. Further, molecular biology experiments nearly always involve hazardous reagents and often, radioactive materials and potentially infectious materials that each require expensive safety equipment and employee training.

Consequently, it is not uncommon for any series of molecular biology experiments to take months and cost upwards of \$100,000. Thus, the cost of having some degree of confidence that one understands even simple biological processes at the molecular level is often enormous.

Understanding a single gene regulatory mechanism through bench experiments is expensive and time-consuming. Given the estimated 30-35,000 genes in the human genome, elucidating the mechanisms involved in every gene's regulation would constitute a colossal task. Thus we see that using bench experiments alone, it is for all practical purposes impossible to truly understand the detailed workings of a whole genetic regulatory network in even simple organisms, let alone humans.

A tempting alternative strategy is to try to harness a combination of two recent major technological advances from completely different fields. One is the dramatic improvements in computational power achieved during the last two decades. The other advance is the development in the mid-1990s of DNA microarray technology that makes it possible to examine the expression of thousands of genes simultaneously.

The computational power now commonly found in personal computers makes it much more feasible for the average researcher to simulate or model a large network of interactions, such as that seen in gene expression. This means that instead of only taking the intervention or variable manipulation approach to inferring causal network, we have

the computational power to additionally pursue the observational or passive conditioning approach for helping to infer the correct causal graph.

The newest generation of DNA microarrays makes it possible to obtain large amounts of input data: expression levels of up to a hundred thousand genes can be obtained on a single microarray simultaneously [36], far exceeding the number of genes known to exist in any particular biological system. Using several microarrays within a single series of experiments, it is also feasible to do a number of gene knockout or overexpression studies simultaneously, as well as time-series experiments to help uncover the time course of gene expression. The use of microarrays does not reduce the number of knockout experiments necessary to discover a causal network through experimental intervention, but it does give one a means of simultaneously measuring the effects of manipulating one particular gene (or set of genes such as in the form of multiple knockouts) on all of the other genes. The use of DNA microarrays, then, may not be the perfect solution but it does offer us more hope we can gather the large amounts of data necessary to use either the observational approach, the intervention approach, or some combination of both to help infer the likely causal network involved in gene regulation.

Brief overview of microarray technology

The first microarrays were produced in 1995 by Mark Schena and colleagues in Patrick Brown's lab at Stanford University. Schena used a robotic instrument to deposit a large number of spots of DNA of known sequences in a tiny grid on a glass slide [36]. Schena then took a digest of total cellular mRNA and with an enzyme produced DNA

sequences complementary to all the mRNA strands found in the digest. The DNA sequences produced from RNA in this manner are called complementary DNA (abbreviated as “cDNA”). The cDNA sequences Schena produced were also labeled, meaning a molecule that could be detected later with instrumentation was attached to the cDNA.²⁶ This labeled cDNA was then allowed to hybridize to any matching complementary nucleic acid sequences on the glass slide, and the excess was washed off. By quantifying the amount of labeled cDNA that remained on the slide with instrumentation that detects the label, Schena and his colleagues were able to provide a good, albeit indirect, measure of the amount of different types of mRNAs expressed in the cell. Thus was the beginning of the DNA microarray, in which hundreds or thousands of simultaneous measurements of mRNA expression levels could be performed.

Today, two major types of DNA microarrays are routinely used. One is based on Schena’s original idea, in which tiny spots of cDNA sequences are bound in a grid pattern on a glass or nylon membrane substrate. The other scheme is an array patented by Affymetrix, Inc., in which short sequences of DNA of different sequences, called oligonucleotides²⁷, are synthesized directly on a silicon chip substrate.

cDNA arrays

cDNA microarrays are fabricated by depositing tiny dots of cDNA on a glass slide or nylon membrane with a robotic mechanism. The robotic deposition is required

²⁶ Common labels used with DNA are molecules that emit visible light when illuminated with fluorescent light, or molecules containing a radioactive isotope.

²⁷ “oligonucleotides” means “few” nucleotides and generally denotes a single-strand of between around 4 and 30 nucleotides, although there is no set formal definition.

since the miniaturization of microarrays results from placing cDNA in a tightly-packed array with perhaps only several tens of nanometers of spacing between individual spots. The cDNA is typically produced by separate PCR amplifications²⁸ of many starter sequences—perhaps even thousands or tens of thousands of different sequences. The cDNA to be spotted is therefore usually double-stranded PCR product and must undergo a denaturing or melting step to produce single-stranded cDNA from the original double-stranded cDNA²⁹. This also minimizes self-hybridization such as “hairpin” loops. Once the double strands are separated, the single cDNA strands are then immobilized to the substrate through cross-linking induced by precisely timed exposure to heat or UV light.

The sample to be analyzed is normally isolated from a digest containing a mixture of all the cellular nucleic acid. To isolate only the expressed mRNA portion that would have been destined for translation into protein, the mRNA is often passed through a column or on in a well containing an immobilized surface of poly-T sequence. As described earlier, cellular mRNA has a polyadenylated tail, so such mRNA preferentially hybridizes to the poly-T sequence and is trapped on the surface. Once thus separated, the isolated mRNA can be rinsed of impurities and then later released in purified form [7].

After isolation, the mRNA is then typically transcribed into a complementary sequence of DNA, using a combination of an RNA-to-DNA polymerase enzyme (known as reverse transcriptase) and equimolar portions of the four necessary constituent

²⁸ PCR stands for *polymerase chain reaction*, and is a method for producing millions of copies of a particular DNA sequence from a single double stranded starting sequence containing around 100-500 nucleotide base pairs.

²⁹ Melting and denaturing means separating double-stranded DNA into two single strands and is essentially the opposite of hybridization.

nucleotides A,C,T,G (dATP, dCTP, dTTP, and dGTP)³⁰. Typically, one of the nucleotides is labeled by either a fluorescent tag, or through incorporation of a radioactive tracer such as dCTP in which the naturally occurring phosphate is replaced with a radioactive isotope such as P-33.³¹

The labeled cDNA molecules in the mixture are then allowed to hybridize to their corresponding sequences—if present—on the microarray. Any excess or unhybridized labeled cDNA is then rinsed away and the intensities of the label or tag measured by a photomultiplier tube (for fluorescent tag) or photographic film (for radioactive tag). The amount of labeled cDNA that matches the corresponding cDNA sequence on the microarray should, in principle, be proportional to the quantity of mRNA that had been present in the original cellular digest. If a particular mRNA having a corresponding sequence on the microarray had not been expressed in the cell at all, then its spot on the microarray should be essentially blank. Conversely, mRNA sequences that had been highly expressed should exhibit very intense signals. Intermediate expression levels of mRNA should produce intermediate levels of signal on the microarray accordingly. Thus, microarrays should produce fairly accurate quantitative data reflecting the expression levels of thousands of different mRNA sequences simultaneously.

While in principle this procedure sounds fine, in practice there are at least two serious potential sources of error that can fortunately be eliminated or minimized through making some adjustments to the above design.

³⁰ dATP means *deoxyadenine*, and serves as the starter molecule for the reaction. If one wants to talk generically about any of the four deoxynucleotides, then the traditional abbreviation is dNTP.

³¹ It is boldly assumed that the number of cytosines is the same for each cDNA sequence formed or at least that any difference would be negligible, but obviously the number of cytosines may very well vary appreciably from sequence to sequence.

First, hybridization is a chemical process that is dependent upon thermodynamic parameters such as salt concentrations and temperature.³² If one for instance lowers the temperature to make sure the matching sequences more completely hybridize, then one gets more mismatching sequences hybridizing as well. Conversely, if one raises the temperature to eliminate non-specific mismatch sequence hybridization, then one promotes incomplete hybridization of the perfectly matching sequences. Hybridization is therefore not an all-or-none process, but follows a sigmoidal association-dissociation curve. In other words, by setting salt and temperature conditions, one cannot absolutely ensure that all of the matching sequences will hybridize correctly while all the mismatch sequences will not. There will always be some matching sequences that will not hybridize correctly, as there will always be some mismatch sequences that wrongly hybridize. Nonetheless, overall the hybridization will be more or less correct, but the process is never quite perfect. By fine-tuning the thermodynamic parameters (known as setting *stringency* conditions), optimal hybridization can be achieved within acceptable bounds for most practical purposes. However, regardless of any parameter fine-tuning, non-specific hybridization still remains a potential source of error.

Second, and more importantly, there are many separate steps involved in doing a microarray experiment and each step can introduce appreciable variability. For instance, differences in total mRNA concentrations in the cellular digest can vary depending on the efficiency of the nucleic purification and mRNA isolation processes. The cellular digest

³² i.e., thermodynamic in terms of Gibbs free energy, otherwise known as the energy available to do work in a chemical reaction. Also, the rate of the reaction, another thermodynamic parameter, is governed by the Arrhenius equation: $k=A*\exp(-E_a/R*T)$, where k is the rate coefficient, A is a constant, E_a is the activation energy, R is the universal gas constant, and T is the temperature (in degrees Kelvin). This shows that the rate of a reaction increases (non-linearly) with temperature.

itself may consist of a starting sample from a tissue or cell culture containing different number of individual cells. Variations in the efficiency of conversion of mRNA into labeled cDNA is also a potential source of variability. One potential solution to this problem is to use a calibration standard of some sort with which to compare mRNA expression levels. However, with so many steps involved, there still may be appreciable error from microarray to microarray that needs to be taken into account.

One way to remove microarray-to-microarray error completely is to incorporate both the experimental sample and the control sample within the same microarray. Typically, the cDNA from, say the control tissue digest, is labeled with the fluorescent cyanine dye Cy3 and the experimental sample cDNA is labeled with a different fluorescent dye such as Cy5. The two labeled cDNA samples are allowed to hybridize simultaneously to the microarray cDNA and then the microarray is rinsed to wash away the excess. The microarray is read out by exciting the dyes with an He-Ne laser. The fluorescent Cy3 dye emits at 543nm (green) and the Cy5 dye emits at 633nm (red). The ratio of the two light intensities produces a measure of the relative level of mRNA expression between the experimental sample and the control. Thus, the microarray data is produced relative to an internal control, minimizing the effects of the sources of error inherent in microarray technology described earlier. Using this technique then, the microarray data is internally standardized to give relative mRNA expression values between the experimental sample and its corresponding control [36].

There are two major assumptions here that should be recognized, however. One concerns the fact that both the Cy3 and Cy5 labeled sequences must compete for hybridization sites. It is assumed that both types of labeled sequences always hybridize in

equal proportions—50% of the Cy3 variety and 50% Cy5 variety—on every site. This may not necessarily be the case, especially when the copy number of a particular sequence being detected is in the one-to-tens range. The other assumption is that the signal-to-noise ratio does not matter despite any error introduced by non-specific hybridization. In other words, consider the case in which a microarray might have very high levels of non-specific hybridization. Even if both Cy3 and Cy5 labeled sequences were responsible for producing this non-specific hybridization signal equally, the signal of interest will be swamped by this “noise” signal. It would be difficult to have much confidence in a small signal when appreciable non-hybridization signal is present, even if this non-hybridization signal is subtracted out.

Affymetrix arrays

Affymetrix GeneChips use a very different approach to that of cDNA chips. These chips use photolithography techniques similar to those developed in the semiconductor industry. Affymetrix uses quartz wafers which are chemically processed to produce a stable substrate for growing oligonucleotide chains. A thin film of photo-sensitive linker is then deposited on the wafer. By using a tiny mask perforated with holes in specific places, some spots are sensitized with UV light, while the masked portions are not. One of the four nucleotides can then be washed over the surface, but the nucleotide will only attach where the UV light had been allowed to sensitize the surface. Similarly, more photo-sensitive linker is added to the entire surface, binding to all the nucleotides already bound. On the next pass, a new mask is placed over the wafer having a different pattern. The wafer is exposed to UV light again, sensitizing the exposed areas. A

different nucleotide washed over the surface then attaches only to the sensitized spots, extending the anchored chains with only that nucleotide where appropriate. By repeating this process many times with all the nucleotides in turn, chains containing any combination of A, C, T, or G are grown in specific locations on the wafer. Affymetrix grows specified oligonucleotide sequences on quartz wafers in this manner up to 25 total bases long each. Using photolithography masking techniques, Affymetrix can also achieve a density of over 10,000 spots of a different oligonucleotide sequence each within a single square centimeter [1,36].

Within each Affymetrix chip are 20 (usually) replicates of the same oligonucleotide sequence along with 20 replicates of the oligonucleotide sequences having just one nucleotide in the center substituted with a different nucleotide. The first set of oligonucleotides are known as perfect match (PM) sequences, while the oligonucleotides differing by just one base are called mismatch (MM) sequences. The idea is that by comparing the signal one gets from the PM sequences against the MM sequences, one gets a resultant value that corrects for any degree of non-specific hybridization. In other words, by subtracting out the MM signal the resultant signal should be a truer indication of the pure specific hybridization signal. The 20 pairs of replicates are included to improve accuracy statistically through the process of averaging [1]. The algorithm for calculating the signal is therefore:

$$\sum_{i=1}^n \left(\frac{PM_i - MM_i}{n} \right)$$

If the difference between PM and MM is negative, then the level of mismatch hybridization is greater than the level of perfect match hybridization. This is biochemically implausible, so such a score is excluded from the average.

VI. Experimental manipulation of gene expression

In Chapter 4 we mentioned two strategies for inferring causal information from data: passive data collection, or *observation*, and variable manipulation, or *intervention*. If we wish to use the intervention strategy in gaining causal information about gene expression, we would ideally hold all variables constant except one, and then fix that single variable at value of our choosing. Biologists tend to design experiments with an intuitive understanding of these considerations in mind, but we will identify and discuss them explicitly in this chapter.

Since gene expression levels change in response to many different physiological and environmental conditions, we will identify those conditions we need to set to hold all gene expression variables constant (within practical limits) during our biological experiments. Then, we will discuss ways to “wiggle” or intervene on one variable at a time in a gene expression experiment. These methods include knockout, knockin, gene underexpression, and gene overexpression.

Physiological and environmental considerations

When designing gene expression experiments in actual biological systems (as opposed to the purely abstract), there are overall physiological and environmental conditions affecting gene expression levels that need to be considered:

- 1) Cell cycle—prokaryotic cells divide by simple binary fission, while eukaryotic cells typically go through a mitotic cell cycle in which chromosomes are

replicated and allocated to each daughter cell in an orderly, systematic fashion (even though it is a eukaryote, yeast can also propagate through budding, which is similar to binary fission). Each of these processes is regulated by gene expression products. Based on traditional molecular biology experiments, Spellman et al (1991) found that of the 6200 genes in yeast (*S. cerevisiae*), 800 genes' expression levels varied in relation to cell cycle [38].

It is not entirely clear, however, whether we can consider the cell cycle to be a consequence of, or cause of, gene regulation, or whether there is a *cyclic* causal structure at work (i.e., gene autoregulation or multi-gene feedback loop). Our use of DAGs is not amenable to looking at such phenomena, and so for practical purposes we would want to be able to avoid changes in gene expression related to cell cycle activity.

We can therefore hold cell cycle gene expression activity constant by holding cell cycle in *stasis*. This can be accomplished in cell cultures by the addition of cell cycle arresting compounds such as . However, arresting the cell cycle obviously cannot be done in whole, multicellular organisms.

An alternative to cell stasis in cultures of yeast is to synchronize the cell cycle [38]. There are several ways to accomplish this. One is to simply sort cells on the basis of size and absence of budding (an indicator of maturity), and retain only those cells within a defined size limit. Another method is to add a pheromone, or cell signaling substance, known as an alpha-factor to arrest the cell cycle. Finally, one can use a strain of yeast in which contains a mutation of the gene CDC-15. CDC-15 codes for a protein needed for exit from mitosis. When raised to a high temperature (37 degrees C), strains of yeast containing a mutated form of CDC-15 do not express that protein. The cells therefore go

into a stage of *cell cycle arrest*. Subsequently reducing the temperature restarts the cell cycle, effectively synchronizing all the cells' cycle.

A serious problem with cell cycle synchronization is that, upon release from cell cycle arrest, individual cells' cycle rates will inevitably begin to diverge. If we were to plot the position in cell cycle for each individual, we would therefore initially see a graph with a very sharp spike signifying that all the cells were in precise lock-step. However, over time as the cells' rates of going through their cycle begin to diverge, we would see the distribution widen. This introduces a variability that is itself a function of time, adding undesired complexity to our analysis.

2) Maintenance, or “house-keeping” genes—these genes are putatively responsible for the production of structural and metabolic proteins that are required to be in constant supply, such as actin or ribosomal proteins. Their concentrations should (in principle) not vary appreciably with respect to any other variable in any particular tissue from a particular organism, and thus expression activity for these genes may already be assumed to be at some steady-state, positive value.

3) Development/differentiation—eukaryotic cells in multicellular organisms typically differentiate and specialize during development. This is a direct consequence of the relative expression activity of certain genes. Once a cell differentiates into a particular cell type, it permanently expresses a specific complement of genes. For instance, all other factors being equal, we can expect a different nearly-permanent pattern of gene expression in leukocytes versus neurons. In practice, in biological experiments we can

hold this variable constant by using a single cell culture line that does not actively differentiate during the course of the experiment.

4) External environmental factors—all cells respond to environmental changes, such as the presence or absence of certain nutrients, drugs, or toxicants, via alterations in gene expression. Additionally, temperature, light, and ionizing radiation can affect the expression of certain genes (e.g., heat shock, DNA repair) [3,7]. In practice, we can hold this variable constant by maintaining cell cultures in constant environmental conditions.

5) Inter-cellular signaling—both prokaryotes and eukaryotes respond to chemical signals given off by other cells. Prokaryotes respond to signals from neighboring cells as well as other cells within its culture/suspension, while eukaryotes in multicellular organisms respond to hormones, peptides, growth factors and other substances for which it has cell surface receptors [3,7]. In practice, we can hold this variable constant by maintaining a cell culture in stasis, but it is an open question whenever more than one cell is present whether each is affecting another through inter-cellular chemical signaling in some fashion.

6) Normal versus transformed cells—transformed cells, derived from tumor cells and often described as “cancerous,” appear to be the result of a fundamental change in gene expression patterns not accounted for by the other factors listed above. Unlike normal cells that divide until some limit of senescence or programmed cell death (known as apoptosis), transformed cells grow and divide indefinitely as long as they are given the

requisite nutrients. Some transformed cell strains commonly used in research have been actively undergoing innumerable cell divisions for many decades [26]. An example is the HeLa cell strain derived from the papillomavirus-induced fatal cervical carcinoma of a patient named Henrietta Lacks in 1951. This variable, normal versus immortalized cell lines, is considered when choosing a strain on which to conduct one's experiments.

Gene expression manipulation

Since the time of Mendel, traditional genetic studies relied on the basic idea of crossing parents that possessed certain traits and then observing the frequency of those traits in offspring. Since the advent of true molecular biology in the 1970s and 80s, researchers began manipulating the expression of genes directly. The first type of gene manipulation consisted of simply disrupting a particular gene in an organism, and then observing the effect on the phenotype. The putative function of the gene was therefore inferred from whatever characteristics seen in the phenotype that differed from the unaltered, or "wild-type" organism. Gene disruption experiments such as this were subsequently described in terms of "knocking out" a gene, so the technique was termed "gene knockout" or alternatively, "gene targeting."

Gene manipulation experiments can be performed a number of different ways. Unlike in other disciplines in which researchers may practice only some small number of canonical techniques, in molecular biology researchers often devise their own unique, *ad hoc* gene manipulation methods to address the particular question they are trying to answer. Further, most gene manipulation experiments are aimed at understanding one particular set of genes or a single metabolic pathway in a particular organism, in contrast

to the aim discussed here of understanding the *general* processes involved in gene regulation. Therefore there is great variation in the actual techniques used to manipulate genes—far too numerous and complex to attempt to describe here. However, we can discuss some general strategies that are representative of the types of methods used in gene manipulation studies.

With this in mind, then, we can attempt to put gene manipulation studies into a rough conceptual framework. First, we can categorize techniques for performing gene manipulation at the cellular level into three very general approaches. These consist of techniques that:

- 1) introduce new genetic material (such as genes or promoter sequences),
- 2) introduce substances into the cell that interfere with gene expression, and
- 3) alter the genetic code, that is, introduce mutations by radiation or chemicals.

It might also be helpful to consider the types of biological systems being studied in gene expression experiments to belong to one of three general categories:

- 1) organisms that exist naturally in single-celled form (e.g., bacteria and yeast),
- 2) cultured cells taken from multicellular organisms (e.g., mouse fibroblasts), and
- 3) whole multicellular organisms (e.g., mouse).

Knockout and knockin experiments

In single celled organisms such as *E. coli* and *S. cerevisiae* (yeast) as well as cultured cells such as mouse fibroblasts, new genetic material is often introduced using a carrier or *vector*, such as a virus known to infect the host organism.³³ Viruses that infect bacteria are called *bacteriophages* (or often simply called *phages*). To introduce new genetic material into an organism then, one synthesizes or purifies a desired DNA sequence and then links to it flanking sequences on both ends similar to genes already present in the host organism. This sequence is placed in the virus or phage and introduced to a culture of host cells. The virus or phage infects the cells, injecting and substituting its DNA for that of its hosts. In this way, the virus or phage inserts the desired foreign DNA, replacing the original gene(s) [3,7,26].

A common refinement to this procedure is to add an additional gene that gives the host organism resistance to some particular antibiotic. Once the virus or phage is introduced to the host cells, the antibiotic is added to the culture. Any cells that did not take up the desired foreign DNA successfully then die, leaving a pure culture of cells containing the manipulated gene(s). Often, in addition to this step researchers will also take a sample from the cell culture and sequence the DNA, additionally confirming the successful replacement of the original gene with the foreign DNA.

To conduct a knockout experiment using this strategy then, the researcher will substitute some non-functional DNA sequence for the original functioning gene in the host cells. Once transformed, the hosts' original gene will be effectively "knocked out."

³³ Common viral vectors include adenoviruses (commonly found in human adenoids and sometimes cause sore throats) and SV40 (simian virus 40).

To add one or more genes, known as a “knockin,” the researcher simply uses one or more fully-functional genes instead.

An alternative to targeting or “knocking out” genes in this manner is to use a substance that interferes with gene expression. One option might be to use a protein that binds to the particular nucleic acid sequence found in the gene of interest that completely blocks the RNA polymerase from transcribing it—essentially using the same idea of a repressor protein found naturally in gene regulation in prokaryotes. The problem here is to find a protein that will be taken up by a cell without being either toxic or degraded by the cell, which cannot always be done.

In multicellular organisms such as mice, the procedure for knocking out a gene through DNA substitution is substantially more involved than that used in single-celled organisms but begins using a similar idea [26]. The type of cell used for introduction of the foreign DNA is called an embryonic stem cell (ES). ES cells are useful because they are totipotent, meaning that they have not yet differentiated into any particular cell type. The ES cells are given the foreign DNA in a method similar to that described above for single cells, and are then injected into a viable embryo. The transformed ES cell’s genetic material combines with that of the embryo, forming a *chimera*. This chimeric embryo is then implanted into a female organism that has been manipulated with sex hormones so that it will accept the artificial pregnancy. The pregnant female then completes normal gestation and gives birth to offspring. Being chimeric, however, the offspring will have one set of manipulated genes and one set of normal genes. Therefore, two chimeric animals are mated, giving rise to 1/4 proportion (on average) of the subsequent offspring

having two sets of the manipulated genome [26]. Such an organism is said to be *transgenic*.³⁴

Often, however, a gene knockout performed this way will prove to be fatal for the organism. Sometimes researchers can gain insight into the function of the gene they are studying by using the offspring containing the one set of normal genes and the one set of manipulated genes, known as a *heterozygous* animal.

Another method for manipulating genes similar to performing knockout experiments is to introduce mutations in a population of individuals. After random mutations are induced in the population, the individuals or their progeny are monitored for the effects of that mutation. The function of the mutated gene is therefore backwardly inferred from its observed effects. This is in stark contrast to the methods discussed previously in which a known gene is manipulated and the subsequent effects observed.

Such a strategy is commonly used in *C. elegans* (flatworms). A population of worms is exposed to a chemical mutagen such as psoralen³⁵ and irradiated with ultraviolet radiation. The mutagens randomly induce mutations in the worms' genomes. The worms are then separated and allowed to multiply into individual subpopulations (if they live). A sample is taken from each subpopulation and tested via sequencing or PCR for the presence of mutations. If a mutation is found, the gene the mutation occurs in is noted and the mutation it is traced back to the originating subpopulation. This subpopulation is then observed for variations in phenotype that might be caused by the

³⁴ As might have already occurred to the reader, the generation of transgenic animals is extremely complex and involved and could take years to perfect; whole laboratories and careers, and indeed entire industries have arisen in response to the demand for transgenic organisms.

³⁵ When activated with UV light, psoralen binds tightly to DNA and cross-links it. Hence it is a powerful mutagen in addition to being systemically toxic. Psoralen is found in celery and is thought to help protect it from insects.

mutation [40]. Clearly, since mutations are rare events, and non-fatal mutations are more rare still, this method relies on having large, easily reproducing populations of organisms.

Gene underexpression and overexpression

Gene underexpression experiments are uncommon, but in a sense a gene underexpression experiment might also be considered to be an incomplete knockout experiment. One example of this is the use of a form of RNA that binds to the DNA sequence in the gene of interest. This is known as small interfering RNA, or siRNA. The siRNA is an RNA oligonucleotide of a desired sequence that binds with the double-stranded DNA found in the gene of interest and forms a tightly-bound triplex. This triplex blocks the RNA polymerase from transcribing the gene. Brown et. al. report an up to 50% reduction in both mRNA transcripts and protein expression using this method [6]. This method can be used in cell cultures as well as whole multicellular organisms.

Gene overexpression experiments are performed through two general strategies. One is to simply insert many copy numbers of the same gene through the method described for gene knockin experiments. Another strategy is to add one or more promoter sequences to an existing gene in a similar manner. This promoter sequence(s) will then increase the transcription rate of the relevant gene. A variation on this strategy is to use a drug-inducible promoter or transcription factor. In this way, transcription rates in the manipulated organism can be further controlled at the experimental level by withholding or administering the drug.

Many variations on all the above procedures exist, however. For instance, in addition to knockout, knockin, or promoter insertions, genes may be inserted that code

for a protein that acts as a dye or light-emitting substance. In this way, the site of gene expression or protein accumulation can be located by gross observation or microscopy.

Most of these methods have been designed and developed for a traditional biological research strategy: alter gene function, observe the phenotype, then infer the function of the altered gene. As outlined, our aim is to alter gene function one gene at a time and measure the subsequent changes in mRNA expression to infer the causal network. Gene manipulation technology may not allow us to perform a complete and exhaustive intervention on all genes and all combinations of genes, however. For instance, any time one or more genes are altered the organism may die as a result, preventing the measurement of mRNA levels. Further, the effort involved in producing viable transgenic animals limits the number of genetic manipulations realistically possible—most such manipulations are aimed at some specific metabolic pathway or phenotypic outcome of separate interest to biomedical researchers, not to a systematic series of consecutive knockouts aimed at inferring in global gene regulatory networks. We therefore have to be content with a somewhat spotty and incomplete set of possible gene manipulation studies in any particular organism.

VII. Search representations and strategies

At this point we have discussed all the requisite components for using machine learning to infer the causal network involved in gene regulation:

- 1) gene expression data from microarrays,
- 2) methods of manipulating gene expression if necessary to get variable intervention data (versus passive observation), and
- 3) a mathematical method of representing the relationships between the data in terms of causal graphs (DAGs) that lends itself to computational evaluation.

The task from this point on is then to find a procedure for inferring the correct causal graph from the collected data. The method we will discuss here is based on the idea of a *Bayes net* [12,120,33,39], or a mathematical representation and implementation of a DAG based on data, with each variable having a probability distribution associated with it.

We can also differentiate between procedures that are appropriate for intervention studies and those that are appropriate for observational studies, or those that might benefit from some combination of both. We will discuss some representative strategies from among the many that have been proposed for inferring causal structure from data.

Experimental data

One way to attempt to establish causation is of course familiar to most scientists: perform an experiment. In the simplest form of experiment we attempt to hold all variables constant except the one we are investigating, and then compare the effects of the single manipulated variable with the unmanipulated version of the variable. While

intuitively scientists understand the objective of an experiment is gaining “knowledge,” it isn’t always explicitly acknowledged that the knowledge sought is in the form of *causal* information [12].

How then are we extracting causal information from an experiment? Whenever we perform an experiment, we implicitly have *background* information in the form of a time-ordering of events (in studies involving passive observation only we may not have this extra information). Whenever we manipulate a variable and then observe the effect of that manipulation, we know that the intervention preceded the observed effect, and so we can infer whether manipulation of that variable caused a specific event. The fact that we know which happened first (our intervention or the observed effect) gives us causal information.

Once we have this background information, how do we determine whether an effect has actually occurred, or whether the observed effect might be due to random processes or “noise”? The most common method is *hypothesis testing*. The null hypothesis usually states that there is no difference between the effects of intervention and non-intervention, while the alternative hypothesis states that there is a difference. We collect data from the effects of the manipulated variable and the unmanipulated variable, and do a statistical test to determine the likelihood of observing those effects purely as a result of chance to see if we are justified in rejecting the null hypothesis in favor of the alternative hypothesis. Typically we must assume our data comes from a particular probability distribution as well as assume a cutoff (called alpha, and often assigned a value of 0.05). If analysis of the data shows that our test statistic falls outside our cutoff, we say that the results are significant, and that they support rejecting the null hypothesis

in favor of the alternative hypothesis.³⁶ The reported p-value tells us just how far we could have set our cutoff (alpha) and maintained significance. Some people interpret this as a degree of significance (although others argue there is no degree of significance: either there is a significant difference or there is not).

Another common approach for detecting the effect of a manipulated variable is to do a *regression* analysis. In its simplest form, one has a series of data from various degrees of manipulating our variable, as well as data from the observed effects of that manipulation. One then attempts to look for a *correlation* between the two data sets, i.e., we attempt to fit a line to the data by some measure that minimizes the distance between each data point and the resultant line.³⁷ The simplest form of correlation is represented on a graph by a line. The slope of the line gives us an indication of how much the manipulated variable appears to affect the observed variable. If there is no association between the two variables, we would expect the slope of the regression line to be zero.³⁸ Again, we are forced to turn to hypothesis testing as described above to decide on some numerical cutoff in case the slope of our regression line is not exactly zero, but a low value near zero.³⁹

To gain causal information from an experiment then, we must: 1) be in the position to perform variable manipulation in an environment in which we can hold all other variables constant (usually called a laboratory), 2) utilize our background

³⁶ Some argue that we are never justified in *accepting* the alternative hypothesis, only in rejecting the null hypothesis.

³⁷ There are a number of ways to do this, but the most common method is minimizing the square of the vertical distance between the data points and the regression line.

³⁸ However, the converse is not true: if the slope of the regression line is zero, this does not necessarily mean that there is no association between the two variables.

³⁹ There are of course many variants of these procedures as well as numerous statistical tests, but these are meant simply to serve to illustrate the concept between data interpretation and extracting causal information.

information concerning the time course of our intervention and the observed effect, and 3) have a method (usually statistical in nature) for determining whether an effect had indeed occurred rather than having observed the data as a result of some random occurrence.

Causation versus association in passive observation

At the turn of the 20th century, there was a long-standing debate between statisticians. Some believed one could deduce causal relationships from the associations derived from a statistical analysis of observed data, whereas others vehemently rejected this idea. Just because two variables might be correlated (in the absence of time-course background knowledge), they argued, it does not mean one can correctly infer that one variable is the cause of the other. One possibility is that there might be a hidden or unrecognized cause of both that produces the correlation—that which we now represent through a latent variable.

For over a century the latter view had been more or less established. However, in the 1990s [39] it had been shown that when a sufficient number of variables are involved, causal information can sometimes be correctly estimated by examining the appropriate conditional independence relationships between variables.⁴⁰ Whereas it was previously thought that the only way one could get causal information was through variable manipulation, it had now been shown that some causal information can be obtained solely through the proper analysis of observational data when more than two variables are

⁴⁰ Assuming certain conditions such as the absence of causal cycles, the Causal Markov Condition, and the Faithfulness Condition as explained earlier in Chapter 4.

involved. For instance, as previously discussed in Chapter 4, if two variables are independent, but show a dependence when conditioned on a third variable, then one can reliably infer the existence of a collider, i.e., the third variable. This means that in a DAG, the causal arrows or arcs go from each of the first two variables into the third, and thus causal information is established. The first phenomena represented by the first two variables each have a causal relationship with the third (in the absence of latent variables).

When dealing with a large number of variables, however, the computational resources needed to check all the conditional independence relations becomes considerable if we resort to the “brute force” approach of checking independence between every possible combinations of variables. If we have n variables, then in any graphical model there are $\frac{n(n-1)}{2}$ possible adjacencies between variables. This means that there $2^{\frac{n(n-1)}{2}}$ possible graphical structures (the factor of 2 comes into it since we can either draw an arc between the adjacencies or not). When we consider directed graphs, we get $3^{\frac{n(n-1)}{2}}$. To do a search on this number of possible graphs from 30 variables, we would have to check 3×10^{435} different graphs [39]! If we consider hundreds or thousands of gene interactions the number of possible DAGs would be far beyond astronomical. Based on computational constraints, then, insisting on an exact answer would be impractical if not impossible.⁴¹

⁴¹ The PC algorithm was designed exactly to avoid this problem when the true graph is sparse, as is surely the case in gene regulation. The discussion here however is to give the reader an appreciation, in very general terms, of the computational issues involved in graphical model search when there are many variables involved.

Further, we might wish to consider the fact that our data is generated by experiment, and therefore is likely to have some sort of error associated with it. Anytime we deal with experimental data we must resort to some kind of hypothesis testing (in which case we have to decide on a choice of numerical cutoff for claims of significance), and therefore all our conclusions come with some sort of measure of how likely it was to have obtained our data purely by chance. For the above two reasons, finding a DAG consistent with the data then is not an exact science—we must be satisfied getting an answer that is nearly right, but maybe not exactly right. We can therefore think of all the total possible DAGs as a nearly infinite space through which we are conducting a search for the best DAG based on our data.

We must realize that we may not be able to get a single, full DAG out of our search. Perhaps the best we can do in some instances is to show that there is an arc between variables, but not necessarily which direction it points. If we ignore directionality of the arcs, then, it is possible that several or many different DAGs would have the same basic structure in terms of possessing arcs between all the variables. We call the collection of possible DAGs an *equivalence class*. When doing searches for the correct DAG then, we must often be satisfied with identifying the proper equivalence class instead of the actual single DAG [8].

Different strategies have been devised to more efficiently search for the correct (or more likely to be correct) DAG that is consistent with observed data. The two broad categories of strategies are called *scoring* searches, and *constraint-based* searches [12,14,39].

Scoring searches

A scoring search looks at how entire graphical models fit the observed data as a whole. Each model is assigned a score based on the likelihood that data can be explained by that model along with a penalty for model complexity. We can think of the likelihood as the probability of observing the data given a particular model (graph) and parameterization (θ):

$$\text{likelihood} = p(\text{data} \mid \text{model}, \theta)$$

The maximum likelihood estimator is therefore the value maximized by choice of value θ . By increasing model complexity (e.g., increasing the number of edges in the graph), it is often easier to fit data. Therefore, scoring procedures contain a term that penalizes the score for higher complexity, i.e., there is a subtraction from the score for each additional edge added to the model.⁴² The score then consists of the likelihood the model fits the data along with a penalty for complexity:

$$\text{score} = p(\text{data} \mid \text{model}, \theta) - \text{complexity measure}$$

Examples of this sort of scoring method are the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC):

$$\text{AIC} = -\left(2 \log(\hat{L}) - 2K\right)$$

$$\text{BIC} = -\left(2 \log(\hat{L}) - \log(N)K\right)$$

where \hat{L} is the maximum likelihood estimator, K is the number of edges in the graph, and N is the number of data points.

⁴² Parsimony of model is a virtue and is essentially an application of Occam's Razor.

The difference between these two scoring methods is based on the choice of complexity penalty. The AIC is used in situations where one wants to be assured that once the highest scoring model is found, then subsequent predictions about the data from new data will more likely be correct. The BIC, on the other hand, is used in situations where one believes that there is one “true” model, and it is of no interest whether that model correctly predicts future data.

However, no matter which scoring procedure is used, the search begins through (for all practical purposes) a virtually infinite search space as described previously. If we just begin searching and then find a very high scoring model followed by a series of lesser scoring models, how would we know if we have already found the highest scoring model, or if there is a yet higher scoring model still to be found? Further, how long must we continue searching for a better model? Since the sheer number of graphs is so large, we cannot hope to score every single model individually, so we must come up with some sort of way to help us estimate whether we have identified a *global maximum* or merely a *local maximum* in our score. The drawback of scoring searches then, is that they are susceptible to finding local maxima; it is difficult to know whether one has actually found the true global maximum one desires.

Many methods have been proposed for avoiding being “stuck” in a local maximum. One common strategy is to see the search on some manageable number of models chosen randomly that cover (or it is hoped) the entire search space in a representative fashion. Other methods have been proposed that try to optimize models based on a form of selection akin to natural selection involving random mutation, recombination, and selection of “offspring” based on “fitness” in terms of a score—these

sorts of methods are called *genetic algorithms*. Whatever the case, in models containing many variables we must be content with possibly obtaining the correct equivalence class or perhaps just some approximation to the true causal graph. Devising methods for more accurate scoring searches than can be done efficiently in a reasonable time is currently a very active area of research. [12,39]

Constraint-based searches

Another overall strategy for searching for causal models is to evaluate the data on the basis of conditional independence of variables as described in Chapter 4 previously. Instead of scoring the entire model simultaneously, this method relies on looking at the independence relations of each variable as well as the conditional independence between each variables conditioned on every possible subset of other variables. Again, there are many ways to go about checking these independencies, and some methods are more efficient than others. For instance, a method may first look at the independence relations between each pair of variables, potentially narrowing down the search space considerably on the first pass. Other methods are similarly aimed at optimizing the order of conditional independence tests between variables in an attempt to more quickly pare down the search space in the interest of overall speed and efficiency. In general terms, optimization can be achieved by eliminating redundant testing, or more quickly reducing the size of the search space, or both. Examples of searches that use these sorts of optimization strategies are the PC algorithm (described previously and stated in Appendix A) and the Fast Causal Inference (FCI) algorithm [12,39].

The disadvantage of constraint-based searches is that early mistakes impose significant penalties on the computational efficiency of the process. That is, if an edge or arc is erroneously eliminated due to a questionable degree of dependence between the connected variables, then the entire model selection will go down the wrong path from that point on. Once the mistake is found, the search would have to “back up” a considerable number of steps and start over. This difficulty in dealing with early mistakes undermines the very efficiency the approach was intended to produce.

Hybrid approaches

While scoring-based searches and constraint-based searches are each active areas or research in their own rights, a particularly fruitful approach appears to be hybrid methods based on some combination of both. For instance, it is easily seen that one might first eliminate many models using the first pass of a constraint-based approach, and then use a scoring approach on this smaller search space. The idea is to capitalize on the virtues of each method while minimizing the disadvantages of each—i.e., producing a quick but comparatively accurate algorithm for performing a causal graph search.

Clustering/PCA

While not officially recognized as a causal graph search method, another common family of statistical methods bear some similarity to causal graph searches in that they aim to identify associations between variables. Much microarray data is evaluated using clustering methods or Principal Component Analysis (PCA) [17,20]. The idea behind

these methods is to group variables on some measure of similarity. Clustering is therefore often used on microarray data to help identify similarly expressed genes. PCA is used to identify gene expression measurements that exhibit similar degrees of variability. Such methods do not provide useful causal information, but rather they are more commonly used as exploratory tools for identifying markers or patterns of gene expression that appear to have similar function or genes that are co-expressed under particular biological conditions such as a disease state [19]. Clustering can also help identify genes that may be co-expressed as a result of being in a similar position on a chromosome (i.e., regions of histone exposure to transcription factors)[9].

VIII. Evaluating a search method based on actual data

Model discovery versus model validation

In the context of gene expression data, how might we go about searching for the most correct representation of a causal graph? In biology, we already have considerable background information about which genes regulate which other genes in certain organisms, although most often this consists of just partial networks. Researchers have therefore typically tried to evaluate gene regulatory networks through several ways using this background information.

One is that we may have an idea of what the structure of the true gene regulatory network should look like based on traditional molecular biology data. In this instance we should have an idea, at least within the context of a particular metabolic pathway, of the genes involved in regulating the proteins expressed that are involved in that particular pathway. Once we decide on our causal model, we would then simulate gene expression data, run that simulation through our model, and then see if the resultant output data recovers our initial network when evaluated by a search method. How do we quantify the results? We would perform a goodness-of-fit test on the parameters in our model, i.e., perform a scoring measurement (as described above) to see how good the model fits the data.

Another evaluation method is based on performing a causal graph search on actual data (such as microarray data) and then identifying a causal network from that search. Another set of actual data is then inputted into the model, and one evaluates the model based upon how closely the output resembles actual gene expression (output) data.

In this instance the first set of data would constitute a *training* set, while the second set of data a *validation* set.

Finally, many biologists do not use gene expression data as the output of their model, but rather view gene regulation in terms of developmental biology. In such an instance, biologists would test their gene regulatory network models by predicting a particular phenotype that they hypothesize will result from a specified regulatory network. In this instance it is difficult to quantify “degrees of similarity” between the ideal and actual phenotype, so we must regard this method of evaluating a regulatory model as being more qualitative than quantitative.

Other considerations

So far we have been concerned with how closely a given model fits the data (or vice versa). However, there are other considerations to be taken into account as well. One consideration, mentioned briefly already, is whether a model can deal effectively with latent variables. If a model is forced to ignore them, then the model may not fit the data as well as it could. If the model has the latitude to add latent variables as necessary to improve the score, then the model may fit the data much better in addition to identifying latent variables that might warrant further empirical study.

Another consideration is how a model deals with *noisy* data. Variation or noise can be introduced either as a fundamental aspect of the phenomena under study (i.e., stochastic processes involved in chemical binding), or through the measurement process itself (i.e., measuring the mRNA expression in an aggregate of cells). While in some instances it may be possible to minimize experimental variation through refining

experimental technique, it may not be possible to remove all variation completely.

Therefore it is advantageous to use a method of modeling gene regulatory networks that can tolerate some variation or noise in the data.

Another consideration is the degree of complexity of the transmission functions involved between the variables. In the simplest case we have a *boolean* network, in which each variable can have only one of two values: “on” or “off.” [43] Obviously this fails to capture the quantitative aspect of both gene expression and regulation completely, and arguably the qualitative nature as well. Further, this approach fails to model gene expression signal amplification and thresholds, which may be important features of regulatory networks. As a first approximation a boolean network may be easier to implement; however, the resultant causal graph may be much less informative than we desire. At the other end of the extreme are models which use ordinary differential equations [27,23] or stochastic Petri nets [22,34] to model the chemical kinetics (i.e., reaction rates, equilibrium constants, etc.) and time course involved in gene regulation in great detail. While ultimately this is exactly the type of information we would like to have, the drawback is that with microarray data, the sort of detailed input data necessary is completely lacking (unless one wants to perform a series of many microarray measurements in very quick succession as well as measure all the protein intermediates’ levels—this would be worthwhile but difficult as well as extremely expensive to implement). A third alternative, one that lies between these two extremes, is the Bayes net approach described above [12,20,33,39]. It is sufficiently quantitative to provide interesting and meaningful picture of gene regulatory networks, while its level of detail does not outpace the degree of detail provided in the input data. Further, since the Bayes

net approach is based on probabilistic variables to begin with, it is naturally amenable to modeling noisy systems. Finally, some implementations of a Bayes net can address latent variables (e.g., FCI algorithm) [12,39].

“Gold Standards”

If we wish to evaluate the performance of a Bayes net on the basis of actual experimental data, we have one of two options. One is to infer a causal network, or at least a partial causal network, from data obtained independently through standard molecular biology experiments. The other is to perform a search for a DAG on actual microarray data. As of yet, the entire gene regulatory network for an organism is not known—this is indeed the purpose of the whole enterprise described in this thesis. We do however have partial networks and what is believed to be whole networks from the partial genome of some organisms.

Why are these “gold standards” important? Most algorithms that have been proposed have not even been tested on actual biological data. Those that have been tested have been done only on simulated data. Second, no one knows what an actual global gene regulatory network looks like, and few people know what a partial gene regulatory network looks like.

Why hasn’t anyone assembled these “gold standards” before? In researching these networks I had found a great deal of interest among many biologists who wished to obtain these standards once they were assembled. However, the information necessary to reconstruct these regulatory networks is dispersed widely, and in many formats. Also, these regulatory networks resulted from the work of researchers who were not aiming at

elucidating global regulatory networks, but rather particular regulatory networks involved in certain metabolic or developmental pathways. Finally, the gene regulatory relationships I found had various degrees of experimental justification.

In assembling these networks I used several criteria. First, the information had to be independently obtained through traditional “bench” experiments, not inferred from microarray data or through the use of machine learning tools. Similarly, the information had to be traceable to published literature. The networks also had to form “closed” regulatory networks (as best as can be determined). Finally, transmission functions between genes were sought (in this case, only up- or down-regulation relationships were available).

Here I will present partial causal networks from several organisms, the prokaryote *Escherichia Coli* and the eukaryotes *Saccharomyces cerevisiae* (yeast), *Strongylocentrotus purpuratus* (sea urchin), and *Melanogaster drosophila* (fruit fly). We may regard these partial networks to be the current “gold standards” against which any machine learning algorithm used to infer gene regulatory networks may be compared and evaluated.

A. *E. coli*—first row is the name of the regulator gene, those listed below are regulated by the gene in the top row (Thieffry et. al.)[41]. Annotation in parentheses denote whether gene is upregulated (+) or downregulated (-), when known. The exact transmission functions are not known.

Ada	AraC	ArgR	AsnC	BetI	BioB	CRP	CynR	CysB	CytR	DeoR	DnaA
Ada(*)	AraC(-)	ArgR(-)	AsnC(-)	BetI(-)	BioB(-)	AraC(+) CytR(+) Fur(+) GalS(+) MalT(+) PapB(+) PutA(+) CRP(*)	CynR(-)	CysB(-)	CytR(-)	DeoR(-)	DnaA(-)
DsdC	FanA	FanB	FIS	FNR	Fur	GalS	GatR	GcvA	GutM	GutR	Hns
DsdC(-)		FanA(+)	FIS(-)	NarL	Fur(-)	GalS(-)	GatR(-)	GcvA(-)	GutM(+)	GutM(-) GutR(-)	Hns(-)
IHF	IleR	IlyY	LexA	Lrp	LysR	MaiI	MalT	MarR	MetJ	MetR	PtlR
IHF(-)	IleR(-)	IlyY(-)	LexA(-)	PapB(+) Lrp(-) FanA(?)	LysR(-)	MaiI(-)		MarR(-)	MetJ(-)	MetR(-)	PtlR(-)
NarL	NR(I)	OxyR	PapB	PdhR	PhoB	PitC(-)	PurR	PutA	RafR	RhaR	RhaS
	NR(I)(*)	OxyR(-)	PapB(-)	PdhR(-)	PhoB(+)	PitC	PurR(-)	PutA(-)	RafR(-)	RhaR(+) RafR(+)	
SoxR	SoxS	TdcA	TdcR	TrpR(-)	TyrR	WrbA					
SoxR(+)			TdcA(+)	TrpR	TyrR(-)	TrpR(-)					

B. S. cerevisiae—first row is the name of the regulator gene, those listed below are regulated by the gene in the top row (Cherry et. al.)[11]. The exact transmission functions are unknown.

13nt_repeat	ABF1,BAF1	ACE1	ACE2	ADR1	AP-1	ARC	ATF	BAS1,PHO2	BAS2
SIN3	CDC19,PYK1 RPL2A HIS7 CHA1 PGK1 RPO21 LPD1 YPT1 ADE5,7 SPR3 COX6 ENO2 POT1 QCR8 ABF1,BAF1 FAS1 MSS51 QR15 RPO31 RPC40	CUP1	CTS1	CTA1 ADH2	TRX2	ARG1 ARG8	LPD1	PHO5 HIS7 HIS4 HO ADE5,7 ADE2	HIS4 ADE5,7
BUF	CCBF,SCB,SWI6	CEN	CPF1	CSRE	CUP2	CuRE,MAC1	DAL82	GA-BF	GAL4
HO CAR1	HO CLN1 CLN2	GAL2	PGK1 CYT1 MET16	ICL1 PCK1 FBP1 MLS1	CUP1 SOD1	FRE1 CTR1	DAL4 DAL7	URA3	GAL7 GAL10 GAL1 GAL2 GAL80 GCY1
GATA	GCFAR	GCN4	GCN4,GCRE	GCR1	GCR1,CTBOX	GCRE,GCN4	GFI,TAF	GFI	GLN3
UGA4 DAL3 GZF3,DEH1 DAL80 GAP1	OLE1 ERG3	HIS7 HIS4 TRP4 ILV1 ILV2 ADE4 ARG1 ARG8 HIS3	HIS7 HIS4 TRP4 ILV1 ILV2 ADE4 ARG1 ARG8 HIS3	CDC19,PYK1 PGK1 TPI1 ENO1 ENO2 ADH1	CDC19,PYK1 PGK1 TPI1 ENO1 ENO2 ADH1	HIS7 HIS4 TRP4 ILV1 ILV2 ADE4 ARG1 ARG8 HIS3	CIII-subVIII,QCR8 CYC1 RP3,TCM1 QCR2	CIII-FES,RIP1	GDH2 UGA1 GLN1
GRF2	HAP1	HAP2\;HAP3\;HAP4	HOMOL	HSE,HSTF	HSE,HTSF	ICRE	IRE	LEU3	MAL63
HSC82	CYC7 CYC7,CYP3 CTT1 CYC1 CYB2 CYT1	SPR3 COX6 QCR8 CYC1		HSP70,SSA1 SSA4 CUP1 HSC82 SIS1	HSP70,SSA1 SSA4 CUP1 HSC82 SIS1	ACS2	IME1	LEU2 LEU1	MAL61

MATalpha1	MATalpha2	MCB	MCM1	MIG1	MOT3	MSE	NBF	PAE	PDR1
MFALPHA2	HO	CDC2	CLN3	MAL61	Ty	SPR3	INO1	Ty1	PDR3
STE3	MFA1	CDC9	CDC28	GAL10				Ty2	SNQ2
MFALPHA1	STE2	CDC6	CDC47	GAL1					PDR15
	BAR1	CLN1	SWI5	GAL3					YQR1
	STE3	POL1	DIT1	SWI5					HXT9
	STE6	CDC21	MFA1	SUC2					HXT11
	MFA2		SWI4	HAP4					PDR5
			STE2	FBS1					
			PMA1	FBP1					
			MFALPHA2	GAL4					
			CLB1						
			BAR1						
			FAR1						
			HSP150						
			CDC6						
			STE3						
			STE6						
			CCP1						
			PCK1						
			CDC46						
			MFA2						
			MET2						
			MFALPHA1						
			PIS1						
			CLB2						
PHO2	PHO4	PQBOX	PRE	PRP1	PUT3	QBP	RAP1,EBF1	RC2;RC1	REB1
PHO5	PHO5	SAG1	Ty1	URA3	PUT2	GAL10	CDC19,PYK1	CYC1	PGK1
HIS4	PHO8		Ty2	PHR1		GAL1	PHO5		RPO21
HO	PHO81		FUS1				TEF2		CDC9
	PHO84						HIS4		TRP1
							PGK1		TPI1
							TPI1		SWI5
							ITR1		ILV1
							RPL16A		ACT1
							ENO1		TRP5
							ENO2		FAS1
							BCY1,SRA1		RAP1
							RNR2		SIN3
							PEM2		TOP1
							FAS1		FAS2
							PDC1		
							TEF1		
RME1	ROX1	RP-A	SCB	SFF	SKO1	STE12,PRE	SWI5	T4C	TAF
IME1	HEM13	POX1	HO	SWI5	SUC2	Ty1	HO	IME2	CIII-subVIII,QCR8
CLN2	ANB1	POT1	CLN1	CLB1		Ty2	EGT2	INO1	CYC1
	ROX1	FOX2	CLN2	CLB2		FUS1			RP3,TCM1
						MFA1			QCR2
						STE2			
						MFA2			

TATA,TBP	Tc\weak_TATA	TSS_Inr	UAS1CHA	UAS1ERG11	UAS2CHA	UAS2ERG11	UASCAR	UASGABA	UASGATA
MAL61		YAT1	CHA1	ERG11	CHA1	ERG11	ARG5,6	UGA4	UGA4
PHO5		HAP3					CAR2	UGA1	
LEU2		PIM1					CAR1		
HIS4		ROX3							
CHA1		RPL2A							
UGA4		CHS2							
CUP1		ALG1							
SUC2		HPC2							
INO1		ARO4							
CYC1		HIS4							
CTS1		GLK1							
HSC82		CBS1							
ADH2		CDC9							
ARG1		UGA4							
ARG8		HEM13							
GCY1		KGD2							
HIS3		HEM1							
CLN2		DIT2							
		CYC7							
		ARG5,6							
		MET6							
		COX4							
		POX1							
		HAP2							
		PRP18							
		MUQ1							
		UGA1							
		SPR3							
		ADE3							
		PHO81							
		ENO1							
		ARG4							
		PUT2							
		DCD1							
		ENO2							
		SGA1							
		SUC2							
		DAL2							
		ANB1							
		CYC1							
		IME1							
		CPA2							
		DAL5							
		MET14							
		STE6							
		URA1							
		MSS51							
		QRI5							
		CTS1							
		URA4							
		PHO84							
		HSC82							
		CLN1							

		ADE4 ADH2 COX5A ACC1 SIN3 ARG1 ARG8 RAT1 GCY1 HIS3 CPA1 TEA1 RPA190 CAR1 RPL1 CLN2 CLB2 RPO26							
--	--	--	--	--	--	--	--	--	--

UASH	UASINO	UASPHR	UASRAD	UASSGA	UAST52,ORE	UESPHR	UIS	UME6	URS1ERG11
SPO7	INO1	MGT1	RAD18	SGA1	FOX3,POT1	PHR1	DAL4	SPO13	ERG11
RFA1		RAD16	RAD6				DAL2		
GAL10		RAD9					DAL7		
GAL1		RAD23							
UME6		RAD51							
ZIP1		MAG1							
MEI4		RAD4							
DMC1		RAD6							
SPO11		RNR3							
SPO13		RNR2							
SPO16		RAD26							
REC104		RAD7							
HOP1		RAD52							
RED1		RAD50							
REC114		PHR1							
MER1		RAD1							
MCK1		MEC2,SPK1,RAD53							
MEK1									

URS1H	URS1HO	URS1HSC82	URSF	URSINO	URSPHR	URSPOX1	URSSGA	XBP1
ZIP1	HO	HSC82	HSP70,SSA1	INO1	PHR1	POX1	SGA1	CLN1
MEI4		HSP82						
DMC1								
SPO13								
SPO16								
REC104								
HOP1								
IME2								
RED1								
REC114								
MER1								
MEK1								
CAR1								

postdiauxi_shift_element	heat_shock(not_HSE)	repressor_of_CAR1
SSA3		HSP70,SSA1 CDC19,PYK1 CTA1 HSF1 CTT1 ENO1 MES1 ARG4 G3PDH,TDH2 CYB2 ILV2 TOP1 CAR1

C. Strongylocentrotus purpuratu (sea urchin)—first row is the name of the regulator gene, those listed below are regulated by the gene in the top row [16,21,46]. Annotation in parentheses denote whether gene is upregulated (+) or downregulated (-), when known. The exact transmission functions are unknown.

SpMyb	SpRunt-1	SpGCF1	Endo16	SpOtx	Spec2a	Spec2b	Spec2c	Spec2d	Spec1	SpF1	Sp7II
CyIIIa(-)	CyIIIa(+)	CyIIIa(+) Endo16(+) SM30(+) SpAN(+)		Endo16(+) Spec2a(+) Spec2d SpHE(+)						Spec2a(+) Spec2b Spec2c Spec2d Spec1(+) SpU^(+)	CyIIIa

SpTEF-1	SpZ2-1	SpOct	SpZ12-1	SpP3A2	SM50	SpSHR2	SpCOUP-TF	CyIIIb	SpGCF1	UHF1	SpHMG1	
CyIIIa(+) SpMTA(+)	Spec1 SM50 CyIIIa(-)	CyIIIa(+) H2B_alpha(+)	CyIIIa(-)	Spec1 SM50 CyIIIa(-) SpMTA(-)				CyIIIb(-) CyIIIb(-)		SM30(+) SpAN(+) Endo16(+) CyIIIa(+)	EH4(+) LH4(+)	CyIIIb(-)

D. Melanogaster drosophila (fruit fly), three separate genetic networks involved in development during embryogenesis—first row is the name of the regulator gene, those listed below are regulated by the gene in the top row [21,28]. Annotation in parentheses denote whether gene is upregulated (+) or downregulated (-), when known. The exact transmission functions are unknown.

GAP early cis/trans interactions:

BCD	HB_pst	KR	NOS	CAD	TLL_pst	KNI	GT	HKB_ant	DL	HB_ant	HKB_pst	TOR	TLL_ant	BTD
KR(+)	KR(-)	KNI(-)	HB_ant(-)	GT(+)	HB_pst(+)	KR(-)	KR(-)	OTD(-)	HKB_ant(-)	KR(+)	GT(-)	TLL_ant(+)	BTD(-)	CNC(-)
HB_pst(-)	KNI(-)	GT(-)		KNI(+)		GT(-)	KNI(-)		BTD(-)	CAD(-)	HB_pst(-)	OTB(+)		
CNC(+)	GT(-)	HB_ant(-)							TLL_ant(-)	KNI(-)		HKB_ant(+)		
KNI(+)										GT(-)		CNC(+)		
GT(+)										BTD(-)		HB_pst(+)		
HB_ant(+)												HKB_pst(+)		
EMS(+)												TLL_pst(+)		
HKB_ant(+)														
BTD(+)														
OTD(+)														
TLL_ant(+)														

Head GAP network:

ems	hb	BCD	hkb	abd B	dorsal	t11	otd	Tor	btd
slp(-)	btd(-)	cnc(+)	otd(-)	ems(+)	slp(-)	btd(-)	gsc(+)	t11(+)	cnc(+)
		btd(+)			btd(-)			otd(+)	
		otd(+)						hkb(+)	
		hb(+)							
		ems(+)							

Pair-rule network:

dpp	eve	h	prd	Ten_m	run	abdA	wg	odd	SLP	ems	ftz	Ubx	jak_stat	FTZ_F1_alfa	opa	Autp
opa(-)	prd(+)	eve(+)	en(+)	prd(+)	h(-)	opa(+)	eve(+)	en(-)	wb(+)	SLP(-)	abdA(+)	opa(-)	eve(+)	ftz(+)	wg(+)	opa(+)
	odd(-)	prd(-)	wg(+)	SLP(+)	prd(+)			eve(-)	en(+)		Ubx(+)		run(+)		run(+)	
	SLP(-)	ftz(-)	run(+)		en(+)			wg(-)					ftz(+)			
	ftz(+)	run(-)			ftz(+)			ftz(-)								
	run(+)															

IX. Conclusion

Our goal is to infer global gene regulatory causal networks from observational data on gene expression, or alternatively from some combination of experimental (intentional variable manipulation) and observational data. Ideally, we would also like to represent not only the causal relationship between each variable (level of gene expression) but the transmission functions between them as well. Our method must also be capable of being actually implemented, i.e., it must rely on an algorithm that can be implemented on a computer and return an answer in a reasonable amount of time, and it must be able to use biological data that has a significant amount of noise both from experimental error and inherent biological variability.

Our choices for representing the causal relationships between variables range from specifying each and every variable in the model to allowing for or identifying latent variables. In our very simplified conception of gene regulatory networks, we tend to think of one gene directly influencing another. However, genes can only be indirect causes of expression levels of other genes—they often have many intermediaries. For instance, for a gene to influence the expression levels of another gene we must consider a number of separate events/processes such as transcription initiation, mRNA splicing (in eukaryotes), mRNA transport (in eukaryotes), mRNA degradation, translation initiation, post-translational modification, and degradation of resultant protein. Each of these processes have their stochastic components as well as potentially have complex biochemical interrelationships (as well as external factors) that are not modeled directly and therefore rightly constitute latent causes.

Our choices for representing the transmission functions also range from simple boolean representation to systems of differential equations. Biochemically, the processes sometimes have boolean characteristics (e.g., AND gate in the form of two separate transcription factors needing to be present simultaneously) as well as differential equation characteristics (i.e., chemical kinetics, rate constants, etc.). Despite the ideal characteristics we desire, many approaches have been proposed, such as:

Artificial neural networks (ANNs)—these are good for modeling/prediction, especially when the training data is noisy as in the case of microarray data. However, they do not reveal a causal network.

Circuit simulations [McAdams]—some have proposed “reverse engineering” regulatory networks in terms of their similarity to electric circuitry. However, these have only been thus far implemented on a small scale from substantially detailed data from a limited gene regulatory/metabolic pathway system. Although these models appear to be quite detailed and precise, it is not clear how to use large amounts of data (such as in the case of microarrays) to derive large-scale regulatory networks.

Boolean networks—while easier to implement than many other approaches, boolean networks lack the detail we seek. Further, they do not account for noisy data.

Covariational/clustering/PCA—these show possibly related (e.g., co-regulated genes) but do not give the causal relationship between variables nor do they allow us to derive transmission functions.

Bayes net—in a suitable implementation, this can give us the causal graph, a linear approximation of the transmission functions (i.e., SEM), can use noisy input data, and can be made to account for latent variables. A drawback is that the Bayes net approach is limited to acyclic graphs, and therefore the approach cannot deal with genes that are autoregulated or otherwise form cyclic structures. While not the perfect approach, it is a good compromise given the alternatives.

References

1. Affymetric GeneChip expression analysis manual (<http://www.affymetrix.com/support/technical/manuals.affx>)
2. Akutsu, T., Kuhara, S., Maruyama, O., and S. Miyano, A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions, *Genome Informatics (GIW'98)*, 151-160, (1998).
3. Alberts, B., et. al., *Molecular Biology of the Cell*, Garland Pub, 3rd edition (March 1994).
4. Arkin, A, Ross, J., McAdams, H., Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected *Escherichia coli* cells. *Genetics*. Aug;149(4):1633-48 (1998).
5. Bay, S., Shrager, J., Pohorille, A., & Langley, P. Revising regulatory networks: From expression data to linear causal models. Submitted to *Journal of Biomedical Informatics*. (2002).
6. Brown D, Jarvis R, Pallotta V, Byrom M, and Ford L., RNA Interference in Mammalian Cell Culture: Design, Execution and Analysis of the siRNA Effect. *TechNotes* 9(1): 3-5. (2002).
7. Brown, T., *Genomes*, John Wiley & Sons 2nd edition (2002).
8. Campbell, N., Reece, J., Lawrence, M., *Biology* (5th Edition), Benjamin/Cummings (1999).
9. Caron H, van Schaik B, van der Mee M, Baas F, Riggins G, van Sluis P, Hermus MC, van Asperen R, Boon K, Voute PA, Heisterkamp S, van Kampen A, Versteeg R., The human transcriptome map: clustering of highly expressed genes in chromosomal domains. *Science*. Feb 16;291(5507):1289-92 (2001).
10. Chartrand, Gary, *Introductory Graph Theory*, Dover Publications, New York (1985).
11. Cherry J., Ball C., Weng, S., Juvik, G., Schmidt, R., Adler, C., Dunn, B., Dwight, S., Riles, L., Mortimer, R., Botstein, D., Genetic and physical maps of *Saccharomyces cerevisiae*, *Nature* 387(6632 Suppl):67-73 (1997).
12. *Computation, Causation, and Discovery*. Eds. C. Glymour and G. Cooper. Menlo Park, CA, Cambridge, MA: AAI Press/The MIT Press (1999).
13. Cooper G.F. and Yoo C., "Causal Discovery from a Mixture of Experimental and Observational Data", *Proceedings of Uncertainty in Artificial Intelligence*, Morgan Kaufmann, CA, p116-125, (1999).
14. Cooper, G. and Herskovits, E., A Bayesian Method for the Induction of Probabilistic Networks from Data, *Machine Learning*, 9, pp. 309-347 (1992).

15. Cooper, G., An overview of the representation and discovery of causal relationships using bayesian networks. *Computation, Causation, and Discovery*. Eds. C. Glymour and G. Cooper. Menlo Park, CA, Cambridge, MA: AAAI Press/The MIT Press, pp. 3-62 (1999).
16. Davidson et al., A Genomic Regulatory Network or Development, *Science* 295 (5560): 1669-1677 (2002).
17. D'haeseleer, P., Wen, X., Fuhrman, S., Somogyi, R., Linear Modeling of mRNA expression levels during CNS development and injury. *Pacific Symposium on Biocomputing '99*, pp. 41-52 (1999).
18. D'haeseleer, P., Wen, X., Fuhrman, S., Somogyi, R., Mining the gene expression matrix: inferring gene relationships from large scale gene expression data. In R. C. Paton and M. Holcombe, editors, *Information Processing in Cells and Tissues*, pages 203-212. Plenum Publishing, (1998).
19. Eisen, Spellman, Brown, Botstein, Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci* 95:14863-8, (1998).
20. Friedman, N., Linial, M., Nachman, I., Pe'er, D., Using Bayesian Networks to Analyze Expression Data, RECOMB 2000: 127-135 (2000).
21. GeNet http://www.csa.ru/Inst/gorb_dep/inbios/genet/genet.htm
22. Goss, P., Peccoud, J., Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets, *Proc Natl Acad Sci U S A*. Jun 9;95(12):6750-5 (1998).
23. Knudsen, S., *A Biologist's Guide to Analysis of DNA Microarray Data*, John Wiley & Sons, 1st edition (2002).
24. Koelle, M., et. al., *C. elegans Gene Knockout Protocols* (http://info.med.yale.edu/mbb/koelle/protocols/protocol_Gene_knockouts.html)
25. Liang S, Fuhrman S, Somogyi R., Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pac Symp Biocomput.* 18-29 (1998).
26. Masters, J. *Animal Cell Culture: A Practical Approach*, Oxford University Press, 3rd edition (2000).
27. McAdams, H., Arkin A., Simulation of prokaryotic genetic circuits, *Annu Rev Biophys Biomol Struct.* 27:199-224 (1998).
28. Nasiadka, A., Dietrich, B., Krause, H., Anterior-posterior patterning in the *Drosophila* embryo , *Advances in Developmental Biology and Biochemistry*, Vol. 12 155-204 (2002).
29. National Human Genome Research Institute website (<http://www.genome.gov/>).
30. National Institutes of Health website (<http://www.nih.gov/news/stemcell/>).
31. Oak Ridge National Laboratory website (<http://www.ornl.gov/hgmis/medicine/genetest.html>).

32. Onami, S., Kyoda, K. M., Morohashi, M., and Kitano, H. The DBRF method for inferring a gene network from large-scale steady-state gene expression data. In *Foundations of Systems Biology*, H. Kitano, ed. (Cambridge, MA: MIT Press). pp. 60-75 (2001).
33. Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, Palo Alto (1988).
34. Peleg, M., Yeh, I., Altman, R. B., Modelling biological processes using workflow and Petri Net models., *Bioinformatics*. Jun;18(6):825-37 (2002).
35. Ptashne, M., Gann, A., *Genes & Signals*, Cold Spring Harbor Laboratory Press (2002).
36. Schena, M., *DNA Microarrays: A Practical Approach*, Oxford University Press (1999).
37. Singh, M., Valtorta, M., (1993). An Algorithm for the Construction of Bayesian Network Structures from Data, *Uncertainty in Artificial Intelligence, Proceedings of the Ninth Conference*, Washington, DC, pp. 259- 265 (1993).
38. Spellman et al., Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast *Saccharomyces cerevisiae* by Microarray Hybridization. *Molecular Biology of the Cell* 9, 3273-3297 (1998).
39. Spirtes, P., Glymour, C., and Scheines, R., editors. *Causation, Prediction and Search*. Springer-Verlag, (1993).
40. Spirtes, P., Meek, C., Learning Bayesian Networks with Discrete Variables from Data. KDD 294-299. *Proceedings of the 9th Annual Conference on Uncertainty in AI*, 259-265 (1995).
41. Thieffry, D., Huerta, A., Perez-Rueda, E., Collado-Vides, J., From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in *Escherichia coli*, *BioEssays* 20:433–440 (1998).
42. Thomas, R., Boolean formalization of genetic control circuits. *J. Theor Biol.* 42:563585. (1973).
43. Weizmann Institute of Science (<http://www.rzpd.de/cgi-bin/cards/listdiseasecards?type=full>)
44. White, R., White, J. *Gene Transcription: Mechanisms and Control*, Blackwell Science Inc., 1st edition (2001).
45. Wyrick JJ, Young RA , Deciphering gene expression regulatory networks, *Curr Opin Genet Dev* Apr;12(2):130-6 (2002).
46. Yuh, C.-H., Bolouri, H., and Davidson, E. H. Genomic cis-regulatory logic: experimental and computational analysis of a sea urchin gene. *Science* 279, 1896-1902, (1998).

Appendix A – The PC Algorithm

A.) Form the complete undirected graph C on the vertex set V .

B.)

$n=0$

repeat

 repeat

 select an ordered pair of variables X and Y that are adjacent in C such that $\mathbf{Adjacencies}(C,X)\setminus\{Y\}$ has cardinality greater than or equal to n , and a subset S of $\mathbf{Adjacencies}(C,X)\setminus\{Y\}$ of cardinality n , and if X and Y are d-separated given S delete edge $X - Y$ from C and record S in $\mathbf{Sepset}(X,Y)$ and $\mathbf{Sepset}(Y,X)$;

 until all ordered pairs of adjacent variables X and Y such that $\mathbf{Adjacencies}(C,X)\setminus\{Y\}$ has cardinality greater than or equal to n and all subsets S of $\mathbf{Adjacencies}(C,X)\setminus\{Y\}$ of cardinality n have been tested for d-separation;

$n=n+1$;

 until for each ordered pair of adjacent vertices X, Y , $\mathbf{Adjacencies}(C,X)\setminus\{Y\}$ is of cardinality less than n .

C.) For each triple of vertices X, Y, Z such that the pair X, Y and the pair Y, Z are each adjacent in C but the pair X, Z are not adjacent in C , orient $X - Y - Z$ as $X \rightarrow Y \leftarrow Z$ if and only if Y is not in $\mathbf{Sepset}(X,Z)$.

D.) repeat

 If $A \rightarrow B$, B and C are adjacent, A and C are not adjacent, and there is no arrowhead at B , then orient $B - C$ as $B \rightarrow C$.

 If there is a directed path from A to B , and an edge between A and B , then orient $A - B$ as $A \rightarrow B$.

until no more edges can be oriented.