GENETIC NETWORK MODELLING AND INFERENCE

DANIEL BERGMANN, MMath, MSc

Thesis submitted to The University of Nottingham
for the degree of Doctor of Philosophy

MARCH 2010

Dedicated to the pursuit of knowledge

Scientia Omnia Vincit

Abstract

Modelling and reconstruction of genetic regulatory networks has developed in a wide field of study in the past few decades, with the application of ever sophisticated techniques. This thesis looks at how models for genetic networks have been developed from simple Boolean representations to more complicated models that take into account the inherent stochasticity of the biological system they are modelling.

Statistical techniques are used to help predict the interaction between genes from microarray data in order to recover genetic regulatory networks and provide likely candidates for interactions that can be experimentally verified. The use of Granger causality is applied to statistically assess the effect of one gene upon another and modifications to this are presented, with bootstrapping used to understand the variability present within the parameters. Given the large amounts of data to be analysed from microarray experiments, clustering techniques are used to help reduce the computational burden and novel algorithms are developed to make use of such clustered data. Variability within clusters is also considered, by developing a novel approach with the use of principal component analysis.

These algorithms that are developed are implemented with an observed dataset from Xenopus Laevis that has many genes but few timepoints in order to assess their effectiveness under such limited data. Predictions of likely interactions between genes are provided from the algorithms developed and their limitations discussed. Using extra information is considered, where a further dataset of gene knockout data is used to verify the predictions made for one particular gene.

Acknowledgements

First, and foremost, I would like to thank those who have guided my research. To Professor John King, for providing this interesting project in the first place and allowing me to follow my own direction. To Professor Andy Wood, for his wide-ranging knowledge and the many hours of thought provoking discussions across a variety of areas of statistics which have led to some interesting directions as part of my research. To Dr Matt Loose, for providing the biological expertise and data on which to test my theoretical developments as well as pointing out where the theory made no sense to the application. I must also say a big thankyou to the BBSRC for providing the financial funding for my PhD and allow me to concentrate on research without other distractions.

Thanks must also go to those people in my life who have supported me throughout what has been a period of highs and lows. To my parents for being supportive of my decision to undertake this pursuit. To my extended family who have given me their support and interest in what I'm doing. A lot of thanks and love to my wonderful nephews James, Jack and Logan for reminding me why the pursuit of knowledge is needed in an ever changing world, as well as reminding me that life isn't all about study. Of course, many big thankyous must go to those friends who have been there through all the good and bad times. I won't name you all here as I'm bound to forget someone, but your support has been greatly appreciated.

Thanks are also extended to the many people I have met in the course of exploring statistics, from those who I have attended conferences with and opened my mind to other areas outside of my own, to the great work of the Royal Statistical Society for furthering the use of statistics in this country.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Genetic networks are representations of how genes interact within a cell. The dynamics of such networks give rise to biologically observable change, such as the ability of a stem cell to develop into one of hundreds of possible cells. Understanding how the dynamics change can give deep insight into genetic diseases, such as certain cancers. There has long been interest, therefore, to apply mathematical and statistical techniques to such networks in order to reduce costly and lengthy biological experiments and help direct where resources may be best used.

Modelling of genetic networks has its origins in the 1960s when Kauffman [1] used simple Boolean logic to consider how genes interact with each other. Since then, the area of modelling and reconstructing genetic networks has developed considerably, with much active research into the area. This thesis looks at how such networks are modelled with a focus on using biologically observed data to recover such networks. This is developed to prediction of interactions and applied to a real world example by developing statistical models of causal interactions between genes.

Novel contributions within this thesis are presented by extending the idea from the paper by Mukhopadhyay and Chatterjee [2] to reduce the computational burden generated by analysing all possible pairs of interactions where there may be many thousands of individual genes. These extensions have been implented in three different way. Firstly, gene data is clustered and the centroid of each cluster is used to assess the significance of the interaction between all possible pairings between the clusters. As this may give rise to within cluster

variance, the second extension is to use Principal Components to take into account this variance. Finally, the results generated by the algorithms presented are compared against gene knockout data to assess whether extra support is given for individual actions.

This chapter explains the biological mechanisms underpinning how gene interacts with each other. Further to this, the development of techniques for measuring data from this biological process is explained, by the use of microarrays and their limitations. A review of the literature is then given, which shows how wide and varied the field has become from its origins and the range of applied mathematical and statistical techniques that have been used.

1.1 Biological Foundations

Living organisms are highly complex systems that develop from very small beginnings. A sperm may fuse with an egg to then develop into a fully grown animal, with individual physical characteristics. These characteristics are derived from genes, which act as a source of information for how the organism should develop into many different types of cells. Stem cells, which act as master cells to become one of many different types of individual cell, need to know how to become one cell type over another. The genes carried within the stem cells produce gene products that can interact with other genes. Describing the interaction of genes is the goal of Genetic Regulatory Networks. These are directed networks that describe whether a gene targets another gene in some way and has a direct effect on one another.

1.1.1 Genes and gene function

DNA (Deoxyribonucleic Acid) is found replicated within the nucleus of every cell. It is made out of four bases, which bind together in complementary pairs: Adenine binds to Thymine and Cytosine binds to Guanine. These occur in two strands which bind together in a double helix fashion with a phosphate-deoxyribose backbone to bind each strand. A gene is a sequence of these bases occurring along the DNA strand. Genes provide a blueprint for life by acting as a code to produce protein products.

A gene produces proteins through an intermediary product called mRNA (messenger ribonucleic acid). This is similar to DNA except that the Thymine base is replaced by another called Uracil. Genes undergo transcription into mRNA, by which a polymerase reads the bases of the gene and makes an mRNA copy. From these mRNA molecules a ribosome then translates these into proteins. Triplets of such bases, known as codons, represent any of 20 amino acids (or one of the 3 codons to signify the transcription or translation process to stop). Proteins are strings of these amino acids which differ greatly in size and function.

Genes can interact with one another through their protein products in a highly complex manner. Transcription factors of a gene, a type of protein molecule, can bind upstream of a gene at given binding sites of not only their own gene but other genes as well. There may be multiple binding sites for a gene to which different transcription factors have the ability to bind. It is the arrangement in which they are bound which allows the rate of transcription for this gene to change, either increasing (activation) or decreasing (repression). This measure of the rate of production of mRNA is termed gene expression.

At the heart of this lies an important concept within genetics termed the Central Dogma. The central dogma states that there is a cyclical behaviour of how genes and their products interact, whereby genes may transcribe RNA which is then translated into functional proteins which may then interact with a gene by binding at some promoter site upstream of the gene it is targeting. The central dogma is not strictly true, as retroviruses may cause a direct link from RNA back to the gene without the need for proteins. However, as this is not required for the work within this thesis, the assumption shall be that the central dogma holds in order to consider the biological mechanisms underpinning the use of genetic networks. Figure 1.1 represents this cyclical nature diagrammatically.

1.1.2 Genetic Regulatory Networks

The complex interactions between genes and their protein products such as transcription factors or signalling molecules lead to a network representation. Within a gene network, the nodes of the network are represented by the genes with links representing an interaction. A directional arrow from gene A to gene

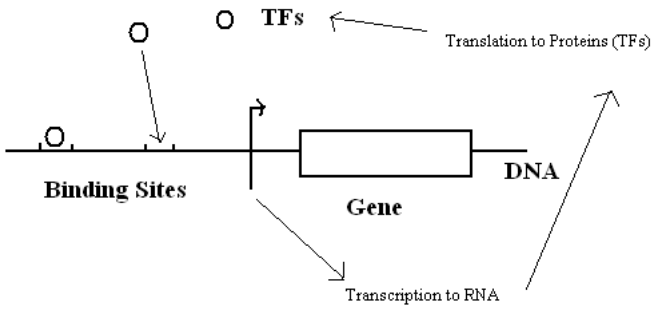

Figure 1.1: Model of the central dogma. The DNA part of the gene transcribes to mRNA, which is then translated into a functional protein. Transcription factors can then bind to promoter sites upstream of genes in order to influence the rate of mRNA transcription.

B shows that the protein products translated from the mRNA transcribed by gene A bind to the promoter site of gene B in order to increase the rate of transcription of mRNA for gene B, or gene A *activates* gene B. A flathead arrow shows that gene A acts to turn off the transcription of gene B, or gene A *represses* gene B. An example is given in Figure 1.2. Where gene A is shown to have exhibit some response on gene B, then it is said that gene A *targets* gene B.

Further reference to the biological foundations of genetic systems can be found in Lewin [4] and Latchman [5].

1.1.3 Microarray Technology

When trying to reconstruct genetic networks, data needs to be extracted from the biological experiment being performed. The two natural choices would be to measure the abundance of the biochemical molecules present within the sample, such as the protein products or the mRNA that has been transcribed. Measuring protein levels is difficult due to the structure of the molecules and the inability to accurately measure them at present. In contrast, methods for measuring the abundance of mRNA have been well developed in the form of

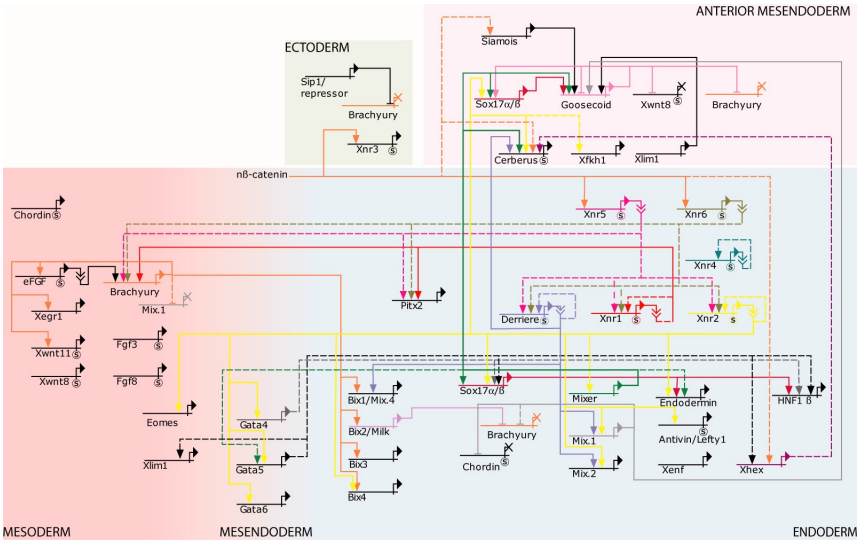

Figure 1.2: A genetic network for mesendoderm formation in Xenopus Laevis as reproduced from Loose and Patient [3]. Interactions between labelled genes are represented by directed lines between these genes, with an arrowhead representing activation and a flathead representing repression.

microarrays.

The amount of mRNA present within a sample is also termed the *gene expression* level of the gene from which the mRNA was transcribed. Microarrays can measure simultaneously many thousands of genes. The two main types of microarrays are cDNA microarrays [6] (which typically compare two samples) and Affymetrix [7], a chip based design which measures mRNA expression level for a single sample in comparison to the known genes present on the chip. A further technology Luminex, a bead based array system, is showing promising development but we do not consider this here.

1.1.4 cDNA Microarrays

Suppose we wish to test two samples to compare their gene expressions, such as a healthy against a cancerous cell, and to determine the variation in the expression levels of various genes. The use of cDNA microarrays is used to test such variation.

Firstly, a glass slide is spotted at known sites with half strands of DNA of known genes for the biological system being measured. These *arrays* also contain genes that are always present and should always be expressed with little variation, known as housekeeping genes, in order to detect any defects within the array. The spots are made by binding together nucleotides that are equivalent to the half strand of gene DNA. Many of these strands are placed on the array at the known spots.

Next, the samples to be analysed are prepared to extract the mRNA and a reverse transcriptase added to obtain cDNA (complementary DNA) strands that can bind to the DNA strands on the array. The reference sample (e.g. a healthy cell) is labeled with a fluorescing dye, Cy3, and the test sample (e.g. cancerous cell) is labeled with a different coloured fluorescing dye, Cy5.

Next, the samples are amplified in solution to create more copies of the same cDNA strands and mixed with the array slide to enable the appropriate cDNA strands to attach, termed *hybridisation*. Any excess is then washed to remove excess sample and the amount of sample that has bound to the array can be used as a measure of how much mRNA is present within the sample.

In order to measure the gene expression level of the genes, the array is scanned with a laser that causes the attached dye to fluoresce and the intensity of this fluorescence can be measured. This output will then contain many spots of varying intensity representing the amount of mRNA for each of the individual genes that each spot represents. The image can then be read by image analysis software to determine the relative level abundance of gene in each sample, from the Cy3 which fluoresces green and the Cy5, which fluoresces red.

If the resulting spots are these individual colours then the gene is only activated in one of the samples, with a gene abundant in both samples showing as yellow in the output sample. Where the gene is not activated in neither the reference nor the test sample, a black spot is shown.

This technique is useful where two samples are to be compared but a more suitable microarray test for a single sample is the use of Affymetrix arrays.

1.1.5 Affymetrix

Affymetrix chips are prefabricated arrays spotted with DNA strands ready for hybridising from a sample [7]. In comparison to cDNA microarrays, the length of the strands is fixed at 25 nucleotides (compared to unlimited length in cDNA microarrays). By repeating various sections of the mRNA strands it means that the same mRNA strand will bind to different spots on the chip. This reduces the variance in spots being misread by combining the measurement values together. Mismatches are also used as a control.

Similar to cDNA microarrays, the Affymetrix chip is scanned with a laser to read the levels of expression for each of the known spots. Using statistical techniques, the expression level can be measured from a combination of the perfect matches and the mismatches for each gene on the array. Details of how this is calculated can be found in Irizarry et al [8].

1.1.6 Reducing Error in Measurement

As microarray measurement is dependent on human and machine interaction in the process, there are potentially many sources of noise. This can come from production of arrays themselves, where printing tips to spot the nucleotides onto the arrays may be slightly out of place, to the sample production and then scanning errors. A more complete list of potential sources of error can be found in Zakharkin et al. [9].

Once the array has been scanned and the array image produced, the raw expression level can be obtained by image analysis software. In the case of cDNA microarrays, the intensity of the signal at each spot is measured. With the Affymetrix procedure of having repeats and mismatches, these are performed by combining these in a meaningful way.

In order to reduce the noisiness of the signals produced in obtaining this raw expression level, *normalisation* can be applied which compares the gene expression levels of the genes with that of the housekeeping genes. This has evolved into a vast field of study in its own right and there are many excellent sources of reference [10], [11], [12], [13], [14], [15].

Once normalisation has been applied, these normalised signal measurements

are often log-transformed and these log-normalised values used as measures of the expression levels for the genes in the samples. There exists much commercial and free software to perform normalisation to produce a useful dataset, such as BRB-Arraytools [16]. This data can then be analysed in order to reconstruct genetic networks.

1.2 Literature Review

Genetics and the understanding of the interplay between genes developed considerably in the mid part of the 20th century. Given the network represenation of such genetic interactions, applying mathematical techniques was a natural progression. The first models were developed by Kauffman in the 1960s [1] by using Boolean logic and Boolean functions to represent the nodes of a network as genes and understand how the dynamics of a network change depending on choice of function and state of the overall network. Such simple models exhibit stability properties, such as steady states, notably through the application of certain canalysing functions that seek to explain behaviour on a few key nodes within the network [17]. The nature of binding and unbinding transcription factors to promoter sites of a gene is inherently stochastic; furthermore, the transcription rate of a gene into mRNA varies greatly and such activity levels are too simplistic to be represented by simple states of being on or off. By extending the Boolean model to consider discrete levels of activity of a gene leads to a more accurate biological representation of the network state, as in [18].

The use of Boolean functions also does not take into account the strength of each individual interaction between genes which may vary greatly. Probabilistic Boolean networks parameterise the interaction between nodes, as in as in Dougherty and Shmulevich [19], Ivanov and Dougherty [20] and Shmulevich et al [21]. Like their Boolean counterparts, they exhibit steady state properties as shown by Shmulevich et al [22].

Gene activity is not a discrete process with the expression level of a gene varying continuously. By using a continuous scale, ordinary differential equation models developed using ideas from neural networks, whereby the expression level of a gene is dependent on the expression levels of genes targeting it and parameterisations of the interactions. As with ODE systems, and in comparison to the discrete level models, there exist steady states. These ideas are considered by Weaver [23] and Vohradsky [24] with a further general overview of some of the popular deterministic models is given in Wessels et al [25] , deJong [26] and Smolen et al [27].

One of the best applications of genetic networks is at the developmental stage,

such as where as stem cell undergoes a series of decisions in order to specify what cell type it becomes. This set of decisions, or the lineage of the network, is based on the change of transcription levels of key genes at each decision stage. Initially, where low levels of functional proteins are present, it is difficult to predict which choice of fate exists before the decision is made. This biological process has been studied in haematopoietic stem cells, as in Laslo et al [28] and modelled by Roeder and Glauche [29] for which Loose et al [30] describe the developmental network.

The lambda phage repressor system is a simple two gene model widely studied due to the low number of nodes required, such as by Ackers et al [31]. This system switches between different steady states due to the two different biological processes in play, such as in Cinquin and Demongeot [32]. Santillan and Mackey [33] show further that one steady state is more stable than the other. Other such bistable systems are considered by Lai et al [34] and Deineko [35] for a mammalian cell cycle.

Understanding attractors and steady state analysis of genetic networks is important as they can be considered as representations of cell fate and determination of cell type, originally suggested by Kauffman [1] and discussed further in Huang et al [36]. Given that functional proteins govern cell function due to their relationship with mRNA from the central dogma, models have been proposed which consider the expression level of proteins as well as of mRNA, such as by Hatzimanikatis et al [37] and Karmakar et al [38]. Due to the difficulties in measuring protein levels, there is limited scope for studying such protein based models. However, as the experimental techniques are improved, this will be of increasing interest.

There is a wide range of software available for simulating such ODE models, both commercial and freeware. Some examples include Genetic Network Analyzer by deJong [39] and Genexp [40] which implements the model proposed by Vohradsky [24].

Ordinary differential equation models are useful for observing the dynamics of a network but do not take into account the inherent stochasticity of the underlying physical system that binding and unbinding presents, with such origins considered by Kepler and Elston [41]. The first considerations of adding

stochasticity into models of genetic networks was given by McAdams and Arkin [42],[43] in application to the lambda phage system. By considering how the noise in these systems is controlled, such as in Raser and O'Shea [44],[45], stochastic models can be developed that provide a more accurate representation of the genetic networks.

One way of adding stochasticity is by discretisation of the ODE systems with noise added by the use of stochastic jump processes between timepoints, such as by Tian and Burrage [46]. This requires the time between steps to still be considered. A continuous version can be further improved by the use of stochastic differential equations such as in Rao and Arkin [47] or Chen et al [48]. Given the extension of the model to include this stochasticity, the properties of the models can be considered as with the deterministic models.

The bifurcation of the lambda phage system is considered by Arkin, Ross and McAdams [49] and in a similar system by Toulouse et al [50], with stochastic oscillations in a more general framework considered by Bratsun et al [51]. How the noise is structured is of interest in its own right, such as by Blake et al [52] and Pedraza [53]. The variation in fluctuations by altering the amount of noise present in a simple system is studied by Chen and Wang [54] and Chen et al [55].

Noise may be viewed in the frequency domain, such as by Simpson et al [56],[57] or in a multivariate setting, as in Tomikaa et al [58]. These views of the underlying structure of noise and how it affects the results can be split into intrinsic an extrinsic components, where noise within the system and external noise are considered as individual noise systems, as by Swain et al [59]. This internal noise is considered by Tao [60] for a simple two gene network and then extended to add external noise [61], with Thattal and Van Oudenaarden [62] modelling just extrinsic noise. How this split of internal and external noise affects cell fate is considered by Maamar et al [63] for Bacillus subtilis.

Due to the biological realism of stochastic models, there has been a wide variety of literature produced, with De Jong having produced a comprehensive bibliography of the wide range of deterministic and stochastic models [64] and a more technical consideration of the model details and their use [26]. Of course, such models will have limitations as in Kim and Tidor [65]. Constructing the networks themselves also presents issues, as explained in Blais and Dynlacht

[66], with Bornholdt [67] widening this further to the context of dynamical systems. Francois and Hakim [68] consider design issues both for deterministic and stochastic models. The book by Bower and Bolouri [18] provides a worthwhile reference to the range of models for genetic networks, with a good overall view of biological networks as objects for modelling given in Alon [69].

Most expression data obtained from experiments is based on multiple cells. As experimental procedures develop, the use of single cell data becomes feasible, although expression levels at the individual cell level have already been studies. Gibson [70] first considered modelling activity within a single and the measure of of single cell gene expressions, also considered by Elowitz et al [71], Isaacs et al [72], and Rosenfeld et al[73]. As before, the noise within a single cell should be considered, as in Ozbudak et al [74], with Zhang et al [75] exploring the binary decisions present and their effect within the single cell.

Instead of modelling the dynamics of large scale networks possibly containing thousands of genes, expression dynamics for subnetworks can be considered. The smallest such building blocks, which are repeated throughout larger networks, are the motifs as described by Milo et al [76]. Of particular interest in genetic networks is the feedforward loop described in Dekel et al [77] and Mangan et al [78]. Detecting these motifs, as opposed to larger scale structures, is considered by Keles et al [79] with Ingram et al [80] demonstrating that motif structures alone do not determine overall cell function. Similarly, widening the structure to larger subnetworks of the full network can show that certain parts may be more important to the network dynamics than other parts [79]. The use of such motifs in simple network is considered by Shen-Orr et al [81] applied to a system which governs e. coli.

Differential equation approaches, both with and without stochasticity considered, measure gene expression on a relative, continuous scale. However, individual molecules are present within a sample so modelling at the molecular level can be implemented with the use of stochastic processes. These stochastic simulation algorithms relate back to chemical master equations [18] which have been widely used to model chemical reactions and the number of molecules present. An equivalence to genetic networks exists, where biological molecules

are used instead of chemical molecules [82].

The Gillespie algorithm, as developed by Gillespie [83],[84], provides an equivalent to a master equation which may not be easy to solve. As they are stochastically equivalent, the Gillespie algorithm is widely used in practise for modelling chemical system. This original algorithm is inefficient, so successive refinements have been proposed, such as by Gibson and Bruck [85]. Large scale approximations can be made via the use of Langevin and Fokker-Planck equations [86], [87], [88],[89].. Gillespie et al substantially improved the speed of calculation based on approximation with the use of tau-leaping [90], [91], [92], [93], [94], where the timestep between reactions is not fixed and many reactions may occur. Problems may occur if the number of molecules is allowed to go negative, which is rectified by binomial approximations by Cao et al [95]. Similarly, efficient choice of timestep needs to be considered [96].

Reactions may occur at different timescales, with recent developments exploiting such slow and fast timescales. Purely slow dynamics are investigated by Bundschuh et al [97] with Cao et al [98],[99], and Burrage and Tian [100] extending the Gillespie algorithm to such multiscale reactions. This multiscale Gillespie algorithm is investigated for steady state analysis by Rawool and Ventakash [101]. As an extension, Puchaka and Kierzek [102] combine by deterministic and stochastic regimes, with Vasudeva and Bhalla [103] using this approach to extend the Gibson and Bruck algorithm. Concise overviews of these stochastic simulation algorithms and their limitations are given by Samad et al [104] and Turner et al [105].

The application of stochastic simulation algorithms to genetic networks has been used for various biological systems, such as the lambda phage switch by Salis and Kaznessis [106] and E.coli by Rodriguez et al[107], with Tuttle et al [108] further considering oscillations within this system. Parameter variation for a simple network is explored by Kierzek et al [109]. Using time series data to estimate parameters is considered by Reinker et al [110] and Tomshine and Kaznessis [111], with Wu et al [112] using state space representation to perform this.

As seen in the literature presented so far, a wide variety of techniques has been applied to modelling genetic networks and with such interest in the area, many

software packages have been developed for simulation. Some examples include Biodrive [113], SynTREN [114], SGNSim[115] which implement stochastic algorithms with BioNetS[116] and STOCKS [117] implementing the Gillespie algorithm.

These deterministic and stochastic models are used to model genetic networks in order to understand their dynamics and also to generate artificial data based on known network structures. The reverse view to this is using data obtained from microarrays in order to recover the original network that such data came from. Many of the approaches used for modelling have analogues in network recovery.

Within the Boolean framework, this is considered by Akutsu et al [118],[119] and implemented in the REVEAL algorithm [120].Cho et al [121] using genetic programming and D'Haeseleer et al [122] applying clustering ideas. The use of scoring function for the likelihood of interactions present between nodes is implemented by Gat-Viks and Shamir [123], with Lahdesmaki et al [124] using a consistency based approach.

Differential equation based models are used in an iterative scheme as a means of identifying network structure by Gadkar et al [125]. Using time series data and maximum likelihood methods, De Hoon et al [126] recover ODE models for networks. A perturbation based approach is used by MacCarthy et al [127] for a discretised ODE system, and also by Pe'er et al [128] for inferring subnetworks. A comparative appraoch is used by Ronen et al [129] by assign parameter values to known networks. Such ODE model recovery is implemented in many software applications such as ASIAN [130], BIOREL [131], EXAMINE [132] and SPLINDID [133].

A widespread and popular approach for reconstructing genetic networks is the use of Bayesian networks, which represent the proabilistic state of network depending on directed graphical structure and data at each node, with Beal [134] and Bernard [135] providing overviews and reconstruction in stochastic networks considered by Wilkinson and Boys [82]. Dynamic Bayesian networks introduce the use of directional interactions and data changing over time, such as by the use of time series data for each node with many studies on their appli-

cation to genetic networks such as by Eddy et al [136],Fridman et al[137], Ching et al [138], Chu et al [139], Datta et al [140], Nariai et al [141] and Spirtes et al [142]. Sensitivity within Bayesian networks can alter the specificity in a network, as shown by Husmeier [143]. Lahdesmaki et al [144] also show that these dynamic Bayesian networks exhibit a correspondence with Bayesian network models.

A variety of approaches based on Bayesian networks have been studied. Imoto et al [145], [146] use regression based techniques, whereas Missal et al [147] work with incomplete data that may not be sampled regularly or missing. Noise within such networks can also be considered , as by Streib et al [148]. Motif detection is considered by Tamada et al [149]. Yoo et al [150] use gene knockout data for network inference, where certain genes are switched off in order to understand how the dynamics change without their presence.

As with the stochastic models, these techniques have been applied to specific biological systems. The E.coli network is reconstructed by Ong et al [151] with Perrin et al [152] using an Expectation-Maximisation approach to the same system. Woolf et al [153] look at an embryonic cell fate network. An overview and comparison of some of the variations of Bayesian network methods for reconstruction of genetic networks is presented in Werhli et al [154].

Bayesian networks are part of a larger class of learning algorithms, with other such algorithms also used such as in Li and Yang [155]. Genetic programming and neural networks are used by Motsinger et al [156] with Sokhansanj et al [157] using fuzzy neural networks and exhaustive search algorithms. Scoring based algorithms are used by Nacu et al [158] and also by Nemenman [159] in the context of information theory. External signal perturbations to understand the robustness of network estimates is given by Lipan and Wong [160] and Tegner et al [161]. A Gibbs sampler approach is studied by Brynildsen et al [162]. Shehadeh et al [163] use a slightly different approach of considering the density function of mRNA expression functions in different types of genetic network information to build up a dictionary of functional associations from which true data can be compared.

Given that networks are based on association of genes, clustering techniques

have become useful to identify such groups of genes, such as by Alon and Broad [164], as those with similar expression levels over time may have functional relationships [165],[166]. Similarly, the use of clustering has become an important technique to reduce dimesnion where there may be thousands of genes to analyse and compare. Classification algorithms in genetic networks have been considered by Liu et al [167] and Meltzer et al [168], with an extension via perturbations for prediction by Schreiber and Baumann [169]. Clustering techniques have been combined with other approaches such as spectral methods, by Subramani et al [170], graphical modelling, by Toh and Horimoto [171], principal components, by Yeung and Ruzzo [172], and information theory, by Zhou et al [173].

Obtaining data for inference of genetic networks comes from microarray experiments which measure the expression levels of the genes present within the system being studied. Normalisation techniques convert raw data in the usable data by seeking to eliminate noise within the samples, with many references [10], [11], [12], [13], [14], [15], [174], [175]. The normalisation procedure used will effect the outcome of the data, as explored by Qiu et al [176], with Zakharkin et al [9] discussing the origins of the noise. Two channel arrays depend on ratio based methods for comparison between the two samples with a consideration of the reference level made in Chen et al [177], whereas Affymetrix have developed normalisation techniques based on mismatches, with Irizarry et al [8] discussing some of the most common.

Identifying genes that are differentially expressed, where the expression levels shown are statistically significant, are of interest to determine which genes are most important with a sample and is explored by Storey et al [178] and Li et al [179]. These ideas are developed by Liang et al [180] who use bootstrapping , which is also used by Ma [181] in combination with other statistical techniques. A variety of the methods used for identifying these differentially exrpessed genes is given in Pan [182] and Park et al [183]. Periodic expressed genes, such as those which exhibit circadian rhythms in relation to the cycle of a day, are considered by Wichert et al [184]. The use of maximum likelihood methods applied to these circadian genes are considered by Bakewell and Wit [185].

Instead of trying to reconstruct a whole network or even a subnetwork, such as the approach taken by Bayesian networks, pairwise interactions between all possible pairings of genes can be studied. Correlation measures between genes have been studied along with partial correlation measures, which conditions a correlation on other measures, such as by de la Fuente et al [186] and Schafer and Strimmer [187],[188]. Conditional correlations are used by Rice et al [189] with Wang et al [190] extending this to considering the use of graphical models in particular. This can be extended to multivariate methods for pairwise inteactions, such as matrix methods akin to multivariate linear modelling. Gao et al [191] use this approach for mRNA and protein data, with Ghosh [192] using singular value decomposition.

For such large scale interactions between many genes, the use of computational tools to reduce calculation time are increasingly important. One of these tools is to use parallel computing [193] with Salis et al [194] using this for a large scale networks based on stochastic simulation algorithms and Schwehm [195] applying parallel techniques to large stochastic models. Furthermore, public databases have been created from microarray experiment data described by Penkett and Bahler [196] with ArrayExpress as an example [197]. Such data is usually in the form of time sampled data, with time series analysis used to analyse these.

Time series analysis of recovering genetic networks is considered by Bansal et al [198], Bar-Joseph [199], Bay et al [200] and Bickel [201], with multivariate time series models increasingly being used to not only detect interactions but also estimate parameters. The books by Hannan [202], Lutkepohl [203] and Priestley [204] provide a comprehensive overview of multivariate time series. Identifiyng time series models is possible by the use of information criteria, such as Akaike's Information Criterion [205] and Schwarz's Bayesian Criterion [206], with Kadilar and Erdemier [207] comparing these to show an optimal performance of the BIC in multivariate time series.

Estimating parameters in multivariate time series models is also necessary for inference purposes. Specific classes of time series models are considered by Bagarinao and Sato [208] and Mauricio [209], with software provided for these

purposes, such as that of Schneider and Neumaier [210] for least squares estimation. A particular class of time series mdoels is applied to gene expression data in Fujita et al [211].

Causality is a measure of how one variable may affect and frequently used in economic data modelling. These have an arrow of time associated, whereby one variable may influence another but not the other way round, which naturally leads to their application in recovering genetic networks. Time series models looking at the general change in trend, such as in He and Zeng [212], act as a precursor to more formal statistical models. Granger causality in particular is a statistical technique for quantifying the influence of one variable upon another from time series data of each [213], [214], with Chatterjee and Mukhopadhyay [2] applying this to recovering genetic networks.

Experimental techniques are limited by financial and time resources so inference may typically be based on short range time series. Bootstrapping methods, developed by Efron [215], [216], are resampling techniques used to infer statistical properties of parameters where the data is limited. Their use in time series has been developed recently [217] so are of increasing interest to inference for microarray time series analysis.

This review of the literature shows that a wide range of mathematical and statistical techniques can be applied to modelling and recoveing genetic networks, from simple Boolean models to more sophisticated stochastic models. This list is by no means exhaustive, with frequent papers in the general area of modelling and recovering genetic networks appearing regularly in leading journals such as Bioinformatics.

CHAPTER 2

Modelling Genetic Networks

By reviewing the literature covering the area of modelling and reconstructing genetic networks, it is clear that there are many approaches that have been developed and a wide range mathematical techniques applied in order to improve the models. This chapter looks at the origins of modelling genetic networks, from a simplistic Boolean approach through continuous ODE models to more sophisticated approaches taking into account the stochastic nature of the underlying biological system. Key biological features may be observed in such models.

2.1 Boolean Models

In the 1960s, Kauffman [1] used Boolean logic in order to model genetic networks. These *Boolean networks* assign values to each node of the network, representing a gene in the network. If a gene is switched on and transcribing mRNA, it takes the value 1, with the value 0 assigned to a gene that is switched off. By the use of Boolean functions, which combine the inputs to a gene, the state at the next timestep can be calculated. A steady state, or attractor, can be found for the Boolean network. Kauffman suggested that these steady states are representative of cell types.

In order to understand the dynamics of Boolean networks, small networks consisting of a few nodes, or *motifs*, are studied.

2.1.1 Motif example

Let X_t^i be the state of gene i for $i = 1, ..., n$ at some timepoint t. Under a Boolean model, the state of the gene at time t is either 0 if the gene is not expressed, or 1 if the gene is expressed. For the network in Figure 2.1, suppose there are two input genes A,B targeting a two output genes, C, D. Furthermore, let A be activating C and B repressing, and A repressing D with B activating. Further, A and B are mutually repressing, with the dotted lines representing possible external outputs.

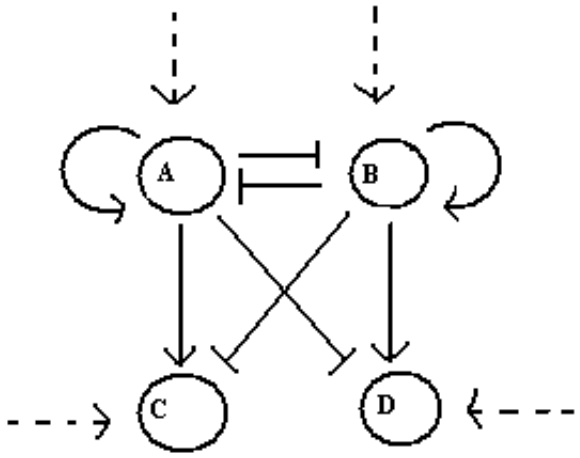

Figure 2.1: Four node motif with two input and two output genes

Consider a smaller subnetwork from this motif, whereby gene A is activating C and B repressing it. The attractor values are dependent on the choice of input and the Boolean function used to combine the inputs. For this example, the three node network has 16 possible choices of Boolean function. Kauffman argues that out of these choices, only one is biologically feasible, which would correspond to the state observed in a real biological system.

In this network, the presence of a repressor (B) which is expressed will always result in the output node (C) being switched off. By having the activator (A) expressed, and the repressor (B) not expressed, the output node (C) will be expressed, as the logic table in Table 2.1 shows. This is equivalent to the logic statement

$$C = A\&\neg B$$

A	B	C
0	0	0
0	1	0
1	0	1
1	1	0

Table 2.1: Logic table for three gene motif

2.1.2 Between input interference

Within a biological network, genes regulating each other will lead to different output states depending on the dynamics of the interaction. For this simple 3 node example let the two input genes A, B be mutually repressing; then the output will eventually be switched off. Table 2.2 shows the initial states of A and B at time t with the output (C) and then at timestep $t + 1$ with all the genes are switched off.

A	B	C	A	B	C
0	0	0	0	0	0
1	0	1	0	0	0
0	1	0	0	0	0
1	1	0	0	0	0

Table 2.2: Three gene motif with inputs mutually repressing.

Clearly this is not representative of what is observed biologically, as many genes would end up switched off and there would be no observed biological development. This simple 3 node network alone is highly insufficient to truly represent a large scale biological network. What it does show, however, is the competing dynamics between nodes are affected by the choice of network interactions. Other inputs in a larger system will have an effect and so will the choice of function used to represent the dynamics.

Certain functions provide a means to fully determine the steady state. Such *canalizing functions*, as suggested by Kauffman [17], uniquely determine the steady state based on the current state of the system. This means that regardless of the inputs, the steady state can always be found and is shown to always be stable.

2.1.3 Extending the Boolean approach

Given a set of inputs, applying a Boolean function as a 'rule' to determine the outcome and then finding the output state is simple to simulate and can be easily extended to large scale Boolean networks. However, it is also too simplistic and not particularly accurate from a biological viewpoint given that genes fluctuate in their ability to regulate and cannot just be viewed as being switched on or off.

Within a pure Boolean framework, the inherent stochasticity of the biological system is not accounted for, as the Boolean dyanmics are wholly deterministic dependent on the Boolean functions and their initial conditions. As opposed to considering just Boolean functions to represent network dynamics, probabilistic Boolean networks add a probabilistic interpretation by parameterising the interactions between nodes[19], [20],[21]. As with a simple Boolean model, they exhibit steady state properties[22].

Another approach taken to extend these Boolean networks is to no longer consider just an on or off state of the gene but to quantify it into multiple levels of expression depending on how active a gene may be [18]. By taking more and more levels this approaches a continuous representation of gene activity, which is more biologically accurate as the level of gene activity may be measured. This continuous representation is now considered.

2.2 ODE Models

Whilst Boolean networks and their multilevel refinements have simple dynamics that are easy to model, they lack biological realism. This is not only due to the simplistic state of expression that they exhibit, but also due to the inherent

stochastic nature of the underlying biological mechanism of binding and unbinding of transcription factors to the promoter sites of genes. Furthermore, assigning values to each node representing a gene as switched on or off does not allow for the wide variation of expression levels of these genes but also for the interactions between genes and how the strength of these interactions are quantified.

A continuous range of values would allow the model greater flexibility. Instead of assigning Boolean values of 0 or 1, the range of values of the expression level of the gene may be in the closed interval $[0, 1]$, with 0 still representing a fully switched off gene and 1 a fully switched on gene, comparable with the Boolean model.

2.2.1 Neural Network Model

The largest class of ODE models used in modelling genetic networks are based on neural networks [23], [46], [25], [24],[218]. Here, the current state of the inputs to a node, combined with weighting factors representing the strength of interaction, are combined to produce a value for the level of expression at that node. The following model is based on that in Tian and Burrage [46].

Let $\boldsymbol{U}(t) = (u_1(t), ..., u_N(t))$ be a vector representing the gene expression level at time t across N genes within a network. Interactions between genes i and j are characterised by a *weight matrix* $\boldsymbol{w}$ where w_{ij} represents the weight of interaction of gene j on gene i and

- $w_{ij} > 0$ gene j activates gene i
- $w_{ij} < 0$ gene j represses gene i
- $w_{ij} = 0$ gene j has no interaction with gene i

The *total regulatory input* r_i for gene i is given by

$$r_i(t) = \sum_{j=1}^{N} w_{ij} u_j(t) + \alpha_i \tag{2.2.1}$$

which represents the weighted linear sum of inputs to gene i. α_i is a parameter representing any external inputs that may not be described by the rest of the model. This may be useful when considering only part of a full network yet some parameterisation is required for those interactions not shown or for some external signal that may not be part of the network.

In order to normalise the regulatory input to the $[0, 1]$ scale, the *normalized transcriptional response* is given by application of a sigmoid function

$$g_i(t) = \frac{1}{1 + e^{-r_i(t)}} \tag{2.2.2}$$

Two further parameters are included within the model. The *maximal expression level* for gene i, s_i, is a rate parameter for proliferation of proteins produced by that gene. As biological molecules will decay over time, a *degradation rate parameter* is included as well, d_i.

The neural network model gives the expression level of gene i by the differential equation

$$\frac{du_i}{dt} = s_i g_i(t) - d_i u_i \tag{2.2.3}$$

This system of N differential equations can then be solved to produce expression profiles (gene expression levels over time) and the dynamics observed.

Instead of just modelling expression levels of genes themselves in the cells, the expression levels of RNA can be included to allow for both transcription and translation effects. An example of this setup is included in Tian and Burrage [46]. Here the expression of both mRNA molecules $\boldsymbol{r} = (r_1, ..., r_N)$ and translated proteins (transcription factors) $\boldsymbol{p} = (p_1, ..., p_N)$ are modelled in the same fashion as above with a pair of coupled ODEs

$$\frac{dr_i}{dt} = s_{1i} f(\boldsymbol{p}, \boldsymbol{w_1}) - d_{1i} r_i \tag{2.2.4}$$

$$\frac{dp_i}{dt} = s_{2i} g(\boldsymbol{r}, \boldsymbol{w_2}) - d_{2i} p_i \tag{2.2.5}$$

These models are not currently widely used in practise, due to the difficulty of measuring levels of proteins in practise. As a consequence, further analysis of this class of models is not considered and solely the expression levels based on mRNA is used.

2.3 Stochastic Models for Gene Expression

Biological systems at the molecular level are inherently stochastic, with particles moving in a random fashion and similarly binding and unbinding at target sites on the DNA. The differential equation models previously therefore do not accurately reflect this stochastic nature and so models need to build in random fluctuations within the gene expression levels. The paper by McAdams and Arkin [43] originally introduced this notion which has led to many ways to describe stochasticity in genetic networks

This section looks at some of the ways in which stochasticity can be built into the model. Discretisation of the differential equation model to build up to a stochastic differential equation based approach is developed. A different interpretation is then considered by using the individual numbers of molecules present within the system in the development of stochastic simulation algorithms.

2.3.1 Stochastic Neural Networks

For the differential equation models in equation 2.2.3, a first order discretisation can be written as

$$u_{i,n+1} = u_{i,n} + s_i g_i(t_n) h_n - d_i u_{i,n} h_n \tag{2.3.1}$$

where $u_{i,n} = u_i(t_n)$ and the stepsize is $h_n = t_{n+1} - t_n$.

By introducing fluctuations into the update state, a *stochastic difference model* can be obtained. Three variations can be described as given in Tian and Burrage [46].

- **Poisson Models** $u_{i,n+1} = u_{i,n} + Poi(s_i g_i(t_n) h_n) - Poi(d_i u_{i,n} h_n)$

  where $X \sim Poi(\lambda)$ is sampled from a Poisson distribution with mean and variance λ

- **Exponential Models** $u_{i,n+1} = u_{i,n} + Exp(s_i g_i(t_n) h_n) - Exp(d_i u_{i,n} h_n)$

  where $X \sim \text{Exp}(\lambda)$ has mean λ and can be sampled as $x = -\lambda \, log u$ for $u \sim U(0,1)$

- **Normal models** $u_{i,n+1} = u_{i,n} + s_i g_i(t_n)(h_n + N_{i1}) - d_i u_{i,n}(h_n + N_{i2})$
  where $N_{ik} \sim N(0, h_n \sigma_{ik}^2)$

- **SDE Model** An equivalent formulation of the Normal model is $du_i = (s_i g_i(t) - d_i u_i)dt + \sigma_{i1} s_i g_i(t) dW_{i1} - \sigma_{i2} d_i u_i dW_{i2}$

For each of these approaches, the update step is now no longer deterministic and is allowed to change under some distributional assumption, with the Poisson and Exponential models requiring a single parameter and the Normal model using a mean based change in the step with a variation in the fluctuation from the standard deviation parameter. The SDE model is provided as a limiting case of the Normal model.

2.3.2 Application

To show how these stochastic models compare, consider a simple three gene model where A activates B, B activates A and C, and C represses A, as given in Tian and Burrage [46]. Consideration of the parameters however is required. What is shown is that under certain choice of parameter, a cyclical pattern of expression can be observed, as in Fig 2.2.

This shows that this cyclical pattern expected from the differential equation model is replicated to varying success with the exponential and poisson stochastic difference models, with the poisson model performing well.

A more general example is now given, based on the four node network as given in Figure 2.1 Here there are two input genes which repress one another and self-regulate, and two output genes, which are regulated by one of the input genes

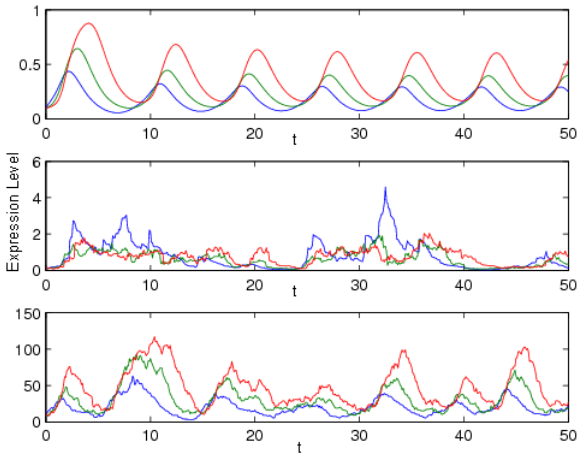

Figure 2.2: A simple 3 gene motif expression pattern under a differtial equation, exponential and poisson model respectively. With the same parameter values (scaled appropriately for the Poisson model) the cyclical pattern is visible for the difference model. The effect of stochasticity becomes visible in the poisson model but the cyclical pattern is still visible. This cannot be said for the exponential model where this noise distorts any detection of the original cycle.

and repressed by the other. In order to compare various properties of the model, the following parameter values are assigned to the model, as used in Equations 2.2.2 and 2.2.3.

$$\boldsymbol{w} = \begin{bmatrix} 10 & -10 & 0 & 0 \\ -10 & 10 & 0 & 0 \\ 10 & -10 & 0 & 0 \\ -10 & 10 & 0 & 0 \end{bmatrix}, \boldsymbol{s} = \begin{bmatrix} 0.7 \\ 0.7 \\ 0.8 \\ 0.8 \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} 0 \\ 0 \\ 4 \\ 5 \end{bmatrix}, \boldsymbol{d} = \begin{bmatrix} 1.0 \\ 1.1 \\ 1.0 \\ 0.8 \end{bmatrix} \tag{2.3.2}$$

where regulatory input may now be written in vector form as $\boldsymbol{r} = \boldsymbol{w}\boldsymbol{u} + \boldsymbol{b}$.

By way of example, the four node network given in Fig 2.1 is used with two input genes targeting two output genes. Figure 2.3 shows the expression levels for the four genes under the deterministic, difference model, exponential and poisson stochastic models. This assumes that genes A and B start fully expressed and genes C and D are not expressed at all initially so that $\boldsymbol{u_0} = [1\ 1\ 0\ 0]$.

These graphs show that noise can affect the dynamics of the modelling of the

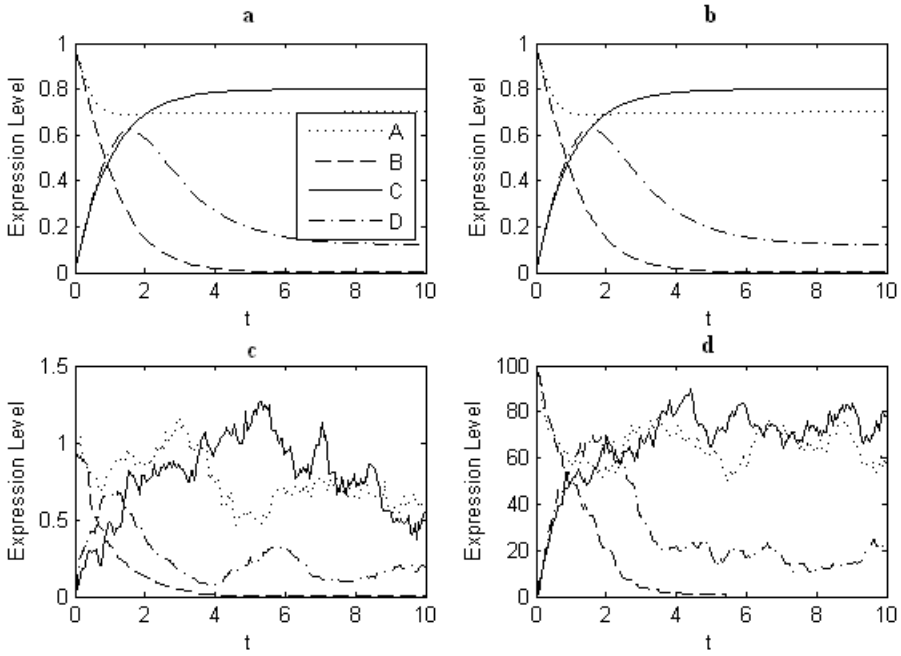

Figure 2.3: For the given 4 gene motif, graphs showing the outputs under the deterministic, difference, exponential and Poisson models respectively.

motif. Under a Poisson model, as the mean and variance are the same, the overall dynamics are comparable to that of the original differential equation model. In the exponential case, where the variance is now the square of the mean, the noise creates a less discernible figure in comparison to that of the other models.

2.3.3 Characterisation of initial conditions

Within the stochastic framework, due to fluctuations occurring, the same initial conditions may lead to different competing states depending on the initial behaviour within the model. Figure 2.4 shows that for the four gene motif with the same set of initial conditions, two different outcomes are shown for the stochastic Poisson model. This is in stark contrast to the deterministic models, where the longer term behaviour is solely dependent on the initial conditions. Consideration therefore needs to be given when using such stochastic models for genetic networks.

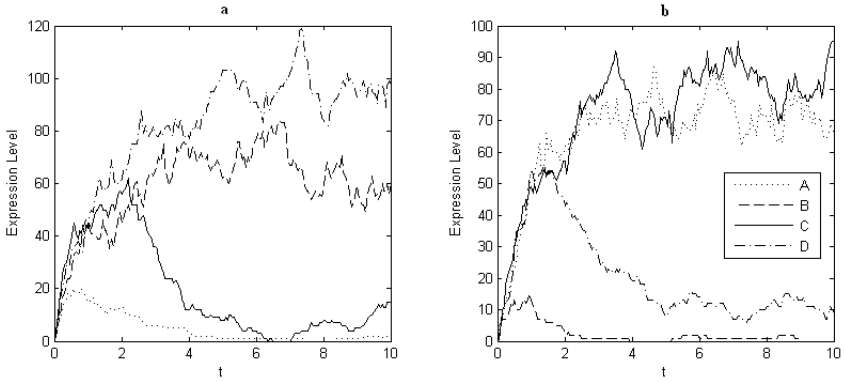

Figure 2.4: For the same initial condition for the two input-two output gene motif, stochastic behaviour in the model may lead to different longer term behaviour. Two simulations from the same initial condition are shown with two different long term states.

2.3.4 Observed Biological Features

In the corresponding biological system cell type is determined by which output gene is more strongly expressed. This binary decision corresponds to the notion of two steady states, where each exhibits greater expression of one of the output genes. This feature of is consistent with Kauffman's hypothesis [1] that an attractor in the Boolean network model corresponds to a particular cell type and helps to validate the model. For the four node motif, Figure 2.5 shows two different steady states depending on the initial conditions of one of the input genes, gene A. Here a critical value, α, is given for the value at which the change in steady state occurs.

Within the stochastic framework, this steady state phenomena is also observed although there may be longer term fluctuations. As in the previous section, the same initial conditions may also give rise to different steady states, as seen in Figure 2.4.

As stem cells differentiate, they become one type of cell from all possible cell types available. This fate can be related to the expression levels of output genes in the steady state of the system after decision via a pathway. These expression levels may appear to be very similar at the same stages of differentiation, and hence it is difficult to uniquely determine the ultimate fate of the stem cell. This similarity in expression at low levels is termed *multi-lineage priming*.

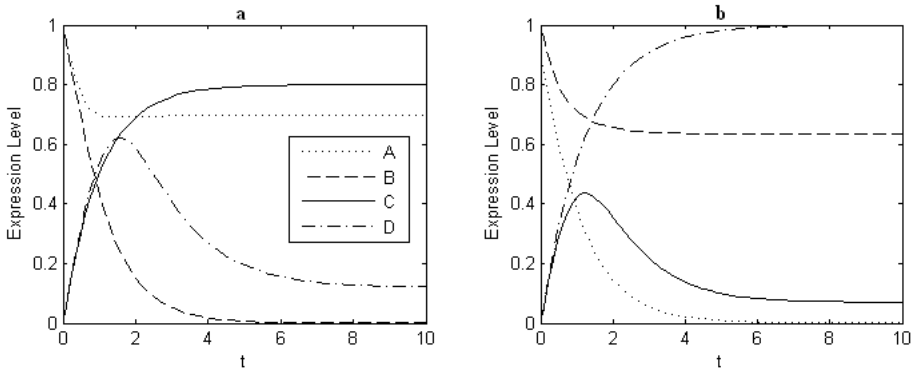

Figure 2.5: Steady state for initial value of gene A above and below critical value $\alpha = 0.976$ respectively, under the deterministic model with initial given parameter values.

This phenomenon can be observed within the deterministic model at short timescales (taking care to note that the timescale is arbitrary depending on choice of parameters). It is observed as the similarity in expression between output genes before a bifurcation occurs and hence a decision on fate is made. Fig 2.6 illustrates an example, in both the deterministic and stochastic setting. In both cases, the levels of the output genes are shown at a similar rate before one wins out against the other.

Roeder & Glauche [29] consider this process for the GATA-1 and PU.1 genes in haematopoietic stem cells, where the priming behaviour is modelled as a bifurcation in a deterministic model. The results show that such a deterministic model suitably provides a measure for this biologically observed feature. One particular limitation within this model is that it is purely deterministic and neglects the inherent stochasticity of the biological system. With the results shown in Figure 2.6, this shows that the stochastic based models can exhibit priming behaviour and so themselves do not lose the biological realism.

Overall, with both the competing dynamics of steady states and long term behaviour of the systems and the short timescale behaviour of multilineage priming, these ODE based models and their stochastic improvements are well suited to modelling genetic networks.

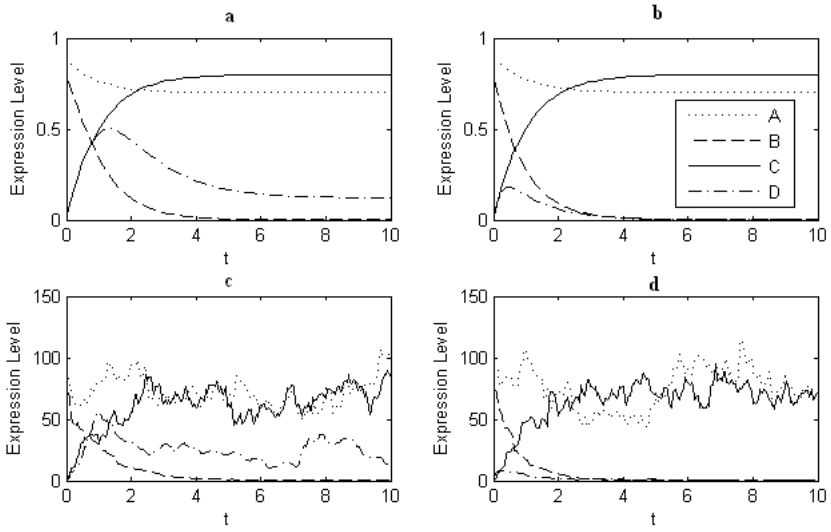

Figure 2.6: Multi-lineage priming behaviour shown for increasing weights in the deterministic model. The output genes are initially fully switched off and can be seen to increase at approximately the same value for a certain time period. The weight of interaction is increased from top to bottom, left to right, and shows the decrease in timespan over which this behaviour occurs.

2.4 Stochastic Simulation Algorithms

The previous section modelled gene expression as a normalised scaling of concentration levels of molecules within a sample. This arbitrary and comparative measure does not take into the individual molecular level modelling within the cell. This section looks at a different method of modelling the amount of molecules in a system by looking at the individual molecule numbers present.

2.4.1 Master Equation

Consider a cell containing a single DNA strand to which both activator and repressor transcription factors can bind. Initially let there be p mRNA molecules of gene A and q mRNA molecules of gene B and each of these can bind and unbind to a pair of promotor sites of a gene G located on the DNA strand. Suppose that gene A activates gene C and gene B repressed gene C. As a simplified model, any other cellular phenomena such as degradation or dimerisation of molecules is disregarded and further let the binding and unbinding be singular

events that do not occur at the same time. The DNA strand can then be in one of four possible states:

- **State 0** No binding.
- **State 1** Activating molecule bound.
- **State 2** Repressing molecule bound.
- **State 3** Both activating and repressing molecules bound.

This relates to the initial Boolean logic models with a set of logic input states for activator A and repressor B (00,10,01,11) respectively, with 0 representing an unbound state and hence the gene being switched off and 1 representing the gene switched on and transcribing mRNA.

The model is parameterised by a reaction coefficient for binding and unbinding of each molecular species. Then over a small time, Δt,the probability of a binding is $k\{\#A\}\Delta t$, where k is the reaction coefficient for binding and $\{\#A\}$ is the number of molecules of type A present in the cell. Similarly, for the unbinding of a molecular species, the probability is $j\Delta t$ with j the reaction coefficient for unbinding.

From this a Markov Chain model can be set up, where the future state depends only on the current state of the system. Let k_{ij} be the reaction coefficient for moving from state i to state j. Over a timestep Δt, where $P(k,t)$ is the probability of being in state k at time t, and supposing there are p molecules of molecular species P and q molecules of molecular species Q then

$$\boldsymbol{P}(t+\Delta t) = \begin{bmatrix} P(0,t+\Delta t) \\ P(1,t+\Delta t) \\ P(2,t+\Delta t) \\ P(3,t+\Delta t) \end{bmatrix} = \boldsymbol{A}\boldsymbol{P}(t) \tag{2.4.1}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 1 - pk_{01}\Delta t - qk_{02}\Delta t & k_{10}\Delta t & k_{20}\Delta t & 0 \\ pk_{01}\Delta t & 1 - k_{10}\Delta t - qk_{13}\Delta t & 0 & k_{31}\Delta t \\ qk_{02}\Delta t & 0 & 1 - k_{20}\Delta t - pk_{23}\Delta t & k_{32}\Delta t \\ 0 & qk_{13}\Delta t & pk_{23}\Delta t & 1 - k_{\Delta}t - k_{32}\Delta t \end{bmatrix}$$

Taking the limit as $\Delta t \to 0$, then

$$\frac{d\boldsymbol{P}}{dt} = lim_{\Delta t \to 0} \frac{\boldsymbol{P}(t + \Delta t) - \boldsymbol{P}(t)}{\Delta t} = \boldsymbol{B}\boldsymbol{P}(t) \tag{2.4.2}$$

where

$$\boldsymbol{B} = \begin{bmatrix} -pk_{01} - qk_{02} & k_{10} & k_{20} & 0 \\ pk_{01} & -k_{10} - qk_{13} & 0 & k_{31} \\ qk_{02} & 0 & -k_{20} - pk_{23} & k_{32} \\ 0 & qk_{13} & pk_{23} & -k_{31} - k_{32} \end{bmatrix} \tag{2.4.3}$$

Here, **B** is a Q-matrix representing a continuous time Markov Chain. This Master Equation, as it is termed, generally cannot be solved explicitly. However, the equilibrium distribution is of considerable interest, which can be shown to exist and is unique, where $\frac{d\boldsymbol{P}}{dt} = 0$ leading to solving $\boldsymbol{B}\boldsymbol{P} = 0$. It is known that $\boldsymbol{B}$ has a zero determinant and, as such, has a zero eigenvalue so the solution for $\boldsymbol{P}(t)$ is the corresponding eigenvector. A full derivation can be found in Bower and Bolouri [18].

2.5 Stochastic Simulation Algorithms

The Master Equation cannot be generally solved algebraically, except for a few simple cases, so simulation methods are required in order to understand the dynamics of larger scale systems. One shortcoming in simulating such large scale systems under the master equation is that the cost of computation can be high when many different molecular species and reactions can occur. In this section, the use of *Stochastic Simulation Algorithms*, and most notably the Gillespie algorithm, give an equivalence to the Master Equation but are less costly to implement.

2.5.1 Gillespie Algorithm

For a well stirred mixture within a finite volume Ω, let there be N chemical species $\{S_1, ..., S_N\}$ reacting with each other by M chemical reactions $\{R_1, ..., R_M\}$. The state of the system at time t is given by the state vector $\boldsymbol{x}(t) = (x_1(t), ..., x_N(t))$ where $x_i(t)$ is the number of molecules of S_i present at time t.

Each reaction R_j is parameterised by two quantities. The *stoichiometric matrix* $\boldsymbol{\nu}$ where ν_{ij} defines the change in species S_i when undergoing reaction R_j, with ν_j a vector representing the change for this reaction across all species S_i. The *propensity function* a_j for reaction j depends on the type of reaction and the rate constants, k_j, for this reaction. For a single reactant, $x_i \rightarrow \textit{products}$ this is of the form $a_j = c_j x_i$. For two reactants, $x_i + x_j \rightarrow \textit{products}$ this is of the form $a_j = c_j x_i x_j$ for $i \neq j$ or of the form $a_j = c_j x_i(x_i - 1)/2$. Similarly this can be extended to multi-species reactions, however, all reactions with multiple species can be broken down into these primary reactions with one or two reactants, further details of which can be found in Gillespie [83].

Now, the master equation for this type of reaction can be expressed as

$$\frac{\partial p(\boldsymbol{x}, t)}{\partial t} = \sum_{j=1}^{M} \left(b_j(\boldsymbol{x} - \nu_j) p(\boldsymbol{x} - \nu_j, t) - b_j(\boldsymbol{x}) p(\boldsymbol{x}, t) \right) \tag{2.5.1}$$

where $p(\boldsymbol{x}, t)$ is the probability of being in state $\boldsymbol{x}$ at time t. This is directly equivalent to the Master Equation from section 2.4.1.

Direct Method

As given in Gillespie [83] a probability distribution for j, the reaction number, and τ, the timestep, is derived from the master equation to give

$$p(\tau, j | x, t) = a_j(x) \exp(-a_0(x)\tau) \tag{2.5.2}$$

where x is the current state vector, t is the current time of the system and $a_0 = \sum_{j=1}^{M} a_j$ for reaction channels $j = 1, \ldots, M$. Then the joint distribution can be factored to a product of distributions $p(\tau, j) = p(\tau)p(j)$ where

$$p(\tau) = a_0(x)\exp(-a_0(x)\tau) \quad ; \quad p(j) = \frac{a_j(x)}{a_0(x)} \tag{2.5.3}$$

Samples from these can be found via inversion. If $u_1, u_2 \sim U(0,1)$ then

$$\tau = \frac{1}{a_0(x)} log\left(\frac{1}{u_1}\right) \tag{2.5.4}$$

and choose j such that

$$\sum_{k=1}^{j-1} a_k(x) \;\leq\; u_2 a_0(x) \;<\; \sum_{k=1}^{j} a_k(x) \tag{2.5.5}$$

Then the Stochastic Simulation Algorithm for this direct method [83] is given by

- Generate initial state vector $x = x_0$ and initial time $t = t_0$.
- Calculate propensity functions $a_j(x)$ and their sum $a_0(x)$.
- Generate samples τ and j.
- Update state vector $x(t)$ by $t + \tau \to t \quad , \quad x + \nu_j \to x$.
- Record (x, t). Return to second step until reach final time, T.

2.5.2 Tau-leaping Method

A couple of important refinements were made to this original algorithm. The First Reaction Method, developed by Gillespie [84], simulates only the timestep τ for each reaction and chooses the smallest of these timesteps. Further to this was the development of the Next Reaction Method by Gibson and Bruck [85]. This method only updates values for reactions that have taken place and keeps all others fixed so that the timestep and reaction do not have to be simulated at every timestep across all reactions and species. It can also be shown to be equivalent to the Direct Method.

All these methods are quite efficient for small scale systems and provide an exact solution to the Master Equation. However, for very large scale systems,

they require a lot of simulation to select the timestep over which an individual reaction occurs. By improving this method to allow for multiple reactions occurring within the same timestep, the computational cost is greatly reduced. This is achieved by an approximation to the Direct Method which is suitable for large systems with many species and many reactions. This *tau-leaping* method [91] simulates timestep, τ, and then calculates the numbers of each reaction that occur within this timestep.

Let τ be small enough so that the propensity functions $a_j(x)$ are approximately constant on $[t, t+\tau)$. Define $k_j(\tau; x, t)$ to be the number of R_j reactions occurring within the time interval $[t, t+\tau)$ given the current state vector x and current time t. Then the change in state vector over this timestep $X(t+\tau)$ is

$$X(t+\tau) = x + \sum_{j=1}^{M} k_j(\tau; x, t)\nu_j \tag{2.5.6}$$

One thing to consider is how to generate the k_j. If the a_j are approximately constant in the time interval considered then they can be modelled as a Poisson random variable $k_j(\tau; x, t) \sim Poi(a_j(x), \tau)$. From this, a sample can be taken from a Poisson random variable with parameter $a_j\tau$ and hence

$$X(t+\tau) = x + \sum_{j=1}^{M} P(a_j(x), \tau)\nu_j \tag{2.5.7}$$

This is all dependent on generating a suitable τ so that the propensity functions don't change significantly over the time period. Currently there exists no superior way to generate this τ and it is chosen by a bounding condition as explained in [91].

Where low count numbers of species are involved, it is often more suitable to use the Direct Method. In practise, for large scale systems, a mixture of this and the tau-leaping method is combined. Similarly, care needs to be taken to ensure that negative amounts of species are not generated within a leap and there exist suitable methods for implementing this.

2.5.3 Example

To illustrate the algorithm in practise we use the following system with three different substances, X,Y and Z. They react according to the reactions

$$R_1 : \quad X + X \xrightarrow{k_1} Y$$

$$R_2 : \quad Y \xrightarrow{k_2} X + X$$

$$R_3 : \quad Y \xrightarrow{k_3} Z$$

with parameter values $k_1 = 0.002$, $k_2 = 0.16$ and $k_3 = 0.07$ and initial numbers of molecules $X = 1000$, $Y = 1$ and $Z = 0$. Figure 2.7 shows a realisation of this example under both the Direct Method and the tau-leaping method. It can be seen that the results are the same for both the direct method and the tau-leaping method, yet the time taken to execute the algorithm was 2.35 seconds under the direct method but only 1.05 seconds under the tau-leaping method. Although this difference is small, for much larger scale systems this speed up is significant.

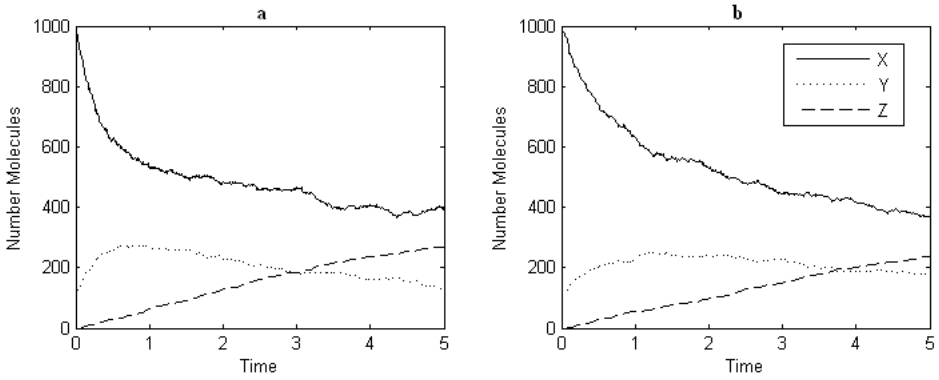

Figure 2.7: A realisation of the Gillespie algorithm with parameter values $k_1 = 0.002$,$k_2 = 0.16$ and $k_3 = 0.07$. On the left is the direct method and the right shows the tau-leaping method.

2.5.4 Application to Genetic Networks

Stochastic Simulation Algorithms are widely used to model chemically reaction systems where reactions and species are known. They have been extended to other interacting systems, such as genetic networks. As an example, for some transcription factor TF undergoing a reaction at a promoter site on some strand of DNA DNA, then genetic properties can be characterised as such:

- **Binding** : $TF + DNA \rightarrow TF \cdot DNA$
- **Unbinding** : $TF \cdot DNA \rightarrow TF + DNA$
- **Degradation** : $TF \rightarrow \emptyset$
- **Dimerization** : $TF + TF \rightarrow TF_2$

In a genetic system, typically the number of transcription factors will be in the orders of tens or hundreds, potentially even thousands. Clearly, to model these in a differential equation framework would require a great amount of computational power. Stochastic simulation algorithms, especially where good approximations can be used such as tau-leaping, therefore provide the means to greatly reduce this computational cost yet still providing an insight into these large scale systems.

For further applications of this modelling methodology, Wilkinson and Boys [82] include transcription and translation each as reactions, and a good overview of the whole modelling concept is provided in the book by Wilkinson [219]. A real world example of the well studied Lambda phage system is given by Gibson [70].

CHAPTER 3

Network Inference

In the previous chapter, the aim was to build a model of a genetic network and how to make it biologically plausible, such as by the addition of stochasticity. This is useful for generating data for simulation purposes and to understand the dynamics of such known networks. This chapter, and the subsequent chapters, look at the alternate view, by using this data sampled over time in order to reconstruct the underlying network such data came from.

For artificial networks, the comparison can be made as to how well the network is reconstructed and comparison measurements can be made. This is not quite so easy for real data, where the existing networks may only be partially known. Here the aim is to then predict which interactions are most likely to occur in order that the biological experiments can be performed to verify this and new interactions discovered.

3.1 Genetic Network Inference Models

Existing models can be split into two categories; directional networks and correlation based networks. These differ in the ability to reconstruct the direction of interaction between networks but both offer different perspectives on how genes are linked together to form overall networks. Directional networks, such as Bayesian networks, are more realistic due the known directional nature of observed genetic networks, where a gene has an influence on another yet this may not be true the other way round. Correlation based models look at whether there is similarity between the nodes to state whether there is some interactional

effect between them.

3.1.1 Bayesian Networks

A *Bayesian Network* is a directed graph whereby a series of nodes are connected by a series of direct interactions to each other and where such interactions may be parameterised in some way. Where these nodes are represented by a sequence of variables, such as a time series, these give rise to *Dynamic Bayesian networks*. An excellent overview of Bayesian Networks is given in the book by Pearl [220], with the application of dynamic Bayesian networks to genetic network recovery presented in Spirtes et al. [142] and Werhli et al. [154].

Inference on a given node is given by knowledge of only the values of its parents nodes i.e. the nodes leading directly into it. Then a distribution for the network can be inferred via the conditional independence of each of the nodes and use of Bayes' Theorem. For example, let $\boldsymbol{X}(t) = (X_1(t), ..., X_N(t))$ represent parameterised values of the nodes, corresponding to expression levels in the genetic sense. Then if $Pa(X_i(t))$ represents the "parents" of node i at time t i.e. only the nodes directed *into* gene i then the update is

$$P(\boldsymbol{X}(t)|\boldsymbol{X}(t-1)) = \prod_{i=1}^{N} P(X_i(t)|Pa(X_i(t-1)))$$

3.1.2 Correlation Based Models

By assessing the correlation between two nodes of a network, a measure of similarity can be inferred which would signify an interactional effect between them even if the direction of such interaction may not be known. Soranzo et al. [221] consider the use of Pearson and partial Pearson correlations between time series data to see whether any such interactions are to be found. By using a partial correlation measure, the effect of the presence of other variables can be measured against the Pearson correlation.

One particular use of correlation, and in particular partial correlation, measures is the analysis performed by Schafer and Strimmer [187]. Here, data is used at a steady state, so for a single time point. However, repeated measurements

at the same gene are made in order to calculate the correlations between each gene. This approach does not take into account the temporal nature of gene networks evolving but it does consider the stochasticity by using such repeated measurements.

3.1.3 Single timepoint, multiple samples

For the analysis of repeated samples at a single timepoint, the approach considered is taken from the paper by Schafer and Strimmer [187].

Using a breast cancer dataset X consisting of $G = 3883$ genes with $N = 49$ samples taken, three estimates of the partial correlation matrix are constructed :

$\hat{\Pi}_1$: Use the pseudoinverse to generate an estimate of the partial correlation

$\hat{\Pi}_2$: Estimate P by applying bootstrapping, then use the pseudoinverse to get an estimate of the partial correlation

$\hat{\Pi}_3$: Use the pseudoinverse on each bootstrap replicate of P, and then average the results.

Given a partial correlation matrix, we seek to find which of these is significant enough to represent a direct interaction between two genes. This can be addressed as a hypothesis testing problem

$$H_0 \ : \ \pi_{ij} = 0 \ \text{ vs } \ H_1 \ : \ \pi_{ij} \neq 0$$

The distribution of $p = \hat{\pi}$ under the null distribution can be shown [222] to be

$$f_0(p, \kappa) = (1 - p^2)^{(\kappa-3)/2} \frac{\Gamma(\kappa/2)}{\pi^{1/2}\Gamma((\kappa-1)/2)}$$

However, the degree of freedom κ is required to be positive but is given as $\kappa = N - G + 1$ so with the number of genes much greater than the number of samples, κ has to be estimated from the estimated partial correlation matrix. To estimate this we utilise the assumption that typically a genetic network will be sparse.

Let η_0 be the proportion of genes under the null distribution and η_1 the proportion of genes under the alternative distribution, with $\eta_0 \gg \eta_1$ and $\eta_0 + \eta_1 = 1$. Then the form of the distribution of the partial correlation coefficients is

$$f(p) = \eta_0 f_0(p, \kappa) + \eta_1 f_1(p) \tag{3.1.1}$$

where f_1 is the alternative distribution, set for simplicity as a Uniform distribution on $[-1, 1]$.

This form of problem is naturally solved by the Expectation-Maximisation (EM) algorithm. This uses iterative expectation and maximisation steps that converge to estimates for the parameters, in this case $\boldsymbol{\theta} = (\kappa, \eta_0)$.

From these parameters we can form the distribution of samples to identify the p-value associated with a statistically significant interaction.

3.1.4 Example

Arbitrarily taking the first 100 genes of the breast cancer data set, implementing the algorithm above with partial correlation matrix $\hat{\Pi}_1$ detects $\eta_0 = 0.99923$ and $\kappa = 22.066$. This leads to a p-value cutoff level for each interaction of 0.48 leading to 282 significant interactions. A subset of these genes is shown in Figure 3.1. It is interesting to note that not all genes are shown to link to each other and that there exist pockets of genes with no interactions shown.

3.2 Discussion of Existing Models

When attempting to recover genetic networks, the biological model needs to be taken into account and the salient features that should be reproduced in order to validate the biological realism of the model. The most important features are the stochastic nature of the binding and unbinding mechanism to alter the rate of transcription, and the directional nature of the interactions between genes. The stochastic nature can easily be associated with data taken at regular time intervals, whereby the measurements are not known continuously. The limitation of the ability to measure such data must also be taken into account, such as measurement error.

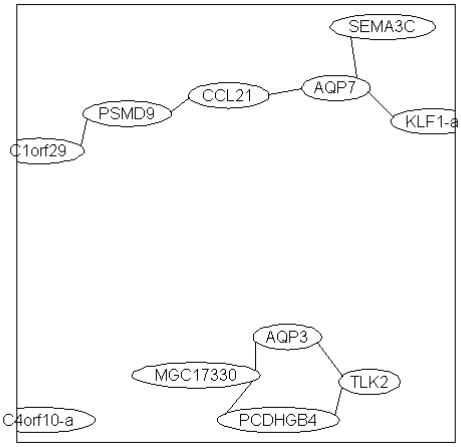

Figure 3.1: Subset shown for the breast cancer network recovered by Schafer and Strimmer [187].

The correlation measures, such as those explored by Schafer and Strimmer, are useful up to a point to understand functional relationship between genes. The use of repeated measures allows the stochastic nature of the system to be considered but this is limited information at a single steady state timepoint. Hence, the truly stochastic nature of a system over time is not taken into account. The use of a correlation based measure does not allow for the directionality of the gene interactions and so also provides limited information. As such, this type of model fails to fully address the issues addressed in terms of fully reconstructing a genetic network.

Bayesian Networks, however, do take into account both of these desired features. By constructing a probability measure of the state of a network depending on the interactions between them, the overall stochasticity of the system is coupled with the directional interaction between genes, as a change in directionality between two nodes would give a different probability distribution over time for the state of the system. This explains why Bayesian network models are prevelant within the area of reconstructing genetic networks as they relate soundly to the biological system being modelled.

CHAPTER 4

Granger Causality for Recovering Genetic Networks

The previous chapter looked at some of the existing ways in which genetic networks can be inferred from microarray data. The approach set out in this chapter is to consider how a causal link may be inferred from such data, so as to say whether there is a statistically significant effect of one gene upon another. Clearly one benefit is the directedness of the approach, whereby gene A may target gene B but vice versa this may not be true.

By using time series data as well, the temporal nature of change within a system can be used to understand whether a change over time in gene A causes a change in gene B, and use this as a means of finding an interaction.

This chapter introduces Granger causality, which provides a statistical measure of how much effect one timeseries has on another by considering whether a model with a variable regressed on itself is different to a model with other variables added. This uses both the temporal and directional nature required to reconstruct genetic networks.

4.1 Time Series

Given that the data used in Granger causality is based on time series, here an introduction to time series modelling is given. Time series analysis has grown into a huge area of research with many applications for prediction and estimation where data is given at time samples for single or multiple variables. For a

wide ranging discussion of time series, both in univariate and multivariate settings, the books by Hannan [202], Lutkepohl [203] and Priestley [204] all give very useful overviews to the area.

4.1.1 Notation

Let X_t be a random variable indexed by a time variable t. Whilst t may be negative or continuous, our focus shall be on time data sampled discretely and indexed by $t = 1, ..., T$ for some T. A *univariate time series* is then $X = \{X_1, \ldots, X_t\}$.

The multivariate equivalent is $\boldsymbol{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_t\}$ for m time series, where the vector of time series at timepoint i is $\boldsymbol{X}_i = \{X_{i1}, \ldots, X_{im}\}$.

One key property that is used in causality detection, particularly with Granger causality, is stationarity of data. Stationarity enforces certain structure of the data.

A time series X is said to be *(weakly) stationary* if $\mathrm{E}(X_t)$ and $\mathrm{Cov}(X_t X_{t+h})$ are independent of t for all h.

When considering stationarity it will be assumed in the context of weak stationarity which is less restrictive. For autoregressive processes, a simple test of stationarity is that all roots of the characteristic equation lie outside the unit circle.

4.1.2 VARMA Models

There are many types of time series model that can be used to describe data. One of these general classes of model are VARMA models - Vector AutoRegressive Moving Average. Autoregression in a time series describes the dependence of data at previous timepoints whereas moving averages represent noise terms in the model.

The VARMA(p,q) model is given by

$$\boldsymbol{X}_t = \boldsymbol{\mu} + \boldsymbol{\Phi}^1 \boldsymbol{X}_{t-1} + ... + \boldsymbol{\Phi}^p \boldsymbol{X}_{t-p} + \boldsymbol{\varepsilon}_t - \boldsymbol{\Theta}^1 \boldsymbol{\varepsilon}_{t-1} - ... - \boldsymbol{\Theta}^q \boldsymbol{\varepsilon}_{t-q} \tag{4.1.1}$$

where the noise term is

$$\boldsymbol{\varepsilon}_t \sim N(\mathbf{0}, \boldsymbol{\Sigma}) \tag{4.1.2}$$

$\boldsymbol{\Phi}^1, \ldots, \boldsymbol{\Phi}^p, \boldsymbol{\Theta}^1, \ldots, \boldsymbol{\Theta}^q$ are parameter matrices and $\boldsymbol{\mu}$ is a vector of intercept terms.

4.1.3 Estimation of Parameters

Estimation of parameters where $q \geq 1$ is challenging due to the requirement to estimate noise in the model. By restricting $q = 0$, this reduces the VARMA models to VAR(p) models which are more easily estimated. Similarly, setting $\mu = 0$ removes the need to estimate the mean of the model. These will be used mostly in the following analyses but the results naturally extended to VARMA models with appropriate estimation of parameters.

Given a time series of observed data, $\boldsymbol{X}$, the model parameters can be fitted to this data depending on the type of model used. Whilst there are many techniques available covered in most books on Time Series such as in Lutkepohl [203], the subsequent analysis shall be focus on VAR(p) models and hence parameter estimation under such models is considered. One popular method is to obtain a series of recurrence relations termed the Yule-Walker equations [223].

Assuming a (weakly) stationary VAR(p) process

$$\boldsymbol{X}_t = \boldsymbol{\Phi}^1 \boldsymbol{X}_{t-1} + ... + \boldsymbol{\Phi}^p \boldsymbol{X}_{t-p} + \boldsymbol{\varepsilon}_t \tag{4.1.3}$$

with $\boldsymbol{\varepsilon}_t \sim N(\mathbf{0}, \boldsymbol{\Sigma})$ we can calculate the Yule-Walker equations by multiplying each side by $\boldsymbol{X}_{t-j}$, $j = 0, ..., p$ and taking expectations. Defining $\Gamma(i)$ as the covariance at lag i, $\Gamma(i) = \text{Cov}(\boldsymbol{X}_t, \boldsymbol{X}_{t-i})$ and noting that $\Gamma(i) = \Gamma(-i)$

$$\mathbf{\Sigma} = \Gamma(0) - \sum_{j=1}^{p} \mathbf{\Phi}^j \Gamma(j) \tag{4.1.4}$$

$$\Gamma(i) = \sum_{j=1}^{p} \mathbf{\Phi}^j \Gamma(i-j) \tag{4.1.5}$$

Estimates for the $\mathbf{\Phi}^j$ can be obtained by the Whittle algorithm [224], a multivariate extension of the Durbin-Levinson algorithm [225], and point estimate of $\Gamma(i)$

$$\widehat{\Gamma}(i) = \widehat{\text{Cov}}(\boldsymbol{X}_t, \boldsymbol{X}_{t-i}) = \frac{1}{T}(\boldsymbol{X}_t - \overline{\boldsymbol{X}})'(\boldsymbol{X}_{t-i} - \overline{\boldsymbol{X}}) \tag{4.1.6}$$

The Yule-Walker equations provide a least squares estimation of the data for a VARMA model but there are many other estimation techniques.

4.2 Model Selection

In order to appropriately fit a time series model, such as the VAR(p) model, the order of the model needs to be first assigned. Care needs to be taken when choosing this order of fit. If the order is too great, there may be no real gain in information and more parameters will be estimated needlessly. If the order is too small, the model may not contain enough information and hence not accurately capture the structure of the data giving rise to performance errors.

One solution is to choose the order of the model in a statistical manner by using *Information Criteria*. Here, multiple models are fitted over a range of orders and larger models penalised unless there is enough extra information to use them.

Let $\boldsymbol{X}$ be modelled with a VAR(p)

$$\boldsymbol{X}_t = \mathbf{\Phi}^1 \boldsymbol{X}_{t-1} + ... + \mathbf{\Phi}^p \boldsymbol{X}_{t-p} + \boldsymbol{\varepsilon}_t \tag{4.2.1}$$

where $\boldsymbol{X}$ is a K-variate time series with n timepoints and m the number of parameters used to estimate the criterion. The value of p chosen to estimate the

Information Criterion	Value
Akaike (AIC) [205]	$log\|\hat{\Sigma}_\varepsilon\| + \frac{2pK^2}{T}$
Schwarz Bayesian (BIC) [206]	$log\|\hat{\Sigma}_\varepsilon\| + \frac{log(n)pK^2}{T}$
Hannan-Quinn [202]	$log\|\hat{\Sigma}_\varepsilon\| + \frac{2log(log(n))pK^2}{T}$

Table 4.1: List of Information Criteria

model is the value of m which minimises the criterion.

4.2.1 Information Criteria

Information criteria seek to find a suitable model whereby a model with an excessive number of parameters is penalised. Data is fitted to the model and the residuals used to calculate how well the model fits. This is then penalised depending on the number of parameters in the model and other information such as the dimension of the data. The optimal model chosen is that which minimises the information criterion chosen.

For a multivariate time series $\boldsymbol{X}$, let K x T be the dimension of the time series data (with K time series each of length T) and let $\hat{\Sigma}_\varepsilon$ be an estimate of the Covariance matrix of the residuals for the fitted model with order p. Then three widely used information critera are given in Table 4.1.

Although these information criteria provide a mechanical way of selecting an appropriate model, care still needs to be taken to ensure the model order is appropriate. A high model order may be chosen as it has the minimal value of the information criterion, yet a smaller model may be neglected even though the information criterion may be close to this value. In order to prevent such spurious choices of model, it can be sometimes beneficial to restrict the order of the model.

In Kadilar and Edemir [207], the use of information criteria is explored specifically in the case of VAR models. Here, the Bayesian Information Criterion is shown to have optimal performance above other widely used information criteria, including the AIC. As subsequent analysis will use VAR models, it is

decided that the BIC will be used for model selection.

4.3 Granger Causality

In simple terms, causality is the effect of one thing on another. As genes are able to directly affect other genes, such as by transcription factors molecules of one gene binding to promoter sites of another gene and altering the rate of transcription, they can be described as having a causal effect on each other. This causality may be direct or indirect. The nature of causality means it is directional and hence suitable for application in recovery of gene networks. Gene A may exhibit some causal effect on gene B, but this doesn't always apply the other way round.

Granger causality is a statistical technique that can be used to give a numerical way of asserting whether one time series has an effect upon another. By looking at two competing models, one where a time series is autoregressed upon itself and another where a time series is autoregressed upon itself and also another time series, the significant difference of the two models would imply that the presence of the extra information from another time series has had some effect on the original time series.

4.3.1 Granger Causality

Granger [214] developed the statistical technique of Granger causality in econometrics to measure the effect of economic variables upon each other, such as whether change in price caused a change in demand or vice versa. The time series are assumed to be either stationary or co-integrated, where a linear superposition of the time series is stationary.

Suppose X^1 and X^2 are two (univariate) time series. By defining the model

$$X_t^1 = \alpha_1 X_{t-1}^1 + ... + \alpha_p X_{t-p}^1 + \beta_1 X_{t-1}^2 + ... + \beta_q X_{t-p}^2 + u_t \tag{4.3.1}$$

then X^2 is said to *Granger cause* X^1 if $\beta_i \neq 0$ for at least one of the i. This can be tested by means of an F-test with the null hypothesis $H_0 : \beta_1 = \beta_2 = ... = \beta_p =$

0.

By estimating the parameters $\hat{\alpha}_i, \hat{\beta}_i$, such as by the Yule-Walker equations in section 4.1.3, the residual sum of squares can be calculated for the fitted model $RSS_1 = \sum_{t=1}^{T} \hat{u}_t^2$, where

$$\hat{u}_t = X_t^1 - \hat{\alpha}_1 X_{t-1}^1 - ... - \hat{\alpha}_p X_{t-q}^1 - \hat{\beta}_1 X_{t-1}^2 - ... - \hat{\beta}_p X_{t-p}^2 \tag{4.3.2}$$

An autoregressive model of the same order p to the series X^1 alone.

$$X_t^1 = \gamma_1 X_{t-1}^1 + ... + \gamma_p X_{t-p}^1 + e_t \tag{4.3.3}$$

Similarly, residual sum of squares for this fitted model is $RSS_0 = \sum_{t=1}^{T} \hat{e}_t^2$.

The test statistic is then

$$S_1 = \frac{(RSS_0 - RSS_1)/p}{RSS_1/(T - 2p - 1)} \tag{4.3.4}$$

and the null hypothesis H_0 is rejected at the α level if $S_1 > F(p, T - 2p - 1)_\alpha$.

An equivalent asymptotic test for large T is to use the statistic

$$S_2 = \frac{T(RSS_0 - RSS_1)}{RSS_1} \tag{4.3.5}$$

and to reject if $S_2 > \chi^2(p)_\alpha$.

4.3.2 Algorithm 4.1 - Bivariate Granger Causality

The Granger causality model can be related to a bivariate VAR(p) model, or VAR(2) model, for pairwise testing whether one time series has an effect on one other time seres and vice versa. For two time series X_t and Y_t, the VAR(2) model is

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} \phi_{11}^1 & \phi_{12}^1 \\ \phi_{21}^1 & \phi_{22}^1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \end{bmatrix} + \cdots + \begin{bmatrix} \phi_{11}^p & \phi_{12}^p \\ \phi_{21}^p & \phi_{22}^p \end{bmatrix} \begin{bmatrix} x_{t-p} \\ y_{t-p} \end{bmatrix} + \begin{bmatrix} \varepsilon_t^x \\ \varepsilon_t^y \end{bmatrix}$$

So y does not Granger cause x if $\phi_{12}^k = 0$ for $k = 1, ..., p$.

For analysis from time series data, a hypothesis test can be formed to decide whether there is statistically significant evidence of causality

$$H_0 : x \overset{G}{\not\to} y \quad vs \quad H_1 : x \overset{G}{\to} y \tag{4.3.6}$$

4.4 Bootstrapping Time Series

One of the assumptions used in assessing Granger causality is that the data is stationary or co-integrated. Whilst there are various tests of this, typical genetic microarray data can be short in which case it may be difficult to quantify whether this assumption holds true. One way to assess how the variation of parameters in the models occurs in these cases is to use *bootstrapping*.

Bootstrapping is a resampling technique to assess the variability in the estimation of parameters. Bootstrapping was developed by Efron [215] as a means of obtaining statistics of interest, such as the mean or variance, when the distribution of the true population from which the sample is taken may not be explicitly known. This is the case where the sample size is very limited.

4.4.1 Bootstrapping

Given an n-vector of observations $\boldsymbol{x} = (x_1, x_2, ..., x_n)$, a bootstrap sample is $\boldsymbol{x^*} = (x_1^*, x_2^*, ..., x_n^*)$ where x_i^* is taken with replacement from the original observations. For example of a boostrap sample from a vector of 7 observations could be $\boldsymbol{x^*} = (x_3, x_5, x_2, x_3, x_7, x_5, x_1)$.

Given B bootstrap samples, $\boldsymbol{x^{*1}}$,$\boldsymbol{x^{*2}}$,...,$\boldsymbol{x^{*B}}$, the statistics of interest for each of

these samples are $\theta^{*1}, \theta^{*2}, ..., \theta^{*B}$. The bootstrap estimate of the statistic of interest is then the mean of these bootstrap samples

$$\hat{\theta}_{bootstrap} = \frac{1}{B}\sum_{i=1}^{B}\theta^{*i}$$

Similarly, the variance of this bootstrap estimate may be calculated as

$$\hat{\sigma}^2_{bootstrap} = \frac{1}{B}\sum_{i=1}^{B}(\theta^{*i} - \hat{\theta}_{bootstrap})^2$$

4.4.2 Bootstrapping applied to Time Series

In the context of time series, the observed sample is the time series itself. Using this sample, bootstrap time series can be created from resampling the observed time series and then the bootstrap parameter estimates obtained for these models.

There are two main types of bootstrapping time series: residual based bootstrap and block based bootstrap. The residual bootstrap resamples from the residuals of the fitted model and adds these to the observed time series to generate the bootstrap time series. Block bootstrapping divides the time series into blocks, either overlapping or not, and rejoins these blocks in order to create the bootstrap time series.

4.4.3 Non-Overlapping Block bootstrap

For sufficiently long time series, the non-overlapping bootstrap can be used to estimate the parameters of interest but also the variability of such parameter estimates.

Select the block length l and hence the number of blocks k such that $kl \geq n$ such that blocks are $[x_1, \ldots, x_l]$, $[x_{l+1}, \ldots, x_{2l}]$ $\ldots$ $[x_{k(l-1)+1}, \ldots, x_{kl}]$. For convenience, relabel the blocks $Y_1, ..., Y_k$.

Then for each bootstrap replicate $b = 1, \ldots, B$,

1. For $i = 1, \ldots, k$, sample a block index α_i uniformly from $1, \ldots, k$.

2. The bth bootstrap time series is then $X^b = Y_{\alpha_1} \ldots Y_{\alpha_k}$.

3. If $kl > n$, remove any extra timepoints so that X^b is of length n.

4. The bootstrap estimate of the parameters is obtained from this time series, $\hat{\phi}_1^b, \ldots, \hat{\phi}_p^b$.

4.4.4 Overlapping block bootstrap

Select the block length l, the overlap shift d and the number of blocks k such that $k\,l \geq n$ such that blocks are

$[x_1, \ldots, x_l]$, $[x_{1+d}, \ldots, x_{l+d}]$, ... , $[x_{1+md}, \ldots, x_{l+md}]$

, ... , $[x_{n-l+1-d}, \ldots, x_{n-d}]$, $[x_{n-l+1}, \ldots, x_n]$.

For convenience, relabel the blocks $Y_1, ..., Y_{k'}$, where k' will vary depending on the value of d as more blocks will be generated than in the non-overlapping block case.

Then for each bootstrap replicate $b = 1, \ldots, B$,

1. For $i = 1, \ldots, k$, sample a block index α_i uniformly from $1, \ldots, k'$.

2. The bth bootstrap time series is then $X^b = Y_{\alpha_1} \ldots Y_{\alpha_k}$.

3. If $kl > n$, remove any extra timepoints so that X^b is of length n.

4. The bootstrap estimate of the parameters is obtained from this time series, $\hat{\phi}_1^b, \ldots, \hat{\phi}_p^b$.

Where the overlap shift $d = 1$, this represents where where all possible overlapping blocks are generated and is referred to as the fully overlapping block bootstrap. Furthermore, blocks of fixed length can be selected at random such that the overlap shift is non-constant but this is not considered here.

4.4.5 Residual bootstrap

The residual bootstrap differs from the approach for the block bootstrap, whereby the bootstrap is applied to the residuals of an already fitted model, as opposed to fitting a model to bootstrapped data straight away.

1. Fit an appropriate AR(p) model to the true data X_t.

2. Estimate parameters for the model $\hat{\phi_1}, \ldots, \hat{\phi_p}$ from X_t.

3. Fit the estimated parameters to the model and obtain the residuals r_i for $i = p+1, \ldots, n$.

4. Obtain the centred residuals $\tilde{r}_i = r_i - \bar{r}$ for $i = p+1, \ldots, n$.

5. Obtain a bootstrap series of residuals sampled from the centred residuals, $\tilde{r}_i^B$

6. Obtain a bootstrap time series by recursion $X_t^b = \hat{\phi_1} X_{t-1}^b + \ldots + \hat{\phi_p} X_{t-p}^b + \tilde{r}_t^B$.

7. Re-estimate the parameters from this time series to obtained the bth bootstrap sample of the parameters $\hat{\phi}_1^b, \ldots, \hat{\phi}_p^b$.

4.4.6 Example

An example for an AR(2) model is given. The true underlying model is given by $X_t = 0.5X_{t-1} - 0.3X_{t-2} + \varepsilon_t$ to give $\phi_1 = 0.5$ and $\phi_2 = -0.3$. A time series of 250 timepoints is generated according to this model to give an observed dataset from which to estimate the parameters. Figures 4.1 - 4.3 show the histograms of $\hat{\phi_1}$ and $\hat{\phi_2}$ for 2000 bootstraps under the non-overlapping block, fully overlapping block and residual bootstraps respectively, with a blocksize of 10 for the block bootstraps.

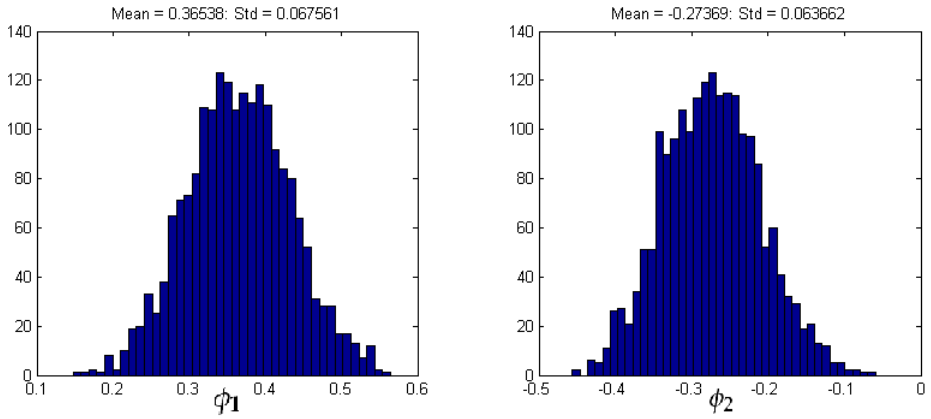

Figure 4.1: Non overlapping block bootstrap with 2000 bootstrap replicates showing bootstrap mean and standard deviation estimates.

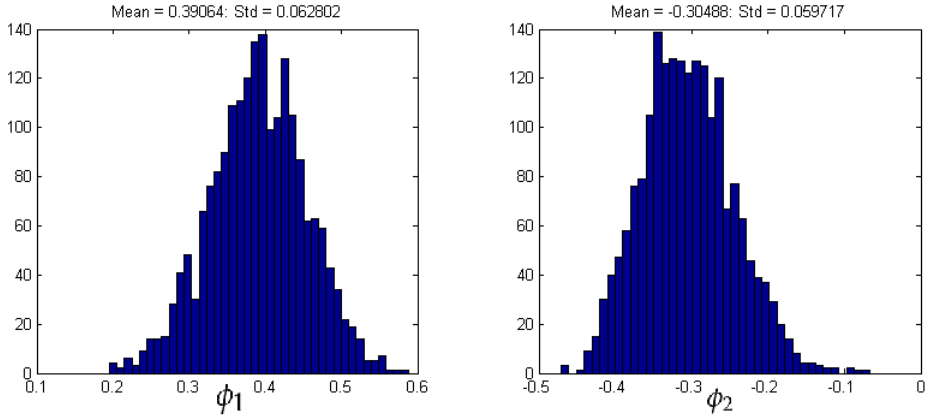

Figure 4.2: Fully overlapping block bootstrap with 2000 bootstrap replicates showing bootstrap mean and standard deviation estimates.

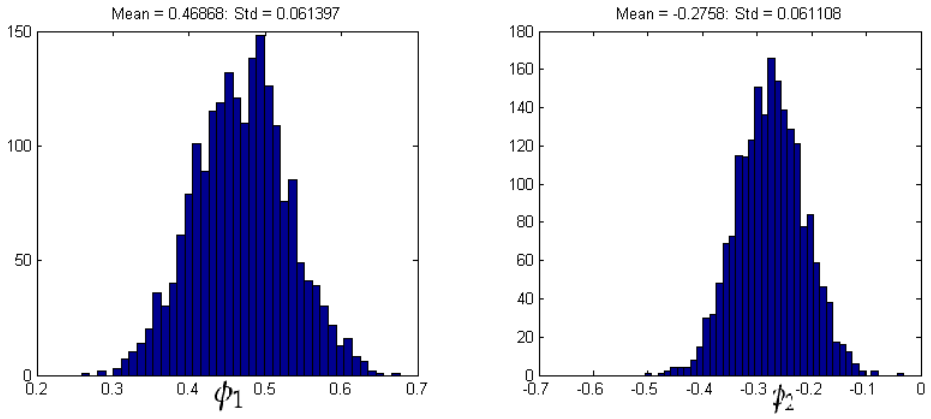

Figure 4.3: Residual bootstrap with 2000 bootstrap replicates showing bootstrap mean and standard deviation estimates.

In order to contrast the performance of the overlapping block bootstrap, Table 4.2 shows the estimate of the parameters and their standard deviation for all possible shifts, as in Section 4.4.4, for $d = 2, \ldots, 9$, along with $d = 1$ (fully overlapping block bootstrap) and $d = 10$ (non-overlapping block bootstrap).

Table 4.2: Performance of the overlapping block bootstrap for all possible shifts from Example 4.4.6 with $\phi_1 = 0.5$ and $\phi_2 = -0.3$.

d	$\hat{\phi_1}$	$sd(\hat{\phi_1})$	$\hat{\phi_2}$	$sd(\hat{\phi_2})$
1 (Full overlap)	0.39064	0.062802	-0.30488	0.059717
2	0.39055	0.063584	-0.29753	0.060148
3	0.38077	0.062986	-0.30048	0.061426
4	0.37044	0.064057	-0.28747	0.062447
5	0.38265	0.065018	-0.29581	0.061843
6	0.38924	0.064756	-0.30542	0.059964
7	0.36857	0.066225	-0.28540	0.062483
8	0.37849	0.065843	-0.29445	0.063415
9	0.37004	0.068027	-0.28382	0.062931
10 (Non-overlap)	0.36538	0.067561	-0.27369	0.063662
Residual	0.46868	0.061397	-0.2758	0.061108

Table 4.3 shows the effect of keeping the overlap fixed, in this instance using the fully overlapping bootstrap, but altering the blocksize used to gain estimates of the parameters. This shows the blocksize indeed has an effect on the ability to estimate the parameters of the model. The best estimate is obtained when the blocksize is 20, however, there is no clear trend in these estimates as the blocksize changes. The standard deviation, on the other hand, has a more noticeable trend where it declines as the blocksize increases. Overall, however, the results are not significantly different from each other. Further discussion of optimal block length is discussed by Buhlmann et al [226].

Overall, the residual bootstrap performs best in terms of the estimate of the parameters. The overlapping block bootstrap generally performs better than the non-overlapping bootstrap. Overall, however, there is little difference between

Table 4.3: Performance of the fully overlapping block bootstrap for all varying blocksize from Example 4.4.6 with $\phi_1 = 0.5$ and $\phi_2 = -0.3$.

Blocksize	$\hat{\phi_1}$	$sd(\hat{\phi_1})$	$\hat{\phi_2}$	$sd(\hat{\phi_2})$
5	0.39414	0.061005	-0.29443	0.063157
8	0.39287	0.063572	-0.31435	0.061082
10	0.39064	0.062802	-0.30488	0.059717
12	0.40371	0.060437	-0.30682	0.060359
15	0.39832	0.059476	-0.31523	0.059114
20	0. 42461	0.058361	-0.32553	0.059327
25	0.39429	0.054190	-0.31874	0.057364
50	0.38727	0.057494	-0.29631	0.058367

the block bootstraps with the variability of the bootstrap estimates reasonably similar and mean estimates further away from the true value than in the residual bootstrap, as shown in Table 4.2. Furthermore, there is evidence that block size will also have an effect on the ability to recover parameter estimates, as shown in Table 4.3. Again, this is generally less effective than the use of the residual bootstrap.

One further consideration is that using the block bootstrap adds extra parameters for the choice of blocklength and also for the choice of overlap. The residual bootstrap removes the need for these extra parameters. Coupled with better performance, the residual bootstrap is favoured over the block bootstraps, as discussed by Buhlmann [227].

4.4.7 Multivariate Time Series Bootstraps

The bootstrapping methods for time series extend naturally into the multivariate setting, with no difference required in the setup of the algorithm. Figure 4.4 shows two time series generated by the VAR(1) model

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} 0.5 & -0.1 \\ 0.2 & -0.3 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_t^x \\ \varepsilon_t^y \end{bmatrix}$$

with $\phi_1 = 0.5, \phi_2 = -0.1, \phi_3 = 0.2$ and $\phi_4 = -0.3$. Figures 4.4 - 4.6 show the parameter estimates for the non-overlapping block and overlapping block with a blocksize of 10 used. As with the univariate example, the non-overlapping block bootstrap works less well than for the overlapping block.

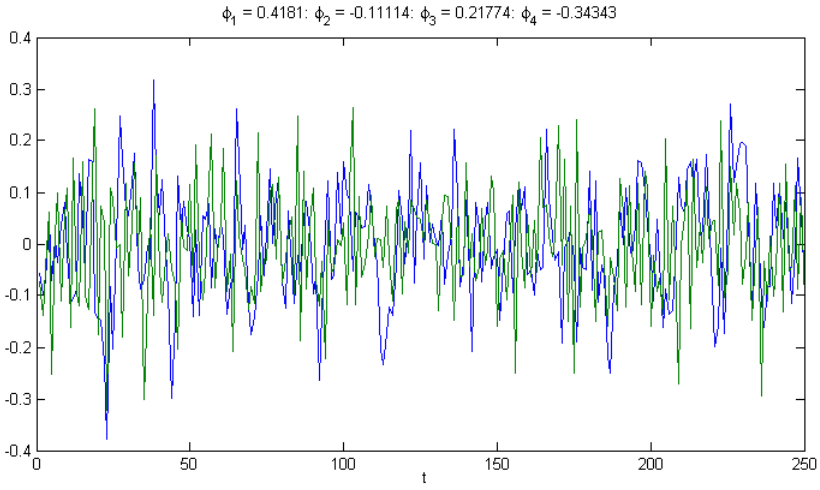

Figure 4.4: A time series of 250 timepoints for an AR(1) model with true parameters $\phi_1 = 0.5, \phi_2 = -0.1, \phi_3 = 0.2$ and $\phi_4 = -0.3$

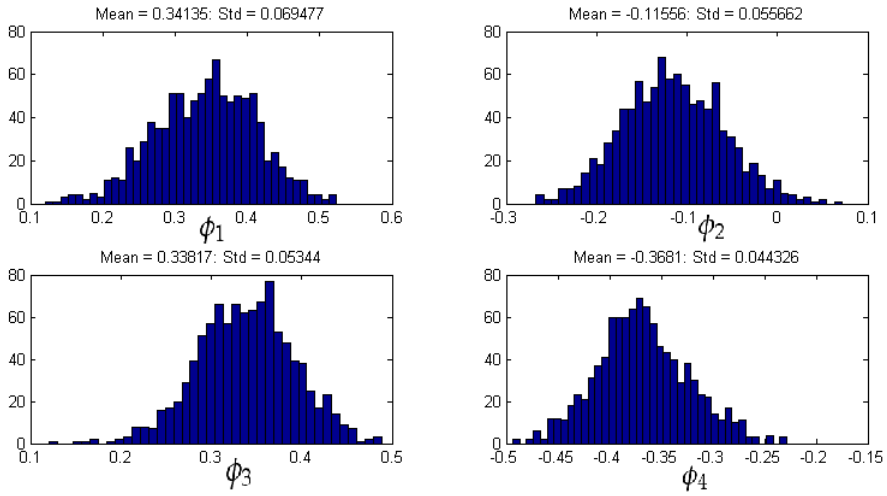

Figure 4.5: Non overlapping block bootstrap with 1000 bootstrap replicates showing bootstrap mean and standard deviation estimates.

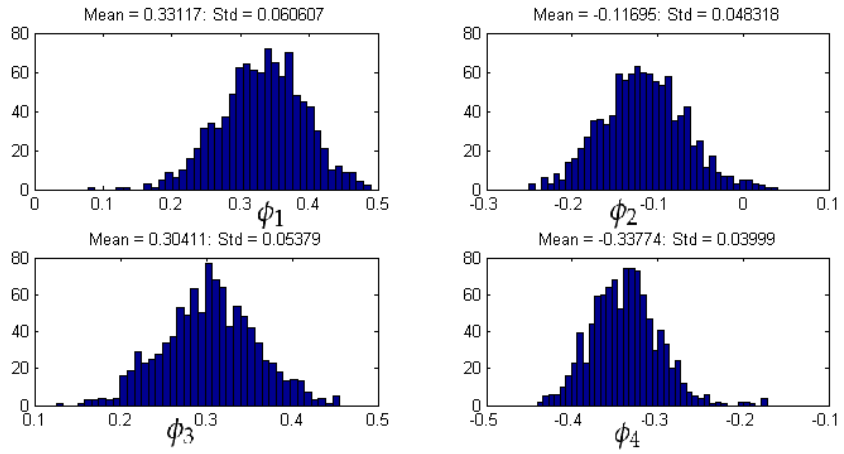

Figure 4.6: Overlapping block bootstrap with 1000 bootstrap replicates showing bootstrap mean and standard deviation estimates.

4.5 Granger causality in the frequency domain

With the multivariate bootstrap extension for Algorithm 4.1, the choice of block-size alters the success of estimation of the parameters of the model. Clearly, this extra parameter needs to be chosen well for the model. By using a residual based bootstrap, this parameter is removed. In a multivariate setting, this residual bootstrap is not so easily implemented. Hidalgo [228] proposes an algorithm to deal with this by using the frequency domain to implement the residual bootstrap. A statistic is then derived which allows us to accept or reject a hypothesis test of the form

$$H_0 : x \overset{G}{\nrightarrow} y \quad vs \quad H_1 : x \overset{G}{\rightarrow} y \tag{4.5.1}$$

Algorithm 4.2 - Hidalgo frequency domain Granger causality

Let $w_t = (y_t, x_t)$ be a vector of observed values for two time series x_t, y_t and for $t = 1, ..., T$. Given that we are using discrete analogues for estimates of continuous values, we also use the following. Let $m = m(T)$ be a number increasing slowly with T such that $m^{-1} + mT^{-1} \to 0$. We use that $m = \sqrt{T}/2$. Let $M = [T/4m]$ where $[z]$ denotes the integer part of z. We use the convention that $\lambda_j = 2\pi j/T$.

The test is based upon an (infinite) autoregressive process representation, as with algorithm 4.1 such that

$$\sum_{j=0}^{\infty} A_j w_{t-j} = \varepsilon_t \tag{4.5.2}$$

and an equivalent representation to perform the hypothesis test is to test for the model

$$y_t = \sum_{j=-\infty}^{\infty} c(j) x_{t-j} + u_t \tag{4.5.3}$$

and simultaneously test that the $c(j)$ are all zero for $j < 0$. This is equivalent to all values of y depending only on previous or current values of x and, as such, future values of x do not have a causal effect on values on y.

The main part of the paper by Hidalgo is the following statistic

$$S^*(\mu) = Re\left(\int_0^{\mu} vec\left(\sum_{j=-\infty}^{0} c(j-1)e^{-i\pi j\lambda}\right)d\lambda\right) \tag{4.5.4}$$

with the null hypothesis that

$$H_0 \ : \ S^*(\mu) = 0 \qquad \forall \mu \in [0,1] \tag{4.5.5}$$

A modified statistic is used, based on a Riemann sum approximation for the integral, given as

$$S_T(\mu) = Re\left(\frac{1}{M}\sum_{p=1}^{[M\mu]} vec\left(\sum_{j=2-M}^{0} \hat{c}(j-1)e^{-ij\lambda_{2mp}}\right)\right) \tag{4.5.6}$$

First, estimates of the $c(j)$ are obtained for the model. The hat notation $\hat{c}(j)$ is used to denote these estimates. The following estimation procedure is based taken from Hannan [202]

1. Calculate periodogram of $\{w_t\}$

$$I_{ww}(\lambda) = (2\pi T)^{-1}(\sum_{t=1}^{T} w_t e^{it\lambda})(\sum_{t=1}^{T} w_t e^{-it\lambda})' \tag{4.5.7}$$

2. Find spectral density matrix estimate

$$\hat{f}_{ww}(\lambda) = \frac{1}{2m+1}\sum_{j=-m}^{m} I_{ww}(\lambda_j + \lambda) \tag{4.5.8}$$

3. Calculate the frequency response function

$$\hat{C}_{2mp} = \hat{f}_{yx,2mp}\hat{f}^{-1}_{xx,2mp} \tag{4.5.9}$$

where f_{xx} and f_{yx} are components of the spectral density matrix f_{ww}

4. Estimate the parameters

$$\check{c}(j) = \frac{1}{2M} \sum_{p=0}^{2M-1} \hat{C}_{2mp} e^{ij\lambda_{2mp}} \tag{4.5.10}$$

However, Hidalgo proposes a slightly modified version of $\check{c}(j)$ to remove estimating $f_{xx}(0)$ which may potentially be infinite. He proposes the use of the estimator

$$\hat{c}(j) = \frac{1}{2M} \sum_{p=1}^{2M-1'} \hat{C}_{2mp} e^{ij\lambda_{2mp}} \tag{4.5.11}$$

where $\sum_{p=1}^{2M-1'} \phi_{2mp} e^{ij\lambda_{2mp}}$ denotes $\sum_{p=1}^{2M-1} \phi_{2mp} e^{ij\lambda_{2mp}+\phi_{2m}}$

Given the estimates for $\hat{c}(j)$, we can obtain estimates of the residuals for $t = 1, ..., T$

$$\hat{u}_t = y_t - \sum_{l=1-M}^{M} \hat{c}(l) x_{t-l} \tag{4.5.12}$$

Now there are two options of performing the bootstrap, either to bootstrap the Fourier transforms of the residuals, or alternately bootstrap the residuals and then take the Fourier transform of these.

Let $\tilde{v}_{\hat{u}}(\lambda_j) = \bar{\hat{f}}^{-1/2}_{\hat{u}\hat{u},j} w_{\hat{u}}(\lambda_j)$ for $j = 1, ..., [T/2]$,

then take the DFT of $\hat{u}_t$, denoted $w_{\hat{u}}(\lambda_j)$ and compute the standardised residuals $v_{\hat{u}}(\lambda_j)$, where

$$v_{\hat{u}}(\lambda_j) = \tilde{\Xi}^{-1/2} \left(\tilde{v}_{\hat{u}}(\lambda_j) - \frac{1}{[T/2]} \sum_{l=1}^{[T/2]} \tilde{v}_{\hat{u}}(\lambda_l) \right) \tag{4.5.13}$$

and

$$\tilde{\Xi} = \frac{1}{[T/2]} \sum_{k=1}^{[T/2]} \left(\tilde{v}_{\hat{u}}(\lambda_k) - \frac{1}{[T/2]} \sum_{l=1}^{[T/2]} \tilde{v}_{\hat{u}}(\lambda_l) \right) \left(\tilde{v}_{\hat{u}}(\lambda_k) - \frac{1}{[T/2]} \sum_{l=1}^{[T/2]} \tilde{v}_{\hat{u}}(\lambda_l) \right)' \tag{4.5.14}$$

Then the bootstrap is taken from these transformed residuals, and denote this boostrap sample as $\eta^*_{j,1}$ for $j = 1, ..., [T/2]$.

Conversely, the centred residuals $\tilde{u}_t$ are found where

$$\tilde{u}_t = \tilde{\Sigma}_{\hat{u}}^{-1/2}\left(\hat{u}_t - T^{-1}\sum_{t=1}^{T}\hat{u}_t\right) \ , \ \tilde{\Sigma}_{\hat{u}} = \frac{1}{T}\sum_{t'=1}^{T}\left(\hat{u}_{t'} - T^{-1}\sum_{t=1}^{T}\hat{u}_t\right)\left(\hat{u}_{t'} - T^{-1}\sum_{t=1}^{T}\hat{u}_t\right)' \tag{4.5.15}$$

From these centred residuals, a bootstrap sample $\tilde{u}^*$ for $t = 1, ..., T$ is formed.

Applying the discrete Fourier transform to these gives the second bootstrap sample

$$\eta_{j,2}^* = \frac{1}{T^{1/2}}\sum_{t=1}^{T}\tilde{u}_t^* e^{-it\lambda_j}$$

for $j = 1, ..., [T/2]$.

Using these bootstrap samples we obtain the bootstrap distributed lag regression model for $k = 1, 2$ and $j = 1, ..., [T/2]$

$$w_{y^*,k}(\lambda_j) = \sum_{l=0}^{M}\hat{c}(l)w_{x,l}(\lambda_j) + \bar{f}_{\hat{u}\hat{u}}^{1/2}(\lambda_j)\eta_{j,k}^*$$

Using these bootstrap samples, analogue bootstrap parameters $\hat{c}_k^*(l)$ can be calculated which are required for the statistic. Then the statistic is given as

$$S_{T,k}^*(\mu) = Re\left(\frac{1}{M}\sum_{p=1}^{[M\mu]} vec\left(\sum_{l=2-M}^{0}\hat{c}_k^*(l-1)e^{-il\lambda_{2mp}}\right)\right)$$

for $\mu \in [0, 1]$ and $k = 1, 2$.

Maximising over μ then gives the maximum value for the statistic and it is this that is the significance level of the interaction required.

In practise, this technique works well for long range time series, as explained in the paper by Hidalgo [228]. One significant problem is the use of optimisation required in the last step for every bootstrap.If the optimisation is taken over a fine grid then the algorithm can be slow to perform; over a coarse grid it may not optimise well. Couple with the number of bootstraps required to obtain meaningful results makes this procedure computationally very expensive.

4.6 Alternative method for Granger causality

Whilst Algorithm 4.2 allows considers long range time series in the frequency domain. However, this is penalised by a heavy cost in runtime by needing to maximise over a grid for each bootstrap and is therefore less practical where there may be many interactions to analyse. The next algorithm is a modified form of the original Granger causality Algorithm 4.1 where bootstrapping can be easily applied in both the block and residual forms. The details of the algorithm are given in the paper by Hatemi and Shukur [229].

Algorithm 4.3 - Hatemi-Shukur Algorithm

Let **x** and **y** be time series of length T and a VAR(p) model fitted in the form

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} \phi_{11}^1 & \phi_{12}^1 \\ \phi_{21}^1 & \phi_{22}^1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \end{bmatrix} + \cdots + \begin{bmatrix} \phi_{11}^p & \phi_{12}^p \\ \phi_{21}^p & \phi_{22}^p \end{bmatrix} \begin{bmatrix} x_{t-p} \\ y_{t-p} \end{bmatrix} + \begin{bmatrix} \varepsilon_t^x \\ \varepsilon_t^y \end{bmatrix}$$

or, more concisely,

$$\mathbf{z}_t = \mathbf{\Phi}^1 \mathbf{z}_{t-1} + \ldots + \mathbf{\Phi}^p \mathbf{z}_{t-p} + \varepsilon_t$$

By setting

$$Y = (\mathbf{z}_1, \ldots, \mathbf{z}_T) \quad ; \quad B = (\mathbf{\Phi}^1, \ldots, \mathbf{\Phi}^p)$$

$$Z_t = (1 \quad \mathbf{z}_t, \ldots, \mathbf{z}_{t-p+1})' \quad ; \quad Z = (Z_0, \ldots, Z_{T-1})$$

and

$$\delta = (\varepsilon_1, \ldots, \varepsilon_T)$$

then the bivariate VAR(p) model can be written in the form

$$Y = BZ + \delta \tag{4.6.1}$$

as described in [229], with B a matrix of parameters to be estimated.

Let $\hat{\delta}_U$ be the residual matrix for the unrestricted regression from the full dataset, with $\hat{\delta}_R$ the corresponding matrix under the null hypothesis that there is no cross interaction (and hence no causality present) and the cross parameters are set to zero (so that $\phi_{12}^j = \phi_{21}^j = 0 \ \forall j$. Further define the cross product of residuals as $S_U = \hat{\delta}_U' \hat{\delta}_U$ and $S_R = \hat{\delta}_R' \hat{\delta}_R$.

The statistic of interest is then

$$Rao = \frac{\phi}{q}(U^{1/s} - 1) \tag{4.6.2}$$

where $s = \sqrt{\frac{q^2-4}{k^2(G^2+1)-5}}$, $r = q/2 - 1$, $\phi = \Delta s - r$, $\Delta = T - (k(kp+1) - Gm) + 0.5[k(G-1) - 1]$ and $U = detS_R / detS_U$. $q = Gm^2$ is the number of restrictions placed under the null hypothesis.

The statistic Rao is approximately distributed as the F-statistic $F(q, \phi)$ under H_0.

In order to overcome the problem where two individual univariate time series are compared against each other, a variation in the statistic is required, as $G = 1$ in this case. This would leave q as the square of the number of parameters fitting the model, which will typically be small i.e. 1 or 2. Clearly s then will be either imaginary or zero. Instead, let $s = \sqrt{\frac{q^2}{k^2(G^2+1)-5}}$. As bootstrapping will be considered, the variation in Rao will still be shown.

4.6.1 Bootstrapping of the Hatemi-Shukur Algorithm

The block bootstrap can be obtained from the original dataset Y to produce a bootstrap equivalent Y^b. From this , the parameters in the model $Y^b = BZ^b + \delta^b$ can be estimated and the corresponding statistic Rao^b found for $b = 1, ..., B$.

The residual bootstrap is applied by replacing the estimated residuals for the model with a resample of centred residuals. Let δ^* be resamples with replacement from the centred residuals $\hat{\delta} - \bar{\delta}$. The estimate of B is then $\hat{B} = YZ'(ZZ')^{-1}$ and the residual bootstrap is based on this adjusted dataset

$$Y^* = \hat{B}Z^* + \delta^* \tag{4.6.3}$$

Correspondingly, the residual bootstrap Rao^{*b} is obtained from this dataset for $b = 1, ..., B$.

4.7 Example of the Granger Causality algorithms

In order to show the performance of each of the algorithms described, simulated datasets are obtained from an underlying model with the parameters known. In order to understand how the variation in each of these algorithms alters the performance, the values of two of the parameters are changed. This should show where Granger causality should indeed occur and also, where the parameters are zero, it should not occur.

The true values are taken from the following models:

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.2 & \alpha \\ -0.1 & 0.1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \end{bmatrix} + \begin{bmatrix} -0.1 & \beta \\ 0.2 & -0.1 \end{bmatrix} \begin{bmatrix} x_{t-2} \\ y_{t-2} \end{bmatrix} + \begin{bmatrix} \varepsilon_t^x \\ \varepsilon_t^y \end{bmatrix}$$

where

$$\begin{bmatrix} \varepsilon_t^x \\ \varepsilon_t^y \end{bmatrix} \sim \mathcal{N}(0, \mathbf{\Sigma})$$

where $\mathbf{\Sigma}$ is the covariance matrix of x and y and is set to exhibit small amounts of noise in the system

$$\mathbf{\Sigma} = \begin{bmatrix} 0.001 & 0.002^2 \\ 0.001^2 & 0.0009 \end{bmatrix}$$

The parameters α and β represent the effect of y_t on x_t; if they are zero then y should not Granger cause x. By varying the magnitude of the values of α and β, this will show how the presence of Granger causality is detected at different levels. Table 4.4 shows the values of α and β used and the label of the causality type, along with the column number shown in the graphs.

Table 4.4: Increasing amounts of Granger causality applied to example

Causality Type	Column Number	α	β
None	1	0	0
Small	2	0.01	0.01
Regular	3	0.1	0.2
Large	4	0.4	0.6

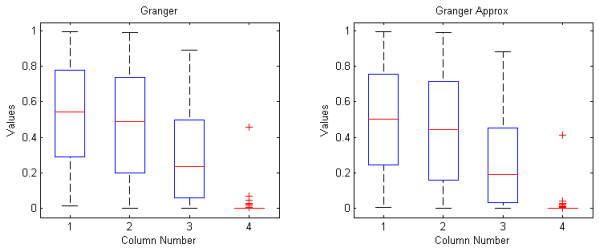

Figure 4.7: Boxplots of Granger causality test p-values for increasing amounts of causality under Algorithm 4.1 and approximate χ^2 model.

The number of timepoints is fixed at $T = 100$ and 100 monte carlo runs are applied with the distribution of the p-values of these monte carlo runs shown. For the bootstrapped algorithms, 100 bootstraps for each of these runs. Further, the block size is set to 10 for the block based bootstrap algorithm. As the data is generated from a VAR(2) model, $p = 2$ parameters are backfitted into the model. The model is to test whether $y \xrightarrow{G} x$.

The results for each of the algorithms 4.1 - 4.3 are shown in Figures 4.7 - 4.9 respectively. As the magnitude of the parameters increases, the p-value should tend towards zero.

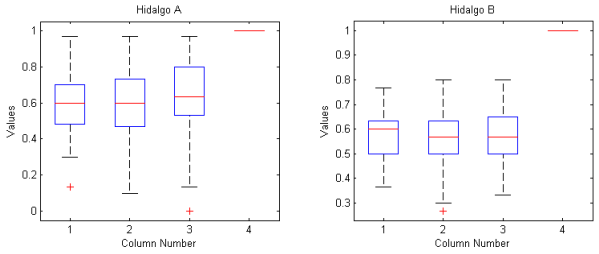

Figure 4.8: Boxplots of Granger causality test p-values for increasing amounts of causality under Hidalgo Algorithm 4.2 model.

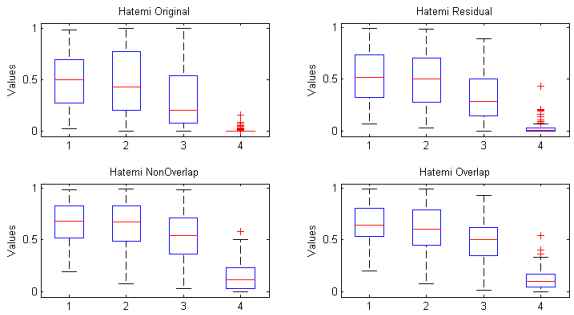

Figure 4.9: Boxplots of Granger causality test p-values for increasing amounts of causality under Hatemi-Shukur Algorithm 4.3 with no bootstrapping

Figure 4.7 shows the application of Algorithm 4.1 under both the F-test p-value found and also the χ^2 approximation. The results are as expected, with zero and low values of the parameters showing wide variation in the p-values with increased magnitude of parameters showing very little distribution around a very small p-value.

Figure 4.8 shows the application of Algorithm 4.2 under the two variations of the Hidalgo algorithm; Hidalgo A refers to the bootstrapping of residuals applied to the Fourier transforms, with Hidalgo B taking the bootstrapping of Fourier transforms of the bootstraps. Indeed, the results at low parameters values are expected but quite unexpected for where the p-values should be very low. This would warrant further investigation as to the cause of such unexpected results.

Figure 4.9 applies the Hatemi Shukur Algorithm 4.3 under all variations of the bootstrap; the original has no bootstrapping applied, and then residual, block overlap and block non-overlap bootstraps are applied. The original and residual bootstrap variations perform well and as expected. The residual bootstrap also shows a narrower distribution of p-values for the 'None' and 'Small' categories of causality. The block bootstrap performs less well, with wide distribution of p-values shown at the 'Large' category.

4.8 Summary of Granger Causality Algorithms

The algorithms 4.1 - 4.3 perform variably for the example shown, where causality should indeed be detected. The Hidalgo Algorithm (Algorithm 4.2) performs unexpectedly and this would warrant further investigation. As such, this algorithm is not used in subsequent analysis.

The original Granger causality Algorithm 4.1 performs well and as anticipated; similarly the Hatemi-Shukur algorithm (Algorithm 4.3) with some improvements shown by application of the residual bootstrap. Whilst the bootstrap algorithm performs well, the extra compuational cost of generating bootstraps does not necessarily justify the minimal improvement. As such, the original Granger causality Algorithm 4.1 performs adequately as well does the Hatemi-Shukur algorithm. Where bootstrapping is to be applied, then the Hatemi-Shukur algorithm is best implemented for comparison; where bootstrapping is not performed, there is equal benefit to apply Algorithm 4.1.

4.9 Application of Granger Causality

The use of Granger causality applied to reconstruction of genetic networks was considered by Mukhopadhyay and Chatterjee [2]. Here, time series for genes in a given network were used to calculate the significance of all possible cross interactions. Autoregulating genes cannot be considered due to the structure of the Granger causality algorithm.

Further to this, for each directional pair of interactions between two genes, the interaction with the lower significance is discarded. The reasoning behind this is unclear and therefore does not take into account where there may be bidirectional interaction as can be biologically seen. In order that these potential interactions are considered, all possible interactions shall be taken into account with the following algorithms.

4.9.1 Recovery of Gene Networks

The first algorithm is a simple refinement of the algorithm developed by Mukhopadhyay and Chatterjee [2], where every interaction is calculated and the Granger causality significance used to decide whether the interaction is statistically significant or not. As pointed out in Section 4.2.1, the information criterion is limited in this and subsequent cases to order 2 or 3.

Algorithm 4.4 - Gene Network Recovery with Granger causality

Let $X^1, ..., X^n$ be n genetic time series

1. Select two gene time series X^i and X^j

2. Combine two time series into a single bivariate time series $Y = [X^i \;\; X^j]'$

3. Estimate order of Y using Bayesian Information Criterion restricted to order 2 or 3

4. Calculate significance of X^i Granger causing X^j and vice versa, S_{ij}, S_{ji}

5. If $S_{ij} > \alpha$ for some α, then interaction $I_{ij} = 1$; else $I_{ij} = 0$

6. Repeat for all $i, j = 1, ..., n; i \neq j$

The choice of α has to be considered, whether it be a fixed value or by the use of a correction if the algorithm is to be considered as a multiple hypothesis test. This is not further explored here and the α level is set in advance as fixed.

4.9.2 Recovery of Gene Networks with Bootstrapping

By extending Algorithm 4.1 to include the use of bootstrapping the bivariate time series used for obtaining Granger causality, the Hatemi-Shukur algorithm (Algorithm 4.3) is used. As shown previously, the residual bootstrap form of the algorithm performs best although the Algorithm described can be more generally extended to any bootstrapping form of any of the Granger causality algorithms.

Algorithm 4.2 - Gene Network Recovery with bootstrapped Granger causality

Let $X^1, ..., X^n$ be n genetic time series and B the number of bootstraps.

1. Select two gene time series X^i and X^j

2. Combine two time series into a single bivariate time series $Y = [X^i \;\; X^j]'$

3. Estimate order of each Y using Bayesian Information Criterion restricted to order 2 or 3

4. Create bootstraps of these bivariate series $Y^1, ..., Y^b$ for $b = 1, ..., B$ for chosen order

5. Calculate bootstrap significance of X^i Granger causing X^j and vice versa, S^b_{ij},S^b_{ji}

6. Use bootstrap mean significance $S_{ij} = \frac{1}{B}\sum_{b=1}^{B} S^b_{ij}$ and $S_{ji} = \frac{1}{B}\sum_{b=1}^{B} S^b_{ji}$

7. If $S_{ij} > \alpha$ for some α, then interaction $I_{ij} = 1$; else $I_{ij} = 0$

8. Repeat for all $i, j = 1, ..., n; i \neq j$

4.10 Measuring Similarity

In order to measure how well a known network and recovered network, similarity measures can be used that look at whether an interaction is correctly predicted or not. Define an indicator variable d_{ij} depending on whether the interaction from i to j is found in both the true and the estimated networks. We state explicitly that we mean not just where an interaction exists but also where an interaction does not exist.

$$d_{ij} = \begin{cases} 0, & \text{interaction in both} \\ 1, & \text{otherwise} \end{cases}$$

Then a similarity measure, S, may be defined as the normalised sum of these indicators. For n nodes, where self regulation is omitted,

$$S = \frac{1}{n(n-1)} \sum_{i \neq j} d_{ij}$$

This may be interpreted that S is a measure of how far from the true network an estimated network is. If $S = 0$ then the networks are in perfect agreement and increasing values of S show a departure from the true network.

With networks that have been obtained from biological observation, these are known to not necessarily be 'true' in the sense that all interactions may not have been studied. However, one way to take into account any biological knowledge is to somehow incorporate this into weight parameters for the network.

$$S = \frac{\sum_{i \neq j} w_{ij}\, d_{ij}}{\sum_{i \neq j} w_{ij}}$$

4.11 Test Network

A test network is considered in order to assess how well the algorithm performs. The following network is taken from the paper by Mukhopadhyay and Chatterjee [2] and consists of 14 nodes with 12 interactions. These are broken down into components where Granger causality should be observed and none should be observed. Further to this, stationary and non-stationary variations are used to illustrate the effect this has on network recovery.

The components where Granger causality is observed are

$$\begin{aligned}
x_{2t} &= 0.29x_{2(t-1)} + 0.65x_{1(t-1)} + \epsilon_{2t} \\
x_{3t} &= 0.15x_{3(t-1)} + 0.29x_{2(t-1)} + 0.65x_{14(t-1)} + \epsilon_{3t} \\
x_{6t} &= 0.12x_{6(t-1)} + 0.3x_{7(t-1)} + 0.3x_{8(t-1)} + 0.3x_{9(t-1)} + \epsilon_{6t} \\
x_{4t} &= 0.17x_{4(t-1)} + 0.4x_{3(t-1)} + 0.7x_{6(t-1)} + \epsilon_{4t} \\
x_{5t} &= 0.6x_{5(t-1)} + 0.8x_{4(t-1)} + \epsilon_{5t} \\
x_{10t} &= 0.4x_{10(t-1)} + 0.3x_{11(t-1)} + \epsilon_{10t} \\
x_{12t} &= 0.4x_{12(t-1)} + 0.4x_{11(t-1)} + \epsilon_{12t} \\
x_{13t} &= 0.4x_{13(t-1)} + 0.4x_{11(t-1)} + \epsilon_{13t}
\end{aligned} \tag{4.11.1}$$

The stationary series are

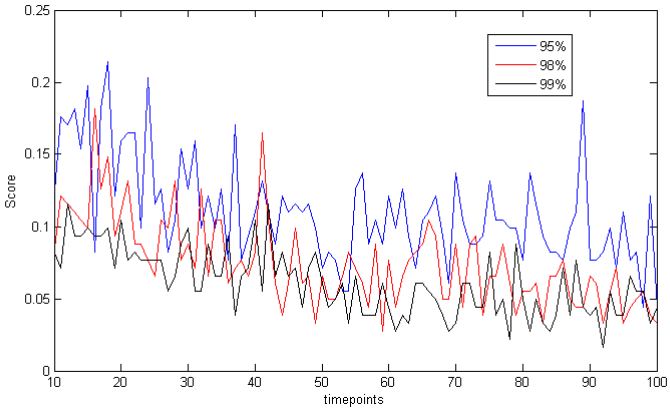

Figure 4.10: Hatemi-Shukur Algorithm 4.3 applied to stationary dataset.

$$
\begin{aligned}
x_{1t} &= 0.7x_{1(t-1)} + \epsilon_{1t} \\
x_{7t} &= 0.8x_{7(t-1)} + \epsilon_{7t} \\
x_{8t} &= 0.7x_{8(t-1)} + \epsilon_{8t} \\
x_{9t} &= 0.77x_{9(t-1)} + \epsilon_{9t} \\
x_{11t} &= 0.7x_{11(t-1)} + \epsilon_{11t} \\
x_{14t} &= 0.65x_{14(t-1)} + \epsilon_{14t}
\end{aligned}
\tag{4.11.2}
$$

with their corresponding non-stationary series

$$
\begin{aligned}
x_{1t} &= \sin\tfrac{\pi t}{40} + 0.7x_{1(t-1)} + \epsilon_{1t} \\
x_{7t} &= 0.8x_{7(t-1)} + \epsilon_{7t} \\
x_{8t} &= \cos\tfrac{\pi t}{40} + 0.7x_{8(t-1)} + \epsilon_{8t} \\
x_{9t} &= 0.77x_{9(t-1)} + \epsilon_{9t} \\
x_{11t} &= \cos\tfrac{\pi t}{40} + 0.7x_{11(t-1)} + \epsilon_{11t} \\
x_{14t} &= 0.65x_{14(t-1)} + \epsilon_{14t}
\end{aligned}
\tag{4.11.3}
$$

Figures 4.10 and 4.11 show the application of the Granger causality algorithm 4.1 with stationarity and non-stationarity present respectively for the α significance levels of 95%, 98% and 99% and the similarity score in section 4.8 achieved as the number of timepoints increases from 10 to 100. The results show that the same general trend is followed, although a lower score (and hence closer to the true network) is achieved at the 99% level. This is due to not only the number of true interactions being found, but also the number of true non-interactions found.

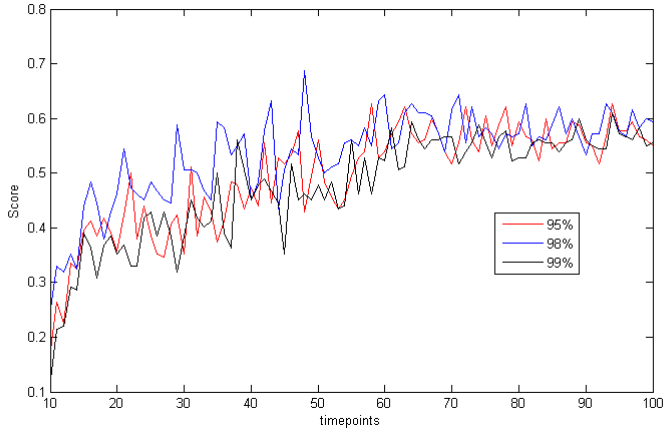

Figure 4.11: Hatemi-Shukur Algorithm 4.3 applied to non-stationary dataset.

Indeed, where stationarity is present, the results improve with the increased number of timepoints. This shows that the increased number of timepoints indeed helps to improve the modelling where the stationarity requirement is upheld. Where stationarity does not exist, the results show very poor performance. This issue of stationarity is therefore key to performing modelling in the Granger causality framework.

4.12 Multivariate Granger Causality

Previously the consideration has been on considering pairwise interaction between two time series to detect a causal link in the Granger sense. This can naturally be extended to considering the impact of many genes on a single target, or indeed vice versa with a single gene targeting many other genes, and more generally for many genes targeting many other genes. The Granger causality model can therefore easily be extended a vector of observations at timepoint t, the multivariate Granger model is

$$\begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{y}_t \end{bmatrix} = \begin{bmatrix} \Phi_{11}^1 & \Phi_{12}^1 \\ \Phi_{21}^1 & \Phi_{22}^1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{y}_{t-1} \end{bmatrix} + \cdots + \begin{bmatrix} \Phi_{11}^p & \Phi_{12}^p \\ \Phi_{21}^p & \Phi_{22}^p \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{t-p} \\ \boldsymbol{y}_{t-p} \end{bmatrix} + \begin{bmatrix} \varepsilon_t^{\boldsymbol{x}} \\ \varepsilon_t^{\boldsymbol{y}} \end{bmatrix} \tag{4.12.1}$$

where now, instead of univariate time series, $\boldsymbol{x}_t = (x_t^1, \ldots, x_t^l)$ and $\boldsymbol{y}_t = (y_t^1, \ldots, y_t^m)$ are multivariate vectors. Similarly Φ^j are block matrices.

Similarly to the univariate case, we can say that the m time series $\boldsymbol{y}$ fail to Granger cause the l time series $\boldsymbol{x}$ iff $\Phi_{12}^{k} = 0$ for $k = 1, ..., p$.

One problem with the use of multivariate Granger causality is the interpretation of what it means for a set of genes to be targeting another full set of genes. The next chapter considers the issue of a more multivariate setting in relation to the interaction of many genes upon each other in a less computationally laborious way than multivariate Granger causality and less parameters are required to be estimated.

CHAPTER 5

Data Reduction

In the previous chapter, algorithms were presented and developed to implement Granger Causality as a means of assessing significant interactions between genes by using time series data for each gene. Where the number of interactions for n genes is $O(n^2)$, the computational expensiveness is exponentially increased as the number of genes increases. Typically a genetic dataset may be of the order of thousands of genes, with even a useful subset of the order of hundreds. Depending on the choice of algorithm used to assess the causality between two genes, this may increase the expensiveness of the overall computation greatly.

This chapter looks at methods for data reduction in order to improve the time taken to compute and assess significant interactions. Clustering can help to reduce the size of a dataset by grouping together genes that have similar expression profiles, such as those which occur in similar families of genes. This is then extended to consider how much variation occurs within these clusters and how this impacts in finding significant interactions.

5.1 Clustering

Clustering algorithms takes multivariate observations of data and groups these observations into a fixed number of subsets so that elements within a cluster are similar to each other, in some specified way. Clustering techniques have progressed a long way and are widely used with large scale genomic data, each with their own features and drawbacks, as detailed in Kerr et al. [230]. Selec-

tion of an appropriate clustering technique is a challenge in itself. Here, two particular clustering algorithms are used.

The *k*-means algorithm [231] is one of the most widely used algorithms due the simplicity of application. This algorithm assigns all the data objects, such as multi-dimensional observations or time series, into a specified number of clusters dependent on the initial configuration of the allocation to cluster and some specified stopping criterion.

The Quality Threshold (QT) algorithm developed by Heyer et al. [232] was developed originally for use with genetic time series. This algorithm removes this initial configuration and pre-specification of the number of clusters required. As will be shown, it does come with an increased computational cost.

5.1.1 Distance between two points

Clustering algorithms rely on defining distance between two points. There are many different distance measures that can be applied. In particular, when dealing with time series, the distance is defined across all timepoints. Let x_i and x_j be two time series of length T, such that $x_i = (x_{i1}, ..., x_{iT})$. Then the distance between these two time series is the sum of the distance between individual points $d(x_i, x_j) = \sum_{t=1}^{T} d(x_{it}, x_{jt})$.

5.1.2 *k*-means Algorithm

Given n multivariate objects (such as time series observations), the *k*-means clustering algorithm [231] fixes the number of clusters, k, a priori and initially assigns each object to one of these clusters at random. From this initial configuration, the centroids (mean of objects within each cluster) are calculated for each cluster and readjusted until convergence within some specified tolerance occurs.

Algorithm 5.1 - k-means Clustering Algorithm

1. Choose k and convergence criterion

2. Assign all objects $X_1, ..., X_n$ uniformly at random to clusters $C_1, ..., C_k$

3. Calculate the centroid (mean point) of each cluster $m_j = \frac{1}{|C_j|} \sum_{X_i \in C_j} X_i$, $j = 1, ..., k$

4. Reassign variables $X_1, ..., X_n$ to nearest centroid

5. Recalculate centroids

6. Stop at convergence, when assignment to clusters is unchanged

For this algorithm, the least squares distance metric is used such that the algorithm seeks to minimise the function

$$V = \sum_{i=1}^{K} \sum_{X_i \in C_j} \left(X_i - m_j\right)^2 \tag{5.1.1}$$

5.1.3 $kmeans++$-algorithm

One of the problems with the k-means algorithm is that it initially assigns objects to each cluster at random, leading to different assignments of objects at convergence for the same dataset. One extension is the k means++ algorithm [233] which chooses an initial configuration that produces a more stable converging configuration and less variability. Here, the centroids are chosen initially as follows.

Algorithm 5.2 - $k++$-means Clustering Algorithm

1. Choose an initial centroid m_1 uniformly at random from objects $X_1, ..., X_n$

2. Choose the next centroid $m_i = x'$ for some object x with probability $\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$ where $D(x)$ is the shortest distance from the variable x to the closest centroid previously chosen.

3. Repeat until all k centroids are chosen.

5.1.4 QT-clustering

The k-means clustering algorithm is widely used due to the easy implementation and widely understood properties. One drawback is that the number of clusters, k ,must be defined in order to implement the algorithm. Further to this, the solution may not converge uniquely and is highly dependent on the initial configuration.

The Quality Threshold (QT) clustering algorithm by Kerr et al. [230] does not specify an initial configuration nor the number of clusters. Instead, by using a threshold value of how close members within a cluster must be, the resulting clustered configuration will be unique for the same dataset. This algorithm was designed for use with genetic time series so is presented here.

Algorithm 5.3 - Quality Threshold Clustering Algorithm

Let ρ_{ij} denote the correlation between the time series X_i and X_j. Now let $\rho_{ij}^{(l)}$ be the correlation between these time series but with the lth timepoint removed. The jackknife correlation between X_i and X_j is defined as $J_{ij} = \min\{\rho_{ij}^{(1)}, \ldots, \rho_{ij}^{(t)}, \ldots, \rho_{ij}^{(t)}\}$.

1. Select time series uniformly at random X_i as a candidate cluster C_j
2. To cluster C_j, add time series with greatest jackknife correlation to cluster
3. Iteratively add remaining time series until each gene is within threshold d
4. Repeat 2 and 3 for every gene, including overlaps
5. Retain largest cluster, remove these genes from dataset
6. Repeat 2-5 on reduced dataset until all time series are assigned to clusters

5.1.5 Choice of Clustering Algorithm

The two clustering algorithms presented (with a modification to one also presented) vary in their approach to implementation and computational cost. The k-means algorithm is quick and easy to implement, but the final configuration depends heavily on the initial configuration, number of clusters chosen and the

convergence criterion. The QT clustering algorithm is computationally more expensive but uniquely defines a final configuration, subject to the choice of d.

Considered first is the choice of threshold parameter d and the effect on the ability of the QT clustering algorithm to recover a set of clusters. Clusters are generated by taking 20 AR(2) models and taking multiple runs in order to give 100 time series overall; the clusters are not all of fixed size with the largest containing 8 time series and the smallest containing a single time series. In order that the time series are similar within each cluster, the noise component of the AR(2) model is made small.

These 100 time series are then clustered with the the QT clustering algorithm. In order to assess the ability of the algorithm to reconstruct the clusters, a score is used where 1 is assigned if the time series is assigned to the correct cluster and 0 if it is wrongly assigned. These scores are summed and normalised in order to give the overall score. Table 5.1 shows how the value of d changes the score, with a lowest threshold of 0.1 and the highest at 1.0.

Table 5.1: The effect of the threshold parameter d in the QT-clustering algorithm on ability to reconstruct a set of clusters.

d	Score
0.1	0.9876
0.2	0.9042
0.3	0.7344
0.4	0.6838
0.5	0.6584
0.6	0.5934
0.7	0.5497
0.8	0.4283
0.9	0.5081
1.0	0.4875

It is noticeable that a lower threshold value results in a better recovery of the network due to members of each cluster needing to be more similar to each

other than those with a higher threshold. Also of note is the fact that it doesn't take a big jump in the threshold parameter before network reconstruction is poor, as shown by the large drop in score when $d = 0.3$.

In order to compare the speed of the QT clustering algorithm against the *k*-means algorithm, clustering on time series is performed for speed under each of these algorithms. Time series of fixed length 25 were generated with 20, 50, 100 and 500 time series objects clustered into 4, 10, 20 and 100 clusters respectively. As seen in Table 5.2, there is little difference in computational time taken to cluster the lower number of time series objects. As the number of objects to be clustered increases, the *k*-means algorithm performs significantly quicker than QT clustering.

Table 5.2: Comparison of *k*-means and QT clustering algorithms

Objects	Clusters	*k*-means	QT clustering
20	4	1.2 secs	3.5 secs
50	10	10.4 secs	40.5 secs
100	20	2.3 mins	12.6 mins
500	100	32.4 mins	4.2 hours

For the subsequent use of clustering algorithms, *k*-means clustering is chosen. Despite the sensitivity of the algorithm, large numbers of time series objects are to be clustered which would be unfeasible with QT clustering due to the length of time taken to perform the clustering.

5.1.6 Number of Clusters

As the *k*-means algorithm requires the number of clusters to be specified in advance, care needs to be taken as to the number of clusters to choose. Too many and the purpose of clustering is diminshed with the increased likelihood of similar clusters; too few and there may be lots of variability within the cluster which could lead to spurious results with any application of the clusters.

Whilst there has been discussion on the optimal number of clusters to use, such as in Ray and Turi [234], one rule of thumb often used is $k \approx (n/2)^{1/2}$ [235].

This approximation is used subsequently when considering large numbers of objects to be clustered.

5.2 Clustering applied to Granger Causality

In the previous chapter, Granger Causality was introduced as a measure of the significance of interaction between two time series and whether there is statistical evidence that change in one time series may cause a change in another time series. For genetic data, where there may be hundreds or even thousands of time series to consider, the number of pairwise interactions would be too cumbersome to easily analyse. By using a clustering algorithm on the data, the number of pairwise interactions can be reduced greatly. Here, the k-means clustering algorithm (Algorithm 5.1) is used and the significance calculated between centroids of the clusters.

5.2.1 Granger Causality with Clustered Time Series

By calculating the significance of interactions between centroids of clusters, this provides a statistical measure between the clusters and not the original genetic time series which are the interactions of interest. In order to use the clusters meaningfully, each possible interaction between members of the clusters being used is assigned the overall significance between the centroids of the clusters, as described in the following algorithm.

Algorithm 5.4 - Clustered Granger Causality Algorithm

Let $X_1, ..., X_n$ be time series of length T.

1. Assign $X_1, ..., X_n$ to clusters $C_1, ..., C_k$ by k-means clustering (Algorithm 5.1), let $N(C_i)$ be the number of time series objects in cluster C_i

2. Calculate centroid of each cluster $M_1, ..., M_k$ where $M_i = \frac{1}{N(C_i)} \sum_j X_{C_i,j}$

3. Calculate significance between all centroids $\hat{S}_{a,b} = S_{M_a,M_b}$ for $a, b = 1, ..., k; a \neq b$ by Granger Causality (Algorithm 4.4)

4. Assign this value to all possible pairings: if $X_\alpha \in C_a$ and $X_\beta \in C_b$ then $S_{\alpha,\beta} = \hat{S}_{a,b}$ for $\alpha, \beta = 1, ..., n$

This then assigns the original $n(n-1)$ pairwise interactions to a reduced number of interactions between clusters $k(k-1)$.

5.2.2 Assessing clustered interactions

Due to the use of the *k*-means clustering algorithm, Algorithm 5.4 will produce different results each time it is run for the same dataset. In turn this will assign different significance values to each of the possible interactions. In order to overcome spurious results from a single run, the solution proposed is to run the algorithm many times and combine the resulting significances so as to find interactions that are repeatedly significant.

By summing the significances obtained over all runs, those interactions that are consistently highly significant will be ranked highly when the resulting sum or product is considered. This can be described as follows.

Algorithm 5.5 - Overall sum significance level for clustered Granger Causality

1. For run m, calculate significance $S_{i,j}^m$ between X_i and X_j using Algorithm 5.4
2. Over r runs, $S_{i,j}^{\text{Sum}} = \sum_{m=1}^{r} S_{i,j}^m$
3. Rank interactions based on S^{Sum}

Figure 5.1 shows the application of Algorithm 5.5 for 500 monte carlo runs for 500 time series of length 25 generated. Here the number of clusters is increased from 25 to 75 in increments of 5. All the curves show similar results with the significance values ranked in order. There is a very slow decline before a steep drop in the overall signifiance value at around the 600th ordered overall significance value ranking.

Whilst these results may provide useful information for the most significant interactions, within the midrange of rankings it may provide uncertain results. One way to deal with this is to consider whether the interaction is significant at some α level.

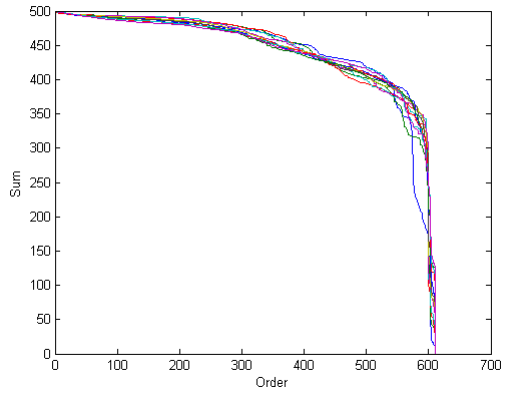

Figure 5.1: Algorithm 5.5 showing the sum of significances for 500 monte carlo runs of 500 time series of length 25. The number of clusters increase from 25 to 75 in increments of 5.

Here, if the interaction is significant at or above the α level then it is kept, otherwise it is set to 0. More formally, if $S^m_{i,j} < \alpha$ then $\tilde{S}^m_{i,j} = 0$; else $\tilde{S}^m_{i,j} = S^m_{i,j}$. Algorithm 5.5 is then altered as follows.

Algorithm 5.5a - Overall count significance level for clustered Granger Causality

1. For run m, calculate significance $\tilde{S}^m_{i,j}$ between X_i and X_j using Algorithm 5.4
2. Over r runs, $\tilde{S}^{\text{Sum}}_{i,j} = \sum_{m=1}^{r} \tilde{S}^m_{i,j}$
3. Rank interactions based on $\tilde{S}^{\text{Sum}}$

Figure 5.2 shows that a less steep decline in the ranking of significances with the α level cutoff is present, and a visible effect from the number of clusters used, with the top curve showing the least number of clusters. This means that the use of a counting based mechanism of significance is useful as a means of assessing overall significance of individual interactions.

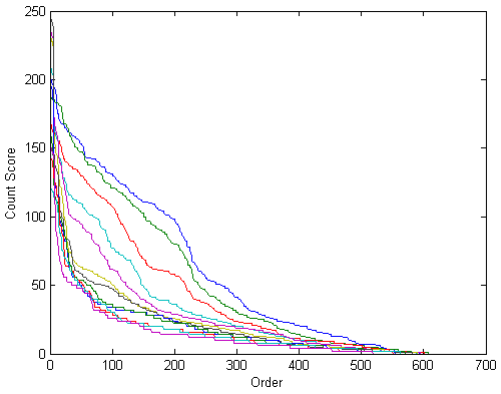

Figure 5.2: Algorithm 5.5 showing the count of significances at 95% significance level for the same 500 monte carlo runs of 500 time series of length 25 as shown in Figure 5.1.

5.3 Principal Components Analysis

When the elements of the clusters are very similar, assigning each possible pairing the significance of the causality based on cluster centroids provides a reasonable and useful estimate of all possible cluster interactions. However, the number of clusters can greatly vary how near each object is to each other within the cluster. Clearly, fewer clusters can lead to greater within cluster variability. This variability is now considered in an extension of Algorithm 5.3 by the application of Principal Components.

5.3.1 Principal Components

Principal Components are used to find where the most amount of variability within a dataset occurs, which may account for large amounts of the overall variability within the data . By transforming the data, this means that the original data can be composed of the principal components and their directions in n-dimensional space, for vectors of length n. By taking the first few principal components, if they account for most of the overall variation, means that the data can be described at a lower dimension than the original data, without too much loss of information. Whilst this is useful in most situations, the principal components as measures of variability within data shall be the focus for the following analysis.

5.4 Granger Causality with PCA Algorithm

In Algorithm 5.4, the significance between interactions is based solely on the centroid or mean of the clusters. The variation within the cluster should be considered for the effect that is may have on the overall significance of the interactions. The derivation of the principal components and their eigenvalues is taken from the book by Jolliffe [236].

For cluster C_a with elements $x_1, ..., x_n$ and cluster C_b with elements $y_1, ..., y_m$, the principal components are $v_1, ..., v_r$ and $w_1, ..., w_s$ respectively, with respective eigenvalues $\lambda_1, ..., \lambda_r$ and $\xi_1, ..., \xi_s$. Then a representation of the clusters C_a, C_b is

$$\bar{x} \pm \lambda_1^{1/2} v_1 \pm \ldots \pm \lambda_r^{1/2} v_r$$

and

$$\bar{y} \pm \xi_1^{1/2} w_1 \pm \ldots \pm \xi_s^{1/2} w_s$$

respectively. By taking one or a few principal components, the most variation within the cluster can be explained.

With Granger causality applied on two single variables, as in Algorithm 4.1, the variables are singly regressed on each other with some chosen lag. Now, the regression is performed on the means, as in Algorithm 5.4, but with the principal components of each cluster added in.

Algorithm 5.6 - Clustered Granger Causality with PCA

For p lags, to determine whether C_a causes C_b, the regression

$$\begin{aligned} \bar{y}(t) \quad = \quad & \beta_1 \bar{y}(t-1) + \ldots + \beta_p \bar{y}(t-p) + \gamma_1 w_1(t-1) + \ldots + \alpha_1 \bar{x}(t-1) + \ldots \\ & \ldots + \alpha_p \bar{x}(t-p) + \delta_1 v_1(t-1) + \ldots \end{aligned}$$

is compared against the model where the elements of cluster C_b are regressed upon themselves

$$\bar{y}(t) = \omega_1 \bar{y}(t-1) + \ldots + \omega_p \bar{y}(t-p) + \pi_1 w_1(t-1) + \ldots \tag{5.4.1}$$

As in Algorithm 4.1, the residual sum of squares is compared to give the significance.

When the principal components are not added in, this reduces to the standard Granger causality algorithm (Algorithm 4.1). By adding in the components one at a time, this will take into account the variability within the data as explained by these principal components. The first few principal components will usually be sufficient.

5.5 Example

In order to show the application of this concept, two time series are generated with 20 timepoints, labeled x and y. These have been generated such that Granger causality is significant only in one direction under Algorithm 4.1. Different amounts of noise are added to these time series to give n repetitions with the amount of noise classified as "small" (s), "medium" (m), "large" (l) and "extra large" (xl). These repetitions form the basis of a cluster with centroid x and y so that the same significance level would be given under Algorithm 5.4 but the within cluster variation is increased.

By Algorithm 4.1, x causes y with significance level 0.7660 (2 lags) and 0.1693 (3 lags), whereas y causes x with significance 0.9949 (2 lags) and 0.9955 (3 lags). For C_a, 48.5% of variation is explained by the first principal components, with a further 23.2% explained by the second principal component.

Algorithm 5.6 is applied to these clusters where n = 50 and repeated 500 times, with the distribution of the significances plotted. This is applied to the case where one and two principal components of C_a are used, and the case where one principal component of each C_a and C_b are used.

5.5.1 Results

In Figures 5.3 - 5.5, the plots shown are the distributions of significances for the 500 repetitions with increasing addition of noise within the cluster from left to right. The top row shows the significances for C_a causing C_b with the bottom row showing the results for C_b causing C_a. Figure 5.3 uses the first principal component of C_a, Figure 5.2 uses the first two principal components of C_a with Figure 5.3 using the first principal components of both clusters C_a and C_b.

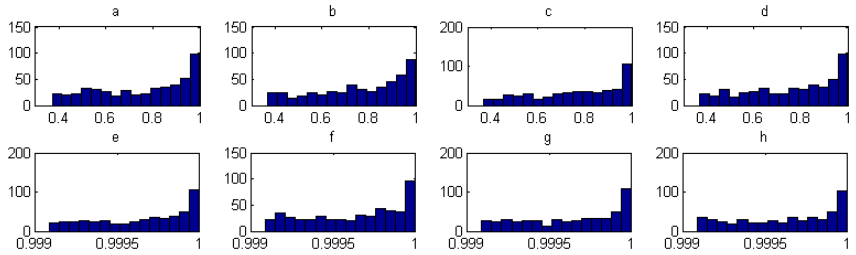

Figure 5.3: Histograms of significances for 50 time series within each cluster based on the first principal component of cluster C_a. a-d show the causal effect of the mean of cluster X on the mean of cluster Y where the noise around the true series is classified as small, medium, large and extra large successively. e-h show the same result with cluster Y causing cluster X.

Where there should be a significant interaction found for the centroids of C_b causing C_a, as the amount of noise increases within the cluster there seems to be little effect on the repeatedly high significances being found. However, as the number of principal components is increased there is a case for the the more significant interactions to be found more times. This is due to the extra information used in the regression to help explain more of the variability.

In the opposite direction, where C_a causes C_b, the spread of the distribution of significances is increased, providing a reasonable argument that the significance of interaction is not so strongly supported as in the other direction. However, it should be noted that there is still a tendency to find the interaction at a high level of significance. The explanation for this could be due to the data itself and is of great interest to consider further.

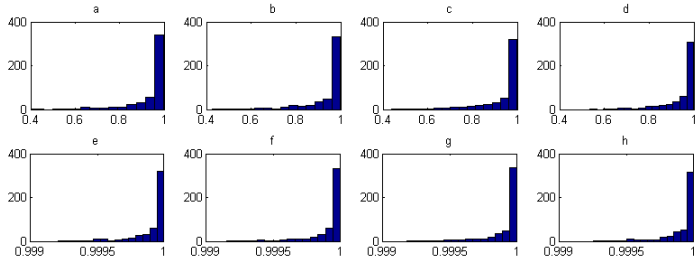

Figure 5.4: Histograms of significances for 50 time series within each cluster based on the first and second principal components of cluster C_a. a-d show the causal effect of the mean of cluster X on the mean of cluster Y where the noise around the true series is classified as small, medium, large and extra large successively. e-h show the same result with cluster Y causing cluster X.

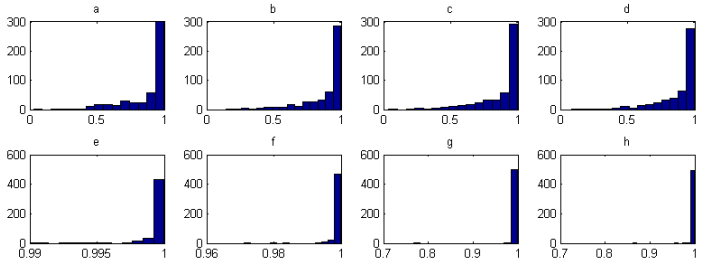

Figure 5.5: Histograms of significances for 50 time series within each cluster based on the first principal component of both clusters C_a and C_b. a-d show the causal effect of the mean of cluster X on the mean of cluster Y where the noise around the true series is classified as small, medium, large and extra large successively. e-h show the same result with cluster Y causing cluster X.

Overall, the use of principal components may help provide extra information where there is low amount of variation within clusters. This can then be used with Algorithm 5.5 and Algorithm 5.5a to help discover the most significant interactions from microarray data.

CHAPTER 6

Application to Observed Data

In the previous two chapters, methods for inferring genetic networks based on Granger causality and extended to include data reduction techniques were presented and their effectiveness considered with a view to understanding which interactions are statistically significant. This is all very well for simulated data where the data may have a well known structure, but in real life situations this may not always be the case. This chapter applies to the algorithms developed in the past two chapters to an observed dataset for a real biological network which is already partially described. Reconstruction of this network is the first aim with the further aim of predicting which interactions would be likely candidates for biological verification.

6.1 Xenopus Laevis Dataset

Data has been obtained from microarray experiments at the developmental stage of the African Clawed frog, Xenopus Laevis. Measurements were taken during this phase from 5 to 16 hours inclusively post fertilisation at hour intervals to give 12 time points of data. These measurements are made on prefabricated Affymetrix microarray slides which measure 15611 genes in total, of which 52 are signalling molecules and 529 are transcription factors.

The raw data has been normalised by the use of the Excel add-in BRB-Array Tools [16], giving log-normalised values for the basis of the subsequent analysis.

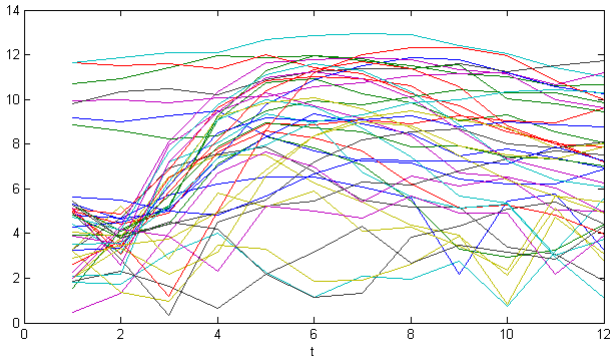

Figure 6.1: Observed time series for 42 genes from the Xenopus Laevis dataset at 12 timepoints.

6.2 Subnetwork evaluation

Analysis is performed firstly on a subnetwork of the overall data available for mesendoderm formation, as from in Figure 1.2 with the interactions described in Table 6.1 and taken from the paper by Loose and Patient [3]. This provides a useful starting point for two key reasons. Firstly, there are only 42 transcriptions factors to analyse so the computation is quick and easy to perform without requiring data reduction techniques and introducing any variability from this. Secondly, many interactions within this subnetwork are already known so their presence can be easily tested as well as providing suitable candidate interactions. These 77 known interactions can be then used as the basis for further biological measurement and verification. Figure 6.1 shows the observed time series for these genes over the 12 timepoints measured.

6.2.1 Results

The results of applying Granger causality algorithms and the variations as described in Chapter 4 are given in table 6.2 . Here direct Granger causality (Algorithm 4.1), Granger causality and the modified Granger algorithm by Hatemi et al (Algorithm 4.3) are applied to the data. The existing known network is assumed to be the true network and results are obtained by comparing against this network. The results show how many of the known interactions in the true network are recovered, how many interactions are shown as significant and the similarity measure of the true network and recovered network. These values

Table 6.1: Mesendoderm subnetwork interactions from Xenopus Laevis

Gene	Targets	Targeted By
Maternal Regulators		
Fast1	Brachyury, Chordin, Goosecoid, Xfkh1, Xlim1	
Sox3	Xnr5	
VegT	Bix1, Bix2, Bix3, Bix4, Brachyury, Cerberus	
	Derriere, Endodermin, Fgf3, Fgf8, Gata4, Gata5	
	Gata6, Goosecoid, Mix1, Mix2, Mixer, Sox17, Xenf	
	Xhex, Xnr1, Xnr2, Xnr4, Xnr5, Xnr6, Xwnt8	
Vg1	Xenf, Xnr1, Xnr2	
Pre-MBT		
β-Catenin	Cerberus, Siamois, Xhex, Xnr3, Xnr5, Xnr6	
Xnr5	Brachyury, Derriere, Pitx2, Xnr1, Xnr2	β-Catenin, Sox3, VegT
Xnr6	Brachyury, Derriere, Pitx2, Xnr1, Xnr2	β-Catenin, VegT
MBT - Stage 9		
Brachyury	Bix1, Bix2, Bix3, Bix4, Brachyury, Efgf	Brachyury, Efgf, Fast1, Goosecoid
	Xbtg1, Xegr1, Xwnt11	Mix1 Sip1, VegT, Xnr1, Xnr5, Xnr6
Derriere	Bix1, Cerberus, Derriere, Mix1, Mix2	Derriere, VegT, Xnr5, Xnr6
Eomes		
Pitx2		Goosecoid, Xnr1, Xnr5, Xnr6
Siamois	Goosecoid	β-Catenin
Sip1	Brachyury	
Sox17	Endodermin, Goosecoid, Hnf1	Goosecoid, Mixer, VegT, Xnr2
Xfkh1		Fast1, Xnr2
Xnr1	Brachyury, Pitx2, Xnr1	VegT, Vg1, Xnr1, Xnr5, Xnr6
Xnr2	Antivin, Bix4, Cerberus, Endodermin, Eomes, Gata4	VegT, Vg1, Xnr1, Xnr5, Xnr6
	Gata5, Gata6, Mix1, Mix2, Mixer, Sox17	
	Xnr2, Xfkh1	
Xnr3		β-catenin
Xnr4	Xnr4	VegT, Xnr4
Xwnt8		Goosecoid, VegT
Dorsal Anterior		
Goosecoid	Brachyury, Goosecoid, Pitx2, Sox17, Xwnt8	Fast1, Goosecoid, Mix1, Mixer, Siamois
		Sox17, VegT
Stage 9		
Antivin/Lefty1		Xnr2
Bix1/Mix4		Brachyury, Derriere, VegT
Bix2/Milk		Brachyury, VegT
Bix3		Brachyury, VegT
Bix4		Brachury, VegT, Xnr2
Cerberus		
Chordin		Fast1, Mix1
Efgf	Brachyury	Brachyury
Endodermin		Mixer, Sox17, VegT, Xnr2
Fgf3		VegT
Fgf8		VegT
Gata4		VegT, Xnr2
Gata5		Mixer, VegT, Xnr2
Gata6		VegT, Xnr2
Hnf1		Sox17
Mix1	Brachyury, Chordin, Goosecoid	Derriere, VegT, Xnr2
Mix2		Derriere, VegT, Xnr2
Mixer	Cerberus, Endodermin, Gata5, Goosecoid, Sox17	VegT, Xnr2
Xbtg1		Brachyury
Xegr1		Brachyury
Xenf		VegT, Vg1
Xhex		β-Catenin, VegT
Xlim1		Fast1
Xwnt11		Brachyury

are shown for three different significance levels at 95%, 99% and 99.5%. The non-overlap block bootstrap uses blocks of size 3 and all bootstrapping algorithms use 250 bootstraps.

The results show that generally network reconstruction is relatively poor. Although the similarity measures are reasonably good, as explained in the derivation of them, they allow for where an interaction affect is correctly identified as not significant. The application of bootstrapping to the Granger causality algorithm shows some slight improvment with the residual bootstrapping algorithm performing best out of all those used. The Hatemi algorithm shows poor performance although again here the use of bootstrapping shows some slight improvement than without.

6.2.2 Predicted Interactions

In complement to recovering those interactions already known, the mesendoderm subnetwork is not fully understood and explored so the predictability of other likely interactions is of interest. Interactions predicted under the Granger causality algorithm 4.1 is shown in Table 6.3 with the direction of interaction from and to given along with whether the interaction is already known.

From these it is shown that two interactions are already known, with Derriere known to target Bix1 and showing significant targeting of Bix3 and Bix4 genes. These results are ranked in order of significance from the most significant. Of particular interest to note is that the same genes are shown as repeatedly most targeted, the Bix family of genes Bix2, Bix3 and Bix4.

6.3 Clustering Transcription Factors

Whilst the subnetwork shows how the algorithms recover known interactions in a network, a subsection of the full dataset is now considered by combining together the 529 transcriptions factors the 52 signalling molecules into a single

Table 6.2: Mesendoderm network recovery

Significance Level	99.5%			99%			95%		
Algorithm	True	Significant	Similarity	True	Significant	Similarity	True	Significant	Similarity
Granger	12	24	0.1383	17	38	0.1124	35	52	0.0845
Granger Bootstrap									
Residual	15	28	0.1106	23	44	0.1044	30	54	0.0655
Non-overlap Block	14	25	0.1225	22	39	0.1085	30	57	0.0683
Hatemi									
No bootstrap	10	14	0.1644	15	30	0.1465	19	42	0.1145
Residual bootstrap	12	18	0.1428	13	28	0.1429	18	40	0.1189

Table 6.3: Predicted Interactions for mesendoderm subnetwork

Ranking	From	To	Known
1	Cerberus	Bix3	
2	Cerberus	Bix4	
3	Brach	Bix3	Y
4	Brach	Bix4	Y
5	Derriere	Bix3	Bix1
6	Derriere	Bix4	Bix1
7	Xenf	Bix3	
8	Xenf	Bix4	
9	Chordin	Bix3	
10	Chordin	Bix4	
11	Gata6	Bix3	
12	Gata6	Bix4	
13	Gata5	Bix3	
14	Gata5	Bix4	
15	Gata5	Bix2	
16	Gata6	Bix2	
17	Chordin	Bix2	
18	Fgf3	Bix3	
19	Fgf3	Bix4	
20	Gata4	Bix2	

dataset. This focuses the inference for a particular type of interaction which is of greatest interest for biological measurements and also provides a suitable upscaling of the previous subnetwork which is built only upon transcription factors.

For Algorithm 5.4, where individual pairings do not require bootstrapping, the algorithm is applied directly. However with the increase in the number of individual genes, clustering algorithms are also to be applied for dimension reduction and to gain improvements in speed of computation. This will also provide a comparison of how well the clustering compares to the direct algorithm.

As now the subnetwork is much larger and would require more computational

time to implement, the data reduction techniques introduced in the previous chapter are used for prediction of significant interactions. Table 6.4 shows the results where the number of runs is fixed at 500 and the count of the significances used for an $\alpha = 0.99$ level. This shows a good amount of overlap for the most significant interactions, supporting that these interactions are repeatedly significant.

Table 6.4: Transcription Factor predicted interactions with Granger causality

from	to
gabpa-A	LOC496377
HoxA1	MGC68588
Bix1	lim2b-A
XSUG	stat3-A
lim5/Lhx5	mafB
lhx2-A	zax-A
LOC398730	nrl20-A
TRH4	AR
cbfa2t2-a	MGC53355
lhx2-A	Lmx1b
nr3-A	PPARg
otx5-A	en1-A
MGC68543	Xlim-3
Clk	nr6a1
hif1a	nkx2-10-A
Bix1	hoxd10
taf10b	HoxA1
atf4-ii	fgf9-A
foxd1-A	gabpa-A
HoxD1	thr

6.3.1 Results

The data was clustered to between 10 and 30 clusters inclusive in steps of 5 using the k-means algorithm due to the simplicity and speed of computation. Using Algorithm 5.5, the standard Granger causality was applied on the means of the clusters to create the significance value for each possible interaction. This was repeated 500 times. From these the summation of the significances was taken to rank the most repeated significant interactions with the results for the top 20 interactions shown in table 6.5.

From these, the ten most consistently significant interactions are given in Table 6.6, by taking the average of the overall sum significances for each of the cluster sizes used. It is then these interactions which should be considered for further experimental verification, in particular the HoxA1-RAB18 interaction.

Table 6.5: Transcription factor most significant interactions changing number of clusters

Clusters	20		30		40		50		60		70	
Ranking	From	To	From	To	From	To	From	To	From	To	From	To
1	hoxb5-A	xhoxc8	HoxA1	RAB18	rxra-A	LOC397824	Mef2	LOC397824	LOC496325	LOC397824	tbx4-A	—
2	nkx3-1	MGC131119	bra3-a	foxn5	sip1	Pax3	rxra-A	LOC397824	sip1	Pax3	chi	hoxa11
3	Dlx5	nkx2.5	vax1-A	myc	MGC114753	tcf	sip1	Pax3	MGC114753	tcf	rela-a	LOC398730
4	odag-pending	xFF1r	hoxa13-A	xldb1	Mef2	LOC397824	MGC114753	tcf	Mef2	LOC397824	Xlim-3	LOC398167
5	fkh1-A	LOC397761	POU 2	Mdk	thibz-a	MGC68543	thibz-a	MGC68543	thibz-a	MGC68543	lft-a	MGC68691
6	DOR2	Rx2A	dnmt1	myc	hoxa13-A	xldb1	hoxa13-A	xldb1	hoxa13-A	xldb1	vax1-A	barh2-a
7	rel-A	Hoxa2	hoxa3a	gabpa-A	vax1-A	myc	HoxA1	RAB18	HoxA1	RAB18	Mef2	LOC397824
8	XIQGAP2	MGC52531	stat3-A	MGC81762	HoxA1	RAB18	vax1-A	myc	vax1-A	myc	dnmt1	tcf
9	MGC68588	brn3d-A	LOC397942	LOC397877	Hoxa2	MGC83056	Hoxa2	MGC83056	rela-a	LOC398730	bix2-A	gene 7
10	LOC397936	LOC10003691	MGC52531	irf2	rxra-A	MGC80584	vax1-A	MGC80584	tbx4-A	—	LOC496325	LOC397824
11	MGC52531	MGC80854	Hoxa2	MGC83056	hoxa3a	gabpa-A	lhx7-B	Xhox3	lft-a	MGC68691	sip1	Pax3
12	pax5-a	NCA	XFD2	tfiiealpha	LOC397942	LOC397877	LOC397942	LOC397877	bix2-A	gene 7	fkh1-A	LOC397761
13	TFIIAa/b-1	Nkx2-4	XGATA-3	MGC114733	hlxb9-A	RAB35	bra3-a	gabpa-A	chi	hoxa11	hoxa13-A	xldb1
14	apeg-A	MGC83414	MGC52531	XSUG	lhx7-B	Xhox3	hlxb9-A	RAB35	Xlim-3	LOC398167	thibz-a	MGC68543
15	MGC84465	MGC131139	rxra-A	MGC80584	GATA-5a	nr6a1	MGC52531	XSUG	MGC52531	XSUG	HoxA1	RAB18
16	wt1-B	koza-A	hlxb9-A	RAB35	MGC52531	XSUG	rab7	LOC10012665	Hoxa2	MGC83056	vax1-A	myc
17	tbx4-A	MGC52531	tef	Mta2	rab7	LOC10012665	GATA-5a	nr6a1	rxra-A	MGC80584	XGATA-3	MGC114733
18	wt1-A / wt1-B	pitx1-A	rab7	LOC10012665	XFD2	tfiiealpha	XFD2	tfiiealpha	Nr2f1	barh2-a	bra3-a	foxn5
19	tbx20-A	Mad2	xsmad4a	TFIIDtau	tef	Mta2	tef	Mta2	hoxa3a	gabpa-A	rxra-A	MGC80584
20	hoxa11	LOC503680	otx1-A	LOC10003685	POU 2	Mdk	cart1-A	—	LOC397942	LOC397877	hoxa3a	gabpa-A

Table 6.6: Transcription factor network most highly predicted interactions

Rank	From	To
1	HoxA1	RAB18
2	vax1-A	myc
3	rxra-A	MGC80584
4	Hoxa2	MGC83056
5	sip1	Pax3
6	hoxa13-A	xldb1
7	MGC114753	tcf
8	MGC52531	XSUG
9	chi	hoxa11
10	GATA-5a	nr6a1

6.3.2 Principal Components

The same dataset for transcription factors is used to observe the effects of the addition of principal components, as outlined in section 5.3 and implmeneted in algorithm 5.6. Here, the number of clusters is fixed at 30 for 500 repetitions with summation of significances used as the measure of overall significance for interactions. Principal components are considered by the addition of the first principal component for both clusters used in the algorithm. Table 6.7 shows the twenty most significant interactions ranked in order of overall significance for the control and with the addition of principal components.

Table 6.7: Transcription Factor network interactions with principal components

	Without PC		With PC	
Rank	From	To	From	To
1	HoxA1	RAB18	thibz-a	tcf
2	bra3-a	foxn5	rxra-A	LOC397824
3	vax1-A	myc	sip1	Pax3
4	hoxa13-A	xldb1	XER81	LOC397824
5	POU 2	Mdk	thibz-a	MGC68543
6	dnmt1	myc	tbx4-A	—
7	hoxa3a	gabpa-A	lft-a	MGC68691
8	stat3-A	MGC81762	LOC397778	LOC398730
9	LOC397942	LOC397877	Xlim-3	LOC398167
10	MGC52531	irf2	xldb1	hoxa11
11	Hoxa2	MGC83056	bix2-A	gene 7
12	XFD2	tfiiealpha	tbx5-B	barh2-a
13	XGATA-3	MGC114733	fkh1-A	LOC397761
14	MGC52531	XSUG	hoxa13-A	xldb1
15	rxra-A	MGC80584	HoxA1	RAB18
16	hlxb9-A	RAB35	Hoxa2	MGC83056
17	tef	Mta2	rxra-A	MGC80584
17	rab7	LOC10012665	vax1-A	myc
19	xsmad4a	TFIIDtau	bra3-a	gabpa-A
20	otx1-A	LOC10003685	hlxb9-A	RAB35

6.4 Discussion of Results

When considering recovery of known existing networks, such as the mesendoderm subnetwork for Xenopus Laevis as shown, the results show that the ability of the algorithms used to successfully recover the network to be poor. One particular caveat should be placed on this, however, that this assumes the known network is indeed truly fully known and this is as given. The only way to fully know whether this network is indeed true is by experimentally measuring every possible interaction which would be costly and time consuming. With such limitation in mind, the results assuming this as a known network should therefore be approached with some caution.

Bootstrapping applied to the Granger causality methods shows some very slight improvement. The improvement cannot be described as significant under the conditions assumed, as the extra number of interactions found is very few. Similarly, the use of the alternative representation of the Granger causality algorithm shows a slight deterioration in performance but this cannot be seen to be significant due to the low level or interactions recovered by all algorithms.

The conclusion to this would be that bootstrapping within the Granger causality framework may provide some benefit for recovery of networks. One particular difficulty as well is the use of such limited data from the very low number of timepoints. This leads to high variability in parameter estimates and the interactions are seen to show very high levels of significance where the interactions are indeed significant but also variable values of significance where the interaction is not significant.

Looking at prediction of interactions, the mesendoderm network shows a perhaps surprising results, where the Bix family of genes is highly significant as a targeted gene. Whilst Derriere is known to already target the Bix1 gene, this extends naturally to provide candidate interactions with other members of the Bix family of genes. The other targeting genes show that Bix genes are highly targeted from many other genes and further investigation would be warranted into this area.

Extending the prediction of interactions to the larger subset of all transcription

factors and signalling molecules, clustering techniques were applied. These results are compared against the interactions without any form of clustering in order to see how the results vary. The most 20 significant interactions in each case are shown. Where the cluster sizes are varied, there is a broad amount of overlap for the cluster sizes from 30 to 60. At cluster size 20, there is a significant deviation, most likely due to the variation within the clusters.

Table 6.6 shows the ten most consistent highly predicted interactions from clustering across all clusters. It is these interactions that would be likely candidates for experimental verification of interaction.

The addition of principal components, as shown in Table 6.7, shows a certain amount of agreement in terms of the interactions predicted for a fixed number of clusters. Further to this, by comparison to table 6.5, it shows more consistent prediction with the greater number of clusters. This could lead to the interpretation that the presence of extra information of the structure of the clusters, such as by principal components, gives rise to extra insight into the performance of clusters. At lower cluster numbers, the ability to replicate higher cluster numbers without loss of information could give a significant improvement in terms of speed and calculation.

CHAPTER 7

Verification from other data

With the previous chapter, the algorithms developed and described in the preceding chapters were implemented for an experimentally observed dataset. From this, interactions were predicted that would be good candidates for experimental verification. However, it would be hoped that experimentally verified interactions should achieve high rankings given existing knowledge that such an interaction should exist. With the interaction of many genes possible for single targets this is clearly a difficult challenge.

This chapter explores the use of other data sources to support the decision to undertake a full scale experimental verification of predicted interactions. Data is provided for a particular gene, the caudal-type homeobox transcription factor CDX4 (also known as Xcad-3), under two sets of conditions: one where the gene is present and another where the gene is switched off by use of a gene knockout. Under conditions described, this should provide some information as to whether there is an interaction of interest with this particular gene targeting other genes. A known interaction with a particular family of genes is also explored for verification.

7.1 Gene Knockout Data

Data has been provided for the Xenopus Laevis chip as used in the previous analysis. There are four types of data given. Firstly, a control, where nothing has been altered and a single reading taken. Secondly, another single reading taken but this time with the CDX4 gene knocked out or silenced. Two further

pieces of data are given as well. One is for the addition of VP16, which acts as a superactivator and should increase the activity of any gene. The other is for the addition of ENR which acts as an inhibitor and hence decrease the activity of any gene.

To interpret the results, when the CDX4 gene is knocked out, if this value is less than control value it supports that CDX4 activates this gene as with the presence of CDX4 the control value is higher. The significance of this is most important and the subsequent analysis considers the variation in measuring what a significant change is.

For reference purposes, it is helpful to be able to identify known existing interactions. One pathway that is well identified is the FGF-CDX-Hox pathway [237]. From this, it is known that CDX4 in particular should target the Hox family of genes. For reference, we focus on one well identified interaction on Hox7a/Hox36.

7.1.1 Results

Firstly, the genes where the control signal is less than 100 are discarded as signal levels at this range are highly prone to noise in the detection. From the full dataset of 15611, this leaves 9626 genes. Similarly from the 611 transcription factors, 319 are above this level.

From these, at the 50% change level with deletion of CDX4 compared to the control and with VP16 levels being greater than the control and ENR level being less than the control, 114 genes are selected of which 17 are transcription factors. Similarly, at the 25% level, 479 genes are selected of which 32 are transcription factors.

For the 17 transcription factor found at the 50% change level, Table 7.1 shows the genes targeted and their consistent count rank for the results used in section 6.3, based across varying amounts of clustering. This shows that indeed Hox genes are well targeted and the data supports this, but the range of rankings is quite variable, from a ranking of 11 for Xhox3 down to 438 for hoxa9-A. However, generally the Hox genes are well predicted considering the overall

Table 7.1: CDX4 targets where 50% change level of CDX4 detected compared to control

Target	Count Rank
hoxa3a	248
stat3-A	395
hox36	84
Xhox3	11
Xvex-1	469
xCAD2	582
XlHbox1	438
Xombi	337
MGC131107	20
hoxd10	83
Hoxb7	103
LOC398337	100
hoxa11	347
hox36	43
hoxa9-A	438
vg1	227
MGC154472	24

number of possible interactions that would be measurable across all genes in the transcription factor dataset. This helps to support that the Algorithm 5.5 developed using Granger causality applied to clustering data is useful in terms of providing predictions for interactions.

In particular, where CDX4 targets Hox7a/Hox36, the change level is measured above 50% and the interaction ranked as 43. Given that this interaction has directly been measured and is known, this strengthens the support for using gene knockout data in combination with the Granger causality algorithms developed prior.

7.2 Known interactions

In order to verify the the results of the previous chapter, using existing known interactions can provide a mechanism to see how these perform under the algorithms used. One known pathway is that of the eFGF-CDX-Hox pathway, as described in Pownall et al [237]. Here the CDX family of genes is known to target the Hox family of genes, and one particular gene from each family is well described, that of the CDX4 gene targeting Hox36 (also known as Hox7a). These genes exist as transcription factors within Xenopus Laevis so the dataset obtained and analysed can be assessed for performance.

As CDX4 is known to target one gene in particular, Hox36, the comparison can be made also against the gene knockout data from the previous section. As such, the use of both datasets can be measured against known interactions to assess the use of extra data.

7.2.1 CDX4 - Hox36 Interaction

The Xenopus Laevis transcription factor dataset is used, as analysed in section 6.3.1, with the cluster size increasing from 20 to 75 in increments of 5. The overall significance is compared from the count of the individual significances as well as the count variation, with a count of 1 for a significant interaction at the 99.5% level. The average significance level is provided for 250 repetitions.

The results for the CDX4-Hox36 interaction are shown in Table 7.2, with the number of clusters showing the rank under the sum and count variations of the overall significance.

For the full dataset clustering, only 1000 clusters were used with 25 repetitions due to the time taken to cluster and analyse such large amounts of interactions. For this, the average rankings for the count and sum overall significances were 1086 and 244 respectively.

Table 7.2: CDX4-Hox36 interaction rankings out of interactions based on transcription factors alone

No. of clusters	Sum Rank	Count Rank
20	149	22
25	170	31
30	129	43
35	85	18
40	86	23
45	40	29
50	83	48
55	77	19
60	198	50
65	415	30
70	426	43
75	386	26

7.2.2 eFGF - CDX4 Interaction

The other interaction shown by the Pownall et al paper is that of the eFGF-CDX4 interaction, both of which appear in the Xenopus Laevis dataset. The data for this interaction from both the transcription factor and the full datasets can therefore be given and is presented in Table 7.3 as for the CDX4-Hox36 interaction.

7.2.3 Results

The CDX4-Hox36 interaction shows good predictability especially under the count ranking method of significances, with results consistently in the top 100 rankings. It performs less well where a sum is assigned based on a 99.5% level of significance for each interaction. However, the results still here are relatively high compared against the total number of possible interactions.

For the eFGF-CDX4 interaction, there is less support for the interaction within the analysis performed, yet again the count rank shows better performance than

Table 7.3: eFGF-CDX4 interaction rankings out of interactions based on transcription factors alone

No. of clusters	Sum Rank	Count Rank
20	1385	233
25	875	205
30	633	195
35	749	226
40	724	198
45	689	212
50	534	221
55	785	164
60	895	185
65	587	192
70	406	144
75	745	168

that of the sum rank. In comparison against all possible interactions, these results are still relatively well performing.

Overall, these two known interactions help to show that the use of clustering with the Granger algorithm from Algorithm 5.4 can be used to predict interactions in genetic networks, even though the data may be limited. One problem lies choosing which interactions are deemed most significant; clearly if the most 20 significant interactions were chosen as contenders for experimental verification, it would potentially miss interactions such as these ones that are known. So whilst Table 6.6 shows the top ten most significant predicted interactions, care needs to be taken to ensure that these are indeed interactions worth persuing.

By combining the knockout data in the previous section, this will help allow to ensure that these predicted significant interactions are indeed worthy of further investigation.

CHAPTER 8

Discussion

The field of genetic network modelling and inference has evolved greatly with the wide range of statistical and applied mathematical techniques that are available. Research groups and sections of well regarded journals, such as Bioinformatics, are devoted to the area and making sense of the huge amounts of experimental data produced. This thesis has considered the origins of modelling genetic networks and how this leads to using statistical approaches to infer interactions between genes in a biological system. This chapter summarises the findings of this thesis and presents some considerations on how to expand and develop on some of the issues and ideas raised.

8.1 Conclusions

Chapter 1 introduced the underlying biological mechanism being studied and what a genetic network means. The use of microarrays to obtain data is presented and the challenges of obtaining this sort of data due to the inherent noise in measurement. A survey of literature for modelling and recovering genetic networks is given which shows the great range and scope of a multitude of techniques used. This review is by no means exhaustive, with a lot of active research into the area generating more literature than could ever be fully described in a single thesis. However, it presents some of the key ideas and approaches that have been used and the variations thereon.

Chapter 2 looks at how modelling of genetic networks has developed, from the

simple binary models of Kauffman, to more developed stochastic models. Biologically observed features, such as competition of binary decisions and multilineage priming, are shown to verify the validity of using such models. Modelling a stochastic system at the molecular level, by stochastic simulation algorithms and the Gillespie algorithm, are shown. These techniques can be used to build artificial networks and understand their dynamics and generate artificial data.

Chapter 3 moves away from modelling genetic networks and looks at the reverse view, of how to reconstruct a genetic network from data provided. Simple least square algorithms are introduced, which are unlikely to be feasible across very large networks. The use of Bayesian networks, currently a widely used technique, is also discussed.

Arguments are presented for using a single timepoint with multiple measurements as this will only give limited information in stochastic system that evolves over time. Further to this, a correlation measure is used for inference between nodes which fails to take into account the directionality that is present in genetic networks. For these two reasons, this approach does not seem to be useful as it neglects arguably the most two key features of a genetic network.

Chapter 4 presents the main underlying technique used to assess the interaction of genes within a genetic network, that of Granger causality. The use of such a statistical technique is appropriate, as it provides a statistical means to assess a directional effect between two genes on data measured over time, the form in which the most useful data is provided. Bootstrapping of multivariate time series is introduced, with the overlapping and non-overlapping block bootstraps and the residual bootstraps compared. The block bootstraps are compared and shown to perform reasonably similarly when considering both the size of block length and the amount of overlap. The residual bootstrap generally outperforms the block bootstraps and require less parameters to consider so is used in subsequent analysis. These are further integrated into the Granger causality approach and three algorithms compared. It is shown that direct Granger causality (Algorithm 4.1) and the Hatemi-Shukur algorithm (Algorithm 4.3) perform well, with the Hidalgo algorithm (Algorithm 4.2) based in the frequency do-

main shown to perform poorly.

The paper by Chatterjee and Mukhophadyay [2] is evaluated with the decision to only take the most significant directional interaction between two genes discouraged. Algorithms for recovering a genetic network on this basis are introduced which can in turn be used to predict the most significant interactions. Performance is shown to be good when used with stationary datasets and improvements for longer time series, with non-stationary data showing generally poor performance.

Chapter 5 develops the application of Granger causality to very large datasets where all possible interactions may be too time consuming to calculate. Clustering is used in order to speed up calculation, with two particular clustering algorithms given. The QT-algorithm is discussed with relation to the choice of threshold parameter. A small threshold recovers networks well but soon falls away the larger this threshold becomes. The k-means algorithm is chosen due to ease and speed of computation, even for large amounts of data. Original algorithms are developed in order to combine clustering with Granger causality to meaningfully ascertain which interactions are most significant, by considering repetitions of the clustering algorithm.

The variation within the clusters is considered by use of Principal Components to help explain the variation. This is combined with the clustered Granger causality algorithm and shown that the first few principal components are useful where a true interaction may exist.

Chapter 6 applies these newly developed algorithms to the case of an observed dataset in order to understand their performance in a real life setting. A subset of the data where some of the network interactions are known is used to attempt to recover this network and give an indication of which interactions may be of interest to further experimentally verify.

A larger subset of signalling molecules and transcription factors is used with clustering and provides some interactions of interest for further verification. Certain interactions are consistently marked as significant, regardless of cluster size. The full data set is clustered to again find interactions of interest. Some of

these are in common with the previous subset, giving weight to these interactions as being the best candidates.

Chapter 7 verifies those interactions that target a particular gene, CDX4, by use of gene knockouts. One particular known pathway, for Hox genes targeting CDX genes, is considered to show that the data obtained is indeed useful. This is combined with the fact that some of these interactions are shown to be highly significant by using the clustered Granger causality algorithms. By combining the results from gene knockout data with the clustered Granger causality algorithms, this helps support which predicted interactions are indeed most likely to be shown as significant.

8.2 Further Study

Application of mathematical and statistical techniques to genetic networks is wide and varied, with a wide range of research being undertaken with the increase of genomic data and applications. As the experimental technology develops, the data obtained can be used to produce better results.

One such technology is the protein assay. This is the protein equivalent of the microarray to measure levels of proteins within a sample as opposed to mRNA. Due to the size and specificity of proteins, the development of technologies to measure such data has encountered significant difficulties. As this improves, the level of protein can be used as well as that of the mRNA, such as as that used in the ODE models given in Chapter 2. Given the current assumption that the level of mRNA and that of proteins is linear, this relationship itself can be explored to better understand and parameterise a genetic network.

Within the context of the causality methods described and developed in this thesis, one key factor has been the limited amount of data measured due to the high cost and laboratory time taken to extract the data. As gene chips become cheaper and faster to produce and analyse, longer range time series data will

be rapidly available, possibly over many tens or even hundreds of time points. Such extra information will be crucial for providing better estimates of parameters to obtain better reconstruction of networks.

Coupled with this increase in timepoints is the actual analysis itself, especially in the context where the data may not fulfil the criteria of the algorithms used, such as stationarity. One way to view this is to split the time series into parts and look at localised causality for each of these, already used in applications such as in Hesse et al [238] . Furthermore, extra information for the model can be obtained by taking multiple samples and combining these to understand variation in the model. Use of non-linear methods for parameter estimation within a Granger causality framework may be of use here [239] [240].

One technique to improve speed of computation is through the use of parallel computing, whereby many processors are able to individually perform computations before being rejoined together. In fact, pairwise analysis is highly suitable to this, as each pairwise interaction is a completely separate action not requiring the knowledge of any other interactions. Such embarrassingly parallel problems are therefore ripe for development with parallel computing allowing very large datasets to be easily and manageably analysed.

For the algorithms developed within this thesis, these can be refined and improved in various ways. Principal components are used to understand the affect of within cluster variation and how it affects the algorithm. One extension to this is to use multivariate Granger causality between the full clusters.

The variation in clustering algorithms would also be worthwhile investigating, such as the QT algorithm. As the computational power of processors increases, such algorithms may provide a more useful insight than the clustering techniques that are sensitive to intitial conditions.

References

[1] S. A. Kauffman. *The Origins of Order*. Oxford University Press, Oxford, 1993.

[2] N. D. Mukhopadhyay and S. Chatterjee. Causality and pathway search in microarray time series experiment. *Bioinformatics*, 23(4):442–449, 2007.

[3] M. Loose and R. Patient. A genetic regulatory network for xenopus mesendoderm formation. *Developmental Biology*, 271:467–478, 2004.

[4] B. Lewin. *Genes IX*. Oxford University Press, Oxford-New York, 2007.

[5] D.S. Latchman. *Gene Regulation*. Taylor & Francis, 2005.

[6] D.J. Stewart. Making and using DNA microarrays: A short course at Cold Spring Harbor Laboratory. *Genome Res.*, 10:1–3, 2000.

[7] Affymetrix. The GeneChip System: An integrated solution for expression and DNA analysis, 2005.

[8] R. A. Irizarry, Z. Wu, and H. A. Jaffee. Comparison of Affymetrix GeneChip expression measures. *Bioinformatics*, 22(7):789–794, 2006.

[9] S. O. Zakharkin, K. Kim, T. Mehta, L. Chen, S. Barnes, K. E. Scheirer, R. S. Parrish, D. B. Allison, and G.P. Page. Sources of variation in Affymetrix microarray experiments. *BMC Bioinformatics*, 6:214, 2005.

[10] D. Amaratunga and J. Cabrera. *Exploration and Analysis of DNA Microarray and Protein Array Data*. Wiley, 2004.

[11] H. Causton, J. Quackenbush, and A. Brazma. *Microarray Gene Expression Data Analysis: A Beginner's Guide*. Blackwell Science, 2003.

[12] S. Knudsen. *A Biologist's Guide to Analysis of DNA Microarray Data*. Wiley, 2002.

[13] D. Stekel. *Microarray Bioinformatics*. CUP, 2003.

[14] E. Wit and J. McClure. *Statistics for Microarrays: Design, Analysis and Inference*. Wiley, 2004.

[15] T. P. Speed, editor. *Statistical Analysis of Gene Expression Microarray Data.* Chapman And Hall, March 2003.

[16] R. Simon. Analysis of gene expression data using BRB-array tools. *Cancer Informatics*, 3:11–17, 2007.

[17] S. Kauffman, C. Peterson, B. Samuelsson, and C. Troein. Genetic networks with canalyzing Boolean rules are always stable. *PNAS*, 101(49):17102–17107, 2004.

[18] J.M. Bower and H. Bolouri, editors. *Computational Modeling of Genetic and Biochemical Networks*. MIT Press, 2001.

[19] E. R. Dougherty and I. Shmulevich. Mappings between probabilistic boolean networks. *Signal Processing*, 83(4):799–809, 2003.

[20] I. Ivanov and E.R. Dougherty. Reduction mappings between probabilistic boolean networks. *EURASIP Journal on Applied Signal Processing*, 1:125–131, 2004.

[21] I. Shmulevich, E.R. Dougherty, and W. Zhang. From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE*, 90(11):1778–1792, 2002.

[22] I. Shmulevich, I. Gluhovsky, R. F. Hashimoto, and E. R. Dougherty. Steady-state analysis of genetic regulatory networks modelled by probabilistic Boolean networks. *Comp. Funct. Genom.*, 4:601–608, 2003.

[23] D. C. Weaver, C. T. Workman, and G. D. Stormo. Modeling regulatory networks with weight matrices. In *Pacific Symposium on Biocomputing*, pages 112–123, 1999.

[24] J. Vohradsky. Neural network model of gene expression. *FASEB Journal*, 15:846–854, 2001.

[25] L. F. A. Wessels, E. P. van Someren, and M. J. T. Reinders. A comparison of genetic network models. In *Pacific Symposium on Biocomputing*, pages 508–519, 2001.

[26] H. de Jong. Modeling and simulation of genetic regulatory networks. In *POSTA*, volume 294 of *Lecture Notes in Control and Information Sciences*, pages 111–118. Springer, 2003. ISBN 3-540-40342-6.

[27] P. Smolen, D.A. Baxter, and J.H. Byrne. Modeling transcriptional control in gene networks : Methods, recent results, and future directions. *Bulletin of Mathematical Biology*, 62:247–292, 2000.

[28] P. Laslo, C.J. Spooner, A. Warmflash, D.W. lancki, H.J. Lee, R. Sciammas, B.N. Gantner, A.R. Dinner, and H. Singh. Multilineage transcriptional priming and determination of alternate hematopoietic cell fates. *Cell*, 126: 755–766, 2006.

[29] I. Roeder and I. Glauche. Towards an understanding of lineage specification in hematopoietic stem cells: a mathematical model for the interaction of transcription factors GATA-1 and PU.1. *J. Theor. Biol.*, 241:852–65, 2006.

[30] M. Loose, R. Patient, and G. Swiers. Genetic regulatory networks programming hematopoietic stem cells and erythroid lineage specification. *Dev. Biol.*, 294:525–40, 2006.

[31] G.K. Ackers, A.D. Johnson, and M.A. Shea. Quantitative model for gene regulation by lambda phage repressor. *Proc. NatL Acad. Sci. USA*, 79:1129–1133, 1982.

[32] O. Cinquin and J. Demongeot. High-dimensional switches and the modelling of cellular differentiation. *Journal of Theoretical Biology*, 233:391–411, 2005.

[33] M. Santillan and M.C. Mackey. Why the lysogenic state of phage lambda is so stable: A mathematical modeling approach. *Biophysical Journal*, 86: 75–84, 2004.

[34] K. Lai, M.J. Robertson, and D.V. Schaffer. The sonic hedgehog signaling system as a bistable genetic switch. *Biophysical Journal*, 86:2748–2757, 2004.

[35] I. V. Deineko, A. E. Kel, O. V. Kel-Margoulis, E. Wingender, and V. A. Ratner. Simulation of the dynamics of gene networks regulating the cell cycle in mammalian cells. *Russian Journal of Genetics*, 39(9):1085–1091, 2003.

[36] S. Huang, G. Eichler, Y. Bar-Yam, and D. E. Ingber. Cell fates as high-dimensional attractor states of a complex gene regulatory network. *Physical Review Letters*, 94:1–4, 2005.

[37] V. Hatzimanikatis and K. H. Lee. Dynamical analysis of gene networks requires both mRNA and protein expression information. *Metabolic Engineering*, 1:275–281, 1999.

[38] R. Karmakar and I. Bose. Stochastic model of transcription factor-regulated gene expression. *Phys. Biol.*, 3:200–208, 2006.

[39] H. de Jong, J. Geiselmann, C. Hernandez, and M. Page. Genetic network analyzer: qualitative simulation of genetic regulatory networks. *Bioinformatics*, 19(3):336–344, 2003.

[40] T.T. Vu and J. Vohradsky. Genexp - a genetic network simulation environment. *Bioinformatics*, 18(10):1400–1401, 2002.

[41] T.B. Kepler and T.C. Elston. Stochasticity in transcriptional regulation: Origins, consequences and mathematical representations. *Biophysical Journal*, 81:3116–3136, 2001.

[42] H.H. McAdams and A. Arkin. Simulation of prokaryotic genetic circuits. *Annu. Rev. Biophys. Biomol. Struct.*, 27:199–224, 1998.

[43] H. H. McAdams and A. Arkin. Stochastic mechanisms in gene expression. *Proc. Natl. Acad. Sci. USA*, 94:814–819, 1997.

[44] J. M. Raser and E. K. OShea. Control of stochasticity in eukaryotic gene expression. *Science*, 304:1811–1814, 2004.

[45] J. M. Raser and E. K. OShea. Noise in gene expression: Origins, consequences and control. *Science*, 309:2010–2013, 2005.

[46] T. Tian and K. Burrage. Stochastic neural network models for gene regulatory networks. In *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003*, pages 162–169. IEEE Press, 2003. ISBN 0-7803-7804-0.

[47] C. Rao and A. Arkin. Stochastic chemical kinetics and the quasi-steadystate assumption: Applications to the Gillespie algorithm. *Journal of Chemical Physics*, 118:4999–5010, March 2003.

[48] K. Chen, T. Wang, H. Tseng, C. F. Huang, and C. Kao. A stochastic differential equation model for quantifying transcriptional regulatory network in saccharomyces cerevisiae. *Bioinformatics*, 21(12):2883–2890, 2005.

[49] Adam Arkin, John Ross, and Harley H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected escherichia coli cells. *Genetics*, 149:1633–1648, 1998.

[50] T. Toulouse, P. Ao, I. Shmulevich, and S. Kauffman. Noise in a small genetic circuit that undergoes bifurcation. *Complexity*, 11(1):45–51, 2005.

[51] D. Bratsun, D. Volfson, L.S. Tsimring, and J. Hasty. Delay-induced stochastic oscillations in gene regulation. *PNAS*, 102(41):14593–14598, 2005.

[52] W.J. Blake, M. Kaern, C.R. Cantor, and J.J. Collins. Noise in eukaryotic gene expression. *Nature*, 422:633–637, 2003.

[53] Juan M. Pedraza and Alexander van Oudenaarden. Noise propagation in gene networks. *Science*, 307:1965–1969, 2005.

[54] B. Chen and Y. Wang. On the attenuation and amplification of molecular noise in genetic regulatory networks. *BMC Bioinformatics*, 7:52, 2006.

[55] L. Chen, R. Wang, and K. Aihar. Genetic networks with stochastic fluctuations. *Genome Informatics*, 14:356–357, 2003.

[56] M.L. Simpson, C.D. Cox, and G.S. Sayler. Frequency domain analysis of noise in autoregulated gene circuits. *PNAS*, 100(8):4551–4556, 2003.

[57] M.L. Simpson, C.D. Cox, and G.S. Sayler. Frequency domain chemical langevin analysis of stochasticity in gene transcriptional regulation. *Journal of Theoretical Biology*, 229:383–394, 2004.

[58] R. Tomiokaa, H. Kimurab, T.J. Kobayashib, and K. Aihara. Multivariate analysis of noise in genetic regulatory networks. *Journal of Theoretical Biology*, 229:501–521, 2004.

[59] P.S. Swain, M.B. Elowitz, and E.D. Siggia. Intrinsic and extrinsic contributions to stochasticity in gene expression. *PNAS*, 99(20):12795–12800, 2002.

[60] Y. Tao. Intrinsic noise, gene regulation and steady-state statistics in a two-gene network. *Journal of Theoretical Biology*, 231:563–568, 2004.

[61] Y. Tao. Intrinsic and external noise in an auto-regulatory genetic network. *Journal of Theoretical Biology*, 229:147–156, 2004.

[62] M. Thattal and A. van Oudenaarden. Intrinsic noise in gene regulatory networks. *PNAS*, 98(15):8614–8619, 2001.

[63] H. Maamar, A. Raj, and D. Dubnau. Noise in gene expression determines cell fate in bacillus subtilis. *Science*, 317:526–529, 2007.

[64] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[65] P. M. Kim and B. Tidor. Limitations of quantitative gene regulation models: A case study. *Genome Research*, 13:2391–2395, 2003.

[66] A. Blais and B. D. Dynlacht. Constructing transcriptional regulatory networks. *Genes & Dev*, 19:1499–1511, 2005.

[67] S. Bornholdt. Modeling genetic networks and their evolution: A complex dynamical systems perspective. *Biol. Chem.*, 382:1289–1299, September 2001.

[68] P. Francois and V. Hakim. Design of genetic networks with specified functions by evolution in silico. *PNAS*, 101(2):580–585, 2004.

[69] U. Alon. Biological networks: The tinkerer as an engineer. *Science*, 301: 1866–1867, 2003.

[70] M. A. Gibson. Modeling the activity of single genes, 1999. PhD Thesis.

[71] M.B. Elowitz, A.J. Levine, E.D. Siggia, and P.S. Swain. Stochastic gene expression in a single cell. *Science*, 297:1183–1186, 2002.

[72] F. J. Isaacs, W.J. Blake, and J.J. Collins. Signal processing in single cells. *Science*, 307:1886–1888, 2005.

[73] N. Rosenfeld, J.W. Young, U. Alon, P.S. Swain, and M.B. Elowitz. Gene regulation at the single-cell level. *Science*, 307:1962–1965, 2005.

[74] E. M. Ozbudak, M. Thattai, Iren Kurtser, Alan D. Grossman, and Alexander van Oudenaarden. Regulation of noise in the expression of a single gene. *Nature Genetics*, 31:69–74, 2002.

[75] Q. Zhang, M.E. Andersen, and R.B. Conolly. Binary gene induction and protein expression in individual cells. *Theoretical Biology and Medical Modelling*, 3:1–15, 2006.

[76] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298:824–827, 2002.

[77] E. Dekel, S. Mangan, and U. Alon. Environmental selection of the feed-forward loop circuit in gene-regulation networks. *Physical Biology*, 2:81–88, 2005.

[78] S. Mangan and U. Alon. Structure and function of the feed-forward loop network motif. *PNAS*, 100(21):11980–11985, 2003.

[79] S. Keles, M. van der Laan, and C. Vulpe. Regulatory motif finding by logic regression. *Bioinformatics*, 20(16):2799–2811, 2004.

[80] P. J. Ingram, M. P.H. Stumpf, and J. Stark. Network motifs: structure does not determine function. *BMC Genomics*, 7:1–12, 2006.

[81] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, 31: 64–68, 2002.

[82] R. Boys and D. Wilkinson. Bayesian inference for stochastic kinetic genetic regulatory networks. Working paper.

[83] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81(25):2340–2361, 1977.

[84] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Chem. Physics*, 22:403–434, 1976.

[85] M. A. Gibson and J. Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Chem. Physics*, 104: 1876–1889, 2000.

[86] D.T. Gillespie. The chemical Langevin and Fokker-Planck equations for the reversible isomerization reactions. *J. Phys. Chem. A*, 106(20):5063 – 5071, 2002.

[87] D.T. Gillespie. The multivariate Langevin and Fokker-Planck equations. *American Journal of Physics*, 64:1246–1257, October 1996.

[88] D.T. Gillespie. Approximating the master equation by Fokker-Planck-type equations for single-variable chemical systems. *J. Chem. Physics*, 72: 5363–5370, May 1980.

[89] D. T. Gillespie. The chemical Langevin equation. *J. Chem. Physics*, 113(1): 297–305, 2000.

[90] D.T. Gillespie and L.R. Petzold. Improved leap-size selection for accelerated stochastic simulation. *J. Chem. Physics*, 119:8229–8234, October 2003.

[91] D. T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys*, 115:1716–1733, 2001.

[92] A. Auger, P. Chatelain, and P. Koumoutsakos. R-leaping: Accelerating the stochastic simulation algorithm by reaction leaps. *Journal of Chemical Physics*, 125:125–137, 2006.

[93] A. Chatterjee, D.G. Vlachos, and M.A. Katsoulakis. Binomial distribution based τ-leap accelerated stochastic simulation. *J. Chem. Phys*, 122:024112, January 2005.

[94] M. Rathinam, L.R. Petzold, Y. Cao, and D.T. Gillespie. Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *J. Chem. Physics*, 119:12784–12794, December 2003.

[95] Y. Cao, D.T. Gillespie, and L.R. Petzold. Avoiding negative populations in explicit Poisson tau-leaping. *J. Chem. Phys*, 123:054104, August 2005.

[96] Y. Cao, D. T. Gillespie, and L. Petzold. Efficient step size selection for the tau-leaping simulation method. *J. Chem. Phys.*, 124(4):1–11, 2006.

[97] R. Bundschuh, F. Hayot, and C. Jayaprakash. Fluctuations and slow variables in genetic networks. *Biophysical Journal*, 84:1606–1615, January 21 2003.

[98] Y. Cao, D. Gillespie, and L. Petzold. Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems. *J. Comput. Phys.*, 206:395–411, 2005.

[99] Y. Cao, D.T. Gillespie, and L.R. Petzold. The slow-scale stochastic simulation algorithm. *J. Chem. Physics*, 122:14116, January 2005.

[100] K. Burrage, T. Tian, and P. Burrage. A multi-scaled approach for simulating chemical reaction systems. *Progress in Biophysics & Molecular Biology*, 85:217–234, 2004.

[101] S. B. Rawool and K.V. Venkatesh. Steady state approach to model gene regulatory networks - simulation of microarray experiments. *Biosystems*, Advance Access:1–20, 2007.

[102] J. Puchaka and A. M. Kierzek. Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks. *Biophysical Journal*, 86:1357–1372, March 2004.

[103] K. Vasudeva and U. S. Bhalla. Adaptive stochastic-deterministic chemical kinetic simulations. *Bioinformatics*, 20(1):78–84, 2004.

[104] H. El Samad, M. Khammash, L. Petzold, and D. Gillespie. Stochastic modeling of gene regulatory networks. *Int. J. Robust Nonlinear Control*, 00:1–6, 2002.

[105] T. E. Turner, S. Schnell, and K. Burrage. Stochastic approaches for modelling in vivo reactions. *Computational Biology and Chemistry*, 28(3):165–178, 2004.

[106] H. Salis and Y. Kaznessis. Numerical simulation of stochastic gene circuits. *Computers and Chemical Engineering*, 29:577–588, 2005.

[107] J. V. Rodríguez, J. A. Kaandorp, M. Dobrzynski, and J. G. Blom. Spatial stochastic modelling of the PTS pathway in escherichia coli. *Bioinformatics*, 22(15):1895–1901, 2006.

[108] L.M. Tuttle, H. Salis, J. Tomshine, and Y.N. Kaznessis. Model-driven designs of an oscillating gene network. *Biophysical Journal*, 89:3873–3883, 2005.

[109] A. M. Kierzek, J. Zaim, and Piotr Zielenkiewicz. The effect of transcription and translation initiation frequencies on the stochastic fluctuations in prokaryotic gene expression. *Journal of Biological Chemistry*, 276(11): 8165–8172, 2001.

[110] S. Reinker, R.M. Altman, and J. Timmer. Parameter estimation in stochastic biochemical reactions. *IEE Proc.-Syst. Biol.*, 153(4):168–176, 2006.

[111] J. Tomshine and Y. N. Kaznessis. Optimization of a stochastically-simulated gene network model via simulated annealing. *Biophysical Journal*, 91:3196–3205, 2006.

[112] F. Wu, W. Zhang, and A. J. Kusalik. Modeling gene expression from microarray expression data with state-space equations. In *Pacific Symposium on Biocomputing*, pages 581–592. World Scientific, 2004. ISBN 981-238-598-3.

[113] K. Kyoda, M. Muraki, S. Hamahashi, M. Morohashi, S. Onami, and H. Kitano. Biodrive: Simulator for biochemical and genetic networks. Working paper.

[114] T. Van den Bulcke, K. Van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor, and K. Marchal. SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, 7:43, 2006.

[115] A. S. Ribeiro and J. Lloyd-Price. SGNSim, a stochastic genetic networks simulator. *Bioinformatics*, Advance Access:1–2, April 03 2007.

[116] D. Adalsteinsson, D. McMillen, and T. Elston. Biochemical network stochastic simulator BioNetS: software for stochastic modeling of biochemical networks. *BMC Bioinformatics*, 5(24):1–21, May 18 2004.

[117] A. M. Kierzek. STOCKS: STOChastic Kinetic Simulations of biochemical systems with Gillespie algorithm. *Bioinformatics*, 18(3):470–481, 2002.

[118] T. Akutsu, S. Miyano, and S. Kuhara. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In *Pacific Symposium on Biocomputing*, pages 17–28, 1999.

[119] T. Akutsu, S. Miyano, and S. Kuhara. Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, 16(8):727–734, 2000.

[120] S. Liang, S. Fuhrman, and R. Somogyi. REVEAL : A general reverse engineering algorithm for inference of genetic network architectures, 1998.

[121] D. Cho, K. Cho, and B. Zhang. Identification of biochemical networks by S-tree based genetic programming. *Bioinformatics*, 22(13):1631–1640, 2006.

[122] P. D'haeseleer, S. Liang, and R. Somogyi. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8):707–726, 2000.

[123] I. Gat-Viks and R. Shamir. Chain functions and scoring functions in genetic networks. *Bioinformatics*, 19:108–117, 2003.

[124] H. Lähdesmäki, I. Shmulevich, and O. Yli-Harja. On learning gene regulatory networks under the boolean network model. *Machine Learning*, 52 (1-2):147–167, 2003.

[125] K. G. Gadkar, R. Gunawan, and F. J. Doyle III. Iterative approach to model identification of biological networks. *BMC Bioinformatics*, 6:155, 2005.

[126] M. J. L. De Hoon, S. Imoto, K. Kobayashi, N. Ogasawara, and S. Miyano. Inferring gene regulatory networks from time-ordered gene expression data of bacillus subtilis using differential equations. In *Pacific Symposium on Biocomputing*, pages 17–28, 2003.

[127] T. MacCarthy, A. Pomiankowski, and R. Seymour. Using large-scale perturbations in gene network reconstruction. *BMC Bioinformatics*, 6:11, 2005.

[128] D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. In *ISMB (Supplement of Bioinformatics)*, pages 215–224, 2001.

[129] M. Ronen, R. Rosenberg, B.I. Shraiman, and U.Alon. Assigning numbers to the arrows: Parameterizing a gene regulation network by using accurate expression kinetics. *PNAS*, 99(16):10555–10560, 2002.

[130] S. Aburatani, K. Goto, S. Saito, H. Toh, and K. Horimoto. ASIAN: a web server for inferring a regulatory network framework from gene expression profiles. *Nucleic Acids Research*, 33:659–664, 2005.

[131] A.V. Antonova and H. W. Mewes. BIOREL: The benchmark resource to estimate the relevance of the gene networks. *FEBS Letters*, 580:844–848, 2006.

[132] X. Deng and H. H. Ali. Examine: A computational approach to reconstructing gene regulatory networks. *Biosystems*, 31:125–136, 2005.

[133] K. Bhasi, A. Forrest, and M. Ramanathan. SPLINDID: a semi-parametric, model-based method for obtaining transcription rates and gene regulation parameters from genomic and proteomic expression profiles. *Bioinformatics*, 21(20):3873–3879, 2005.

[134] M. J. Beal, F. Falciani, Z. Ghahramani, C. Rangel, and D. L. Wild. A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*, 21(3):349–356, 2005.

[135] A. Bernard and A. J. Hartemink. Informative structure priors: Joint learning of dynamic regulatory networks from multiple types of data. In *Pacific Symposium on Biocomputing*. World Scientific, 2005.

[136] S. R. Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–763, 1998.

[137] A. Fridman. Mixed Markov models. *PNAS*, 100(14):8092–8096, 2003.

[138] W. Ching, M. K. Ng, E. S. Fung, and T. Akutsu. On construction of stochastic genetic networks based on gene expression sequences. *Int. J. Neural Syst*, 15(4):297–310, 2005.

[139] T. Chu, C. Glymour, R. Scheines, and P. Spirtes. A statistical problem for inference to regulatory structure from associations of gene expression measurements with microarrays, 2002.

[140] A. Datta, A/ Choudhary, M. L. Bittner, and E. R. Dougherty. External control in markovian genetic regulatory networks. *Machine Learning*, 52 (1-2):169–191, 2003.

[141] N. Nariai, S. Kim, S. Imoto, and S. Miyano. Using protein-protein interactions for refining gene networks estimated from microarray data by Bayesian networks. In *Pacific Symposium on Biocomputing*, pages 336–347. World Scientific, 2004. ISBN 981-238-598-3.

[142] P. Spirtes, G. Glymour, S. Kauffman, V. Aimalie, and F. Wimberly. Constructing bayesian network models of gene expression networks from microarray data. In *Proc. Atlantic Symp. Comp. Biol., Genome Information Systems & Technology.*, 2000.

[143] D. Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, 19(17):2271–2282, 2003.

[144] H. Lähdesmäki, S. Hautaniemi, I. Shmulevich, and O. Yli-Harja. Relationships between probabilistic boolean networks and dynamic bayesian networks as models of gene regulatory networks. *Signal Processing*, 86(4): 814–834, 2006.

[145] S. Imoto, T. Goto, and S. Miyano. Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression. In *Pacific Symposium on Biocomputing*, pages 175–186, 2002.

[146] S. Imoto, S. Kim, T. Goto, S. Aburatani, K. Tashiro, S. Kuhara, and S. Miyano. Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. *J. Bioinformatics and Computational Biology*, 1(2):231–252, 2003.

[147] K. Missal, M. A. Cross, and D. Drasdo. Gene network inference from incomplete expression data: transcriptional control of hematopoietic commitment. *Bioinformatics*, 22(6):731–738, 2006.

[148] F.E. Streib, M. Dehmer, G. H. Bakir, and Max Muhlhauser. Influence of noise on the inference of dynamic bayesian networks from short time series. *Transactions on Engineering, Computing and Technology*, 10:70–74, 2005.

[149] Y. Tamada, S. Kim, H. Bannai, S. Imoto, K. Tashiro, S. Kuhara, and S. Miyano. Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. In *ECCB*, pages 227–236, 2003.

[150] C. Yoo, V. Thorsson, and G. F. Cooper. Discovery of causal relationships in a gene-regulation pathway from a mixture of experimental and observational DNA microarray data. In *Pacific Symposium on Biocomputing*, pages 498–509, 2002.

[151] I. M. Ong, J. D. Glasner, and D. Page. Modelling regulatory pathways in E. coli from time series expression profiles. *Bioinformatics*, 18:241–248, 2002.

[152] B.-E. Perrin, Liva Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and Florence d'Alché Buc. Gene networks inference using dynamical bayesian networks. *Bioinformatics*, 19:138–148, September 2003.

[153] P. J. Woolf, W. Prudhomme, L. Daheron, G. Q. Daley, and D. A. Lauffenburger. Bayesian analysis of signaling networks governing embryonic stem cell fate decisions. *Bioinformatics*, 21(6):741–753, 2005.

[154] A. V. Werhli, M. Grzegorczyk, and D. Husmeier. Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and Bayesian networks. *Bioinformatics*, 22(20):2523–2531, 2006.

[155] F. Li and Y. Yang. Recovering genetic regulatory networks from microarray data and location analysis data. *Genome Informatics*, 15(2):131–140, 2004.

[156] A. A. Motsinger, S. L. Lee, G. Mellick, and M. D. Ritchie. Gpnn: Power studies and applications of a neural network method for detecting gene-gene interactions in studies of human disease. *BMC Bioinformatics*, 7:39, 2006.

[157] B. A. Sokhansanj, J. P. Fitch, J. N. Quong, and A. A. Quong. Linear fuzzy gene network models obtained from microarray data by exhaustive search. *BMC Bioinformatics*, 5:108, 2004.

[158] S. Nacu, R. Critchley-Thorne, P. Lee, and S. Holmes. Gene expression network analysis and applications to immunology. *Bioinformatics*, 23(7): 850–858, 2007.

[159] I. Nemenman. Information theory, multivariate dependence, and genetic network inference. CoRR, 2004.

[160] O. Lipan and W. H. Wong. The use of oscillatory signals in the study of genetic networks. *PNAS*, 102(20):7063–7068, 2005.

[161] J. Tegner, M. K. Yeung, J. Hasty, and J.J. Collins. Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *PNAS*, 100(10):5944–5949, 2003.

[162] M. P. Brynildsen, L. M. Tran, and J. C. Liao. A Gibbs sampler for the identification of gene expression and network connectivity consistency. *Bioinformatics*, 22(24):3040–3046, 2006.

[163] L.A. Shehadeh, L.S. Liebovitch, and V.K. Jirsa. Relationships between the global structure of genetic networks and mRNA levels measured by cDNA microarrays. *Physica A*, 364:297–314, 2006.

[164] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS*, 96:6745–6750, 1999.

[165] T. Schlitt, K. Palin, J. Rung, S. Dietmann, M. Lappe, E. Ukkonen, and A. Brazma. From gene networks to gene function. *Genome Research*, 13: 2568–2576, 2003.

[166] J. Huang, H. Shimuzi, and S. Shioya. Clustering gene expression pattern and extracting relationship in gene network based on artificial neural networks. *Journal of Bioscience and Bioengineering*, 96(5):421–428, 2003.

[167] H. Liu, S. Tarima, A. S. Borders, T. V. Getchell, M. L. Getchell, and A. J. Stromberg. Quadratic regression analysis for gene discovery and pattern recognition for non-cyclic short time-course microarray experiments. *BMC Bioinformatics*, 6:106–122, April 25 2005. ISSN 1471-2105.

[168] P. Meltzer, J. M. Trent, and M. Bittner. Multivariate measurement of gene expression relationships. *Genomics*, 67:201–209, April 21 2000.

[169] A. W. Schreiber and U. Baumann. A framework for gene expression analysis. *Bioinformatics*, 23(2):191–197, 2007.

[170] P. Subramani, R. Sahu, and S. Verma. Feature selection using haar wavelet power spectrum. *BMC Bioinformatics*, 7:432, 2006.

[171] H. Toh and K. Horimoto. Inference of a genetic network by a combined approach of cluster analysis and graphical gaussian modeling. *Bioinformatics*, 18(2):287–297, 2002.

[172] K. Yeung and P. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.

[173] Xi. Zhou, X. Wang, and E. R. Dougherty. Construction of genomic networks using mutual-information clustering and reversible-jump markov-chain monte-carlo predictor design. *Signal Processing*, 83(4):745–761, 2003.

[174] V. Filkov, S. Skiena, and J. Zhi. Analysis techniques for microarray time-series data. *Journanl of Computational Analysis*, 9(2):317–330, 2002.

[175] T. Lu, C. M. Costello, P. J. P. Croucher, R. Häsler, G. Deuschl, and S. Schreiber. Can Zipf's law be adapted to normalize microarrays? *BMC Bioinformatics*, 6:37, 2005.

[176] X. Qiu, A. I. Brooks, L. Klebanov, and A. Yakovlev. The effects of normalization on the correlation structure of microarray data. *BMC Bioinformatics*, 6:120–130, June 22 2005.

[177] Y. Chen, V. Kamat, E. R. Dougherty, M. L. Bittner, P. S. Meltzer, and J. M. Trent. Ratio statistics of gene expression levels and applications to microarray data analysis. *Bioinformatics*, 18(9):1207–1215, 2002.

[178] J.D. Storey, W. Xiao, J.T. leek, R.G. Tompkins, and R.W. Davis. Significance analysis of time course microarray experiments. *PNAS*, 102(36):12837–12842, 2005.

[179] H. Li, C. L. Wood, Y. Liu, T. V. Getchell, M. L. Getchell, and A. J. Stromberg. Identification of gene expression patterns using planned linear contrasts. *BMC Bioinformatics*, 7:245, 2006.

[180] Y. Liang, B. Tayo, X. Cai, and A. Kelemen. Differential and trajectory methods for time course gene expression data. *Bioinformatics*, 21(13): 3009–3016, 2005.

[181] S. Ma. Empirical study of supervised gene screening. *Bioinformatics*, 7: 537–562, 2006.

[182] R. Pan. A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments. *BIOINF: Bioinformatics*, 18(4):546–554, 2002.

[183] T. Park, S. Yi, S. Lee, S. Y. Lee, D. Yoo, J. Ahn, and Y. Lee. Statistical tests for identifying differentially expressed genes in time-course microarray experiments. *Bioinformatics*, 19(6):694–703, 2003.

[184] S. Wichert, K. Fokianos, and K. Strimmer. Identifying periodically expressed transcripts in microarray time series data. *Bioinformatics*, 20(1): 5–20, 2004.

[185] D. J. Bakewell and E. Wit. Weighted analysis of microarray gene expression using maximum-likelihood. *Bioinformatics*, 21(6):723–729, 2005.

[186] A. de la Fuente, N. Bing, I. Hoeschele, and P. Mendes. Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20(18):3565–3574, 2004.

[187] J. Schäfer and K. Strimmer. An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764, 2005.

[188] J. Schafer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1):1–32, 2005.

[189] J. J. Rice, Y. Tu, and G. Stolovitzky. Reconstructing biological networks using conditional correlation analysis. *Bioinformatics*, 21(6):765–773, 2005.

[190] J. Wang, L. W. Cheung, and J. Delabie. New probabilistic graphical models for genetic regulatory networks studies. *Journal of Biomedical Informatics*, 38(6):443–455, 2005.

[191] F. Gao, B. C. Foat, and H. J. Bussemaker. Defining transcriptional networks through integrative modeling of mRNA expression and transcription factor binding data. *BMC Bioinformatics*, 5:31, 2004.

[192] D. Ghosh. Resampling methods for variance estimation of singular value decomposition analyses from microarray experiments. *Funct. Integr. Genomics*, 2:92–97, 2002.

[193] W. S. Bush, S. M. Dudek, and M. D. Ritchie. Parallel multifactor dimensionality reduction: a tool for the large-scale analysis of gene-gene interactions. *Bioinformatics*, 22(17):2173–2174, 2006.

[194] H. Salis, V. Sotiropoulos, and Y. N. Kaznessis. Multiscale Hy3S: Hybrid stochastic simulation for supercomputers. *BMC Bioinformatics*, 7:93, 2006.

[195] M. Schwehm. Parallel stochastic simulation of whole-cell models. *Mitteilungen — Gesellschaft für Informatik e.V., Parallel-Algorithmen und Rechnerstrukturen*, 19:60–68, 2002.

[196] C. J. Penkett and J. Bahler. Navigating public microarray databases. *Comp Funct Genom 2004*, 5:471–479, 2004.

[197] H. Parkinson, M. Kapushesky, M. Shojatalab, N. Abeygunawardena, R. Coulson, A. Farne, E. Holloway, N. Kolesnykov, P. Lilja, M. Lukk, R. Mani, T. Rayner, A. Sharma, E. William, U. Sarkans, and A. Brazma. ArrayExpress - a public database of microarray experiments and gene expression profiles. *Nucleic Acids Research*, 35(Database-Issue):747–750, 2007.

[198] M. Bansal, G. Della Gatta, and D. di Bernardo. Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics*, 22(7):815–822, 2006.

[199] Z. Bar-Joseph. Analyzing time series gene expression data. *Bioinformatics*, 20(16):2493–2503, 2004.

[200] S. D. Bay, L. Chrisman, A. Pohorille, and Jeff J. Shrager. Temporal aggregation bias and inference of causal regulatory networks. *Journal of Computational Biology*, 11(5):971–985, 2004.

[201] D. R. Bickel. Probabilities of spurious connections in gene networks: application to expression time series. *Bioinformatics*, 21(7):1121–1128, 2005.

[202] E. J. Hannan. *Multiple time series*. Wiley, New York, 1970.

[203] H. Lütkepohl. *Introduction to Multiple Time Series Analysis*. Springer-Verlag, Berlin, Germany, 2nd edition, 1993. ISBN 3-540-56940-5.

[204] M. B. Priestley. *Spectral analysis and time series, vol II*. Academic Press, London, 1981.

[205] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974.

[206] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6: 461–464, 1978.

[207] C. Kadilar and C. Erdemir. Comparison of performance among information criteria in var and seasonal var models. *Hacettepe Journal of Mathematics and Statistics*, 31:127–137, 2002.

[208] E. Bagarinao and S. Sato. Algorithm for vector autoregressive model parameter estimation using an orthogonalization procedure. *Annals of Biomedical Engineering*, 30:260–271, 2002.

[209] J. A. Mauricio. Exact maximum likelihood estimation of stationary vector ARMA models. *Journal of the American Statistical Association*, 90(429):282–291, March 1995.

[210] T. Schneider and A. Neumaier. Algorithm 808: ARfit - a Matlab package for the estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Trans. Math. Softw*, 27(1):58–65, 2001.

[211] A. Fujita, J.R. Sato, H.M. Garay-Malpardita, P.A. Morrettin, M.C. Sogayar, and C.E. Ferreira. Time-varying modelling of gene expression regulatory networks using the wavelet dynamic vector autoregressive model. *Bioinformatics*, Preaccess:1–8, 2007.

[212] F. He and A. Zeng. In search of functional association from time-series microarray data based on the change trend and level of gene expression. *BMC Bioinformatics*, 7:69, 2006.

[213] R.F. Engle and C.W.J. Granger. Co-integration and error correction: Representation, estimation, and testing. *Econometrica*, 55(2):251–276, 1987.

[214] C. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

[215] B. Efron. Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7(1):1–26, January 1979.

[216] B. Efron and Robert Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, London, 1993.

[217] D.N. Politis. The impact of bootstrap methods on time series analysis. *Statistical Science*, 18(2):219–230, 2003.

[218] T. Krul, J. A. Kaandorp, and J. G. Blom. Modelling developmental regulatory networks. In *Computational Science - ICCS 2003, Melbourne, Australia and St. Petersburg, Russia, Proceedings Part I*, volume 2660 of *Lecture Notes in Computer Science*, pages 688–697. Springer Verlag, June 2003.

[219] D. Wilkinson. *Stochastic Modelling for Systems Biology*. Chapman And Hall, 2006.

[220] J. Pearl. *Causality : Models, Reasoning, and Inference*. Cambridge University Press, March 2000. ISBN 0521773628.

[221] N.Soranzo, G. Bianconi, and C. Altafini. Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks: synthetic versus real data. *Bioinformatics*, 23(13):1640–1647, 2007.

[222] H. Hotelling. New light on the correlation coefficient and its transforms. *J. R. Statistic. Soc. B.*, 15:193–232, 1953.

[223] M.G. Kendall. The estimation of parameters in linear autoregressive time series. *Econometrica*, 17:44–57, 1949.

[224] P. Whittle. On the fitting of multivariate autoregressions and the approximate canonical factorization of a spectral density matrix. *Biometrika*, 40: 120–134, 1963.

[225] J. Durbin. The Fitting of Time-Series Models. *Rev. Int. Statist. Inst.*, 28(3), 1960.

[226] P. Bühlmann and H. R. Künsch. Block length selection in the bootstrap for time series. *Comput. Statist. Data Anal*, 31:295–310, 1999.

[227] P. Buhlmann. Bootstraps for time series. *Statistical Science*, 17(1):52–72, 2002.

[228] J. Hidalgo. A bootstrap causality test for covariance stationary processes. Sticerd - econometrics paper series, Suntory and Toyota International Centres for Economics and Related Disciplines, LSE, 2003. URL `http://econpapers.repec.org/RePEc:cep:stiecm:/2003/462`.

[229] A. Hatemi and G. Shukur. Multivariate based causality tests of twin deficits in the us. Technical report, International Business School, J önköping University, 2007.

[230] G. Kerr, H. J. Ruskin, M. Crane, and P. Doolan. Techniques for clustering gene expression data. *Computers in Biology and Medicine*, 38:283–293, 2008.

[231] J. B. Macqueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[232] L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research*, 9:1106 – 1115, 1999.

[233] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. URL `http://www.stanford.edu/~darthur/kMeansPlusPlus.pdf`.

[234] S. Ray and R. Turi. Determination of number of clusters in k-means clustering and application in colour image segmentation. *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques*, pages 137 – 143, 1999.

[235] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, London, 1980.

[236] I. T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[237] M. E. Pownall, A. S. Tucker, J. Slack, and H. Isaacs. eFGF, Xcad3 and Hox genes form a molecular pathway that establishes the anteroposterior axis in xenopus. *Development*, 122:3881 – 3892, 1996.

[238] W. Hesse, E. Moller, M. Arnold, and B. Schack. The use of time-variant eeg granger causality for inspecting directed interdependencies of neural assemblies. *Journal of Neuroscience Methods*, 124(1):27 –44, 2003.

[239] D. Marinazzo, M. Pellicoro, and S. Stramaglia. Kernel Granger causality and the analysis of dynamical networks, March 2008.

[240] Y. Chen, G. Rangarajan, J. Feng, and M. Ding. Analyzing multiple nonlinear time series with extended Granger causality. *Physics Letters A*, 324: 26–35, 2004.