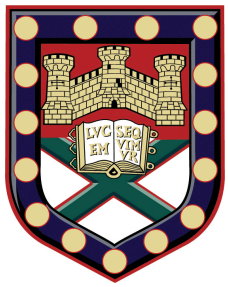
Consensus Network Inference of Microarray Gene Expression Data



Suhaib Mohammed

University of Exeter

Doctor of Philosophy in Biological Sciences

In July 2016

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

(signed)…….....

"Everything is theoretically impossible, until it is done"

- Robert A. Heinlein

Abstract

Genetic and protein interactions are essential to regulate cellular machinery. Their identification has become an important aim of systems biology research. In recent years, a variety of computational network inference algorithms have been employed to reconstruct gene regulatory networks from post-genomic data. However, precisely predicting these regulatory networks remains a challenge.

We began our study by assessing the ability of various network inference algorithms to accurately predict gene regulatory interactions using benchmark simulated datasets. It was observed from our analysis that different algorithms have strengths and weaknesses when identifying regulatory networks, with a gene-pair interaction (edge) predicted by one algorithm not always necessarily consistent with the other. An edge not predicted by most inference algorithms may be an important one, and should not be missed. The naïve consensus (intersection) method is perhaps the most conservative approach and can be used to address this concern by extracting the edges consistently predicted across all inference algorithms; however, it lacks credibility as it does not provide a quantifiable measure for edge weights. Existing quantitative consensus approaches, such as the inverse-variance weighted method (IVWM) and the Borda count election method (BCEM), have been previously implemented to derive consensus networks from diverse datasets. However, the former method was biased towards finding local solutions in the whole network, and the latter considered species diversity to build the consensus network.

In this thesis we proposed a novel consensus approach, in which we used Fishers Combined Probability Test (FCPT) to combine the statistical significance values assigned to each network edge by a number of different networking algorithms to produce a consensus network. We tested our method by applying it to a variety of *in silico* benchmark expression

datasets of different dimensions and evaluated its performance against individual inference methods, Bayesian models and also existing qualitative and quantitative consensus techniques. We also applied our approach to real experimental data from the yeast (*S. cerevisiae*) network as this network has been comprehensively elucidated previously. Our results demonstrated that the FCPT-based consensus method outperforms single algorithms in terms of robustness and accuracy. In developing the consensus approach, we also proposed a scoring technique that quantifies biologically meaningful hierarchical modular networks.

Acknowledgements

This research project would not have been completed without the following people. First and foremost, I would like to thank my primary supervisor, Dr. Zheng Rong Yang, for his guidance, valuable support and critical feedback throughout this project. I also extend my deepest gratitude to my secondary supervisor, Dr. Ozgur E. Akman, for his vital suggestions, and comments in and out of lab meetings. Without their support, it would have been difficult to accomplish this research. Furthermore, I would like to thank my external examiner, Dr. Guido Sanguinetti, and internal examiner, Dr. Ed Keedwell for their feedback and suggestions, which improved the quality of this thesis.

I am very grateful to my friends and lab colleagues, Varun Kothamachu, Abdul Latif, Ben Wareham, Jason Bullock for providing suggestions and support during our period of time spent in Exeter. I was fortunate to come across Erratics Cricket Club, who helped enable me to meet many wonderful people, motivating throughout the duration of the project.

I would personally like to thank my wife, Umme Salma, for her constant moral support and encouragement, especially during the most arduous periods. Despite being pregnant during my final research and thesis write up stage, she motivated me and provided me space to concentrate. I also owe a big thank you to my mother for her unconditional support and constant encouragement by telephone despite her residence in the Asian subcontinent.

Last but not least, I thank my newborn daughter, Safaa. She has given me reason to smile during the most stressful times of my thesis write-up. Finally, I extend my gratitude and thankfulness to the University of Exeter Studentship for supporting me financially during the course of my research.

Table of Contents

List of Figures

List of Tables

Authors Declaration

This work includes material from published papers from conference proceedings, which contribute to the content of the thesis. These published papers, are contributions to the thesis content during the period of the author's study at the University of Exeter. All other material which is not my own, has been identified and referenced accordingly. I hereby declare that I am the sole author of this thesis.

Published papers from the conference proceedings

- Mohammed, S., Akman, O. E., & Yang, Z. R. (2014). A consensus approach to predict regulatory interactions. In *2014 7th International Conference on Biomedical Engineering and Informatics* (pp. 769–775). Dalian, China: IEEE. doi:10.1109/BMEI.2014.7002876 (Included in Chapter 3)

- Mohammed, S. (2013). Comparative analysis of network algorithms to address modularity with gene expression temporal data. In *ACM Conference on Bioinformatics, Computational Biology and Biomedical Informatics Proceedings* (Vol. 978–1–4503, pp. 876–882). Washington, USA: ACM Digital Library. doi:10.1145/2506583.2506698 - (Included in Chapter 4 and Chapter 5)

Chapter 1

Introduction

Abstract

This chapter provides a general introduction to the subject of modelling biological networks; the discussion of different biological networks applied in systems biology research, the purpose of biological network inference and the implication of transcriptional networks. Elementary concepts and definitions of network properties, and biological data used for network reconstruction are also discussed. Finally, the motivation for this study, including its novelty aspects are discussed.

1.1 General Introduction

A cell is a functional unit of life that serves as a building block for all living organisms. The large quantity of coded information stored in the DNA of a cell's genes coordinates complex biological processes. Complexity is the hallmark of cellular systems. The regulatory interaction between the genes and its products (proteins) orchestrates this complexity. As a result the biological information is transferred via several pathways (e.g. signalling, regulatory and metabolic), which form complex regulatory networks between biological entities such as DNA, RNA, proteins and metabolites. Therefore, a key challenge is to understand the structure and relationship between genes that coordinate multiple functions of a living cell within a dynamically changing environment (Bennett et al. 2008).

The post-genomic era is invariably shifting from annotating individual genes and proteins to understanding complex interactions between biological entities inside the cell, to investigating regulatory signalling and metabolic pathways when exposed to external perturbations (Cassman et al. 2007). In order to understand this complexity, contemporary scientists depend on the "reductionist" approach - employing mathematical modeling tools and techniques to understand biological complexity at molecular levels. This approach has been pursued in the past two decades to characterize and identify regulatory interactions, starting from a gene or protein of interest, and trying to uncover its involvement in the same or different pathways. By integrating knowledge of biological data using hypothesis-driven research and employing mathematical modeling tools, we can explore the functions of genes and proteins, and gain insights into the mechanisms underlying biological activity.

Systems biology aims to understand the physiology of living systems on a whole, rather than in parts (Ma'ayan 2011). Networks or graphs provide mathematical abstraction when representing a broad variety of complex systems, such as the internet, social interactions, and

biological and ecological systems (Albert R 2002; Barabási & Oltvai 2004). To some extent, biology researchers embrace the network description as it compactly depicts the control system of the cell, which represents the expression of all genes in tight coordination (Haiyuan Yu, Nicholas M Luscombe 2003) as shown in Figure 1.1.

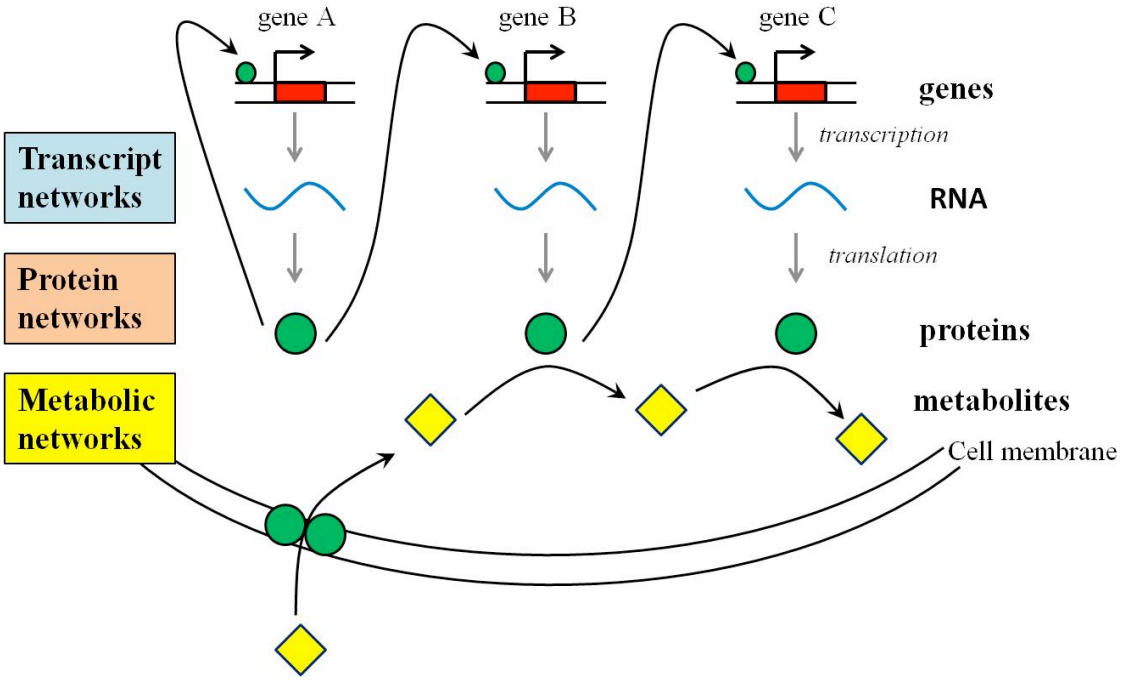


Figure 1.1: The above schematic depicts the different levels at which biological networks can be described. Gene A shows the autocrine effect, in which it regulates its own gene transcription; the product of gene A also influences the transcription of gene B, with gene B having an effect on the transcription of gene C.

Biological networks are composed of nodes and edges. The former represent biological entities, whilst the latter illustrates the regulatory relationship between the entities. High throughput technology data - like transcriptomics, proteomics and metabolomics - have enabled researchers to consider genome-wide approaches to understand and analyze biological entities on a global level. Cellular components do not work alone, but instead, interact with each other within a highly complex structure. The schematics in Figure 1.1 depict the central dogma of molecular biology, whereby genetic material (DNA) is

transcribed into RNA molecules, and then translated into proteins. The proteins are the end products and carry out a vast array of functions, including acting as transcription factors (TFs) that promote (or repress) transcription, catalyzing metabolic reactions and transporting molecules at different locations (Desvergne et al. 2006).

In order to uncover the complex behavior of the biological system, it is imperative to define biological entities and their interactions within a model (Hecker et al. 2009). Therefore, representing the complex interactions between genes and proteins using networks enables us to visualize and unfold the mechanism of the underlying biological process. Biological networks can be reconstructed using various network inference algorithms (Hecker et al. 2009; Markowetz & Spang 2007). Once an algorithm is chosen, optimized parameters are required to fit the data used in the reconstruction process.

Contemporary experimental technologies provide heterogeneous high-throughput data that enables us to measure biological networks and their components at various levels. These include mRNA transcripts measurements, protein abundance and metabolite quantification. The summary of such networks at multiple levels is described below.

- Transcriptional networks describe the transcriptional regulation of genes through proteins called transcription factors (TFs). Nodes indicate genes (or proteins), and edges denote physical or regulatory interactions. A directed edge between a source and target gene, represents a transcriptional activator (positive regulation) or inhibitor (negative regulation) that controls the production of an RNA or protein molecule. Such networks - also referred to as gene regulatory networks (GRNs) - encapsulate direct and indirect regulatory relationships between genes. For example, in Figure 1.1, gene A shows the autocrine effect as it regulates its own gene transcription through synthesised protein. Gene A also influences the transcription of gene B, which has an effect on the transcription of gene C. To study the physical interaction between a TF

and the promoter of a target gene, the Chromatin immunoprecipitation (ChIP) experiment is commonly performed to determine whether a particular protein (TF) binds to the specified DNA sequence (Promoter) (Carey et al. 2009).

- Protein networks describe the physical interactions between their components, like binding and complex formation. In such graphs, also referred to as protein-protein interaction networks, a node indicates a protein, whilst an edge represents the interaction between two nodes (molecules). A yeast-two-hybrid screen is used to experimentally verify the physical interactions between pairs of protein molecules (Miller & Stagljar 2004).
- Metabolic networks describe a set of metabolites and the corresponding set of chemical reactions, which are associated with the metabolites. In such networks, metabolites are assigned as nodes and the edges represent the biochemical reaction catalysed by an enzyme (protein) between a substrate-product pair (Hatzimanikatis et al. 2004). Mass-Spectrometry (MS) techniques are widely employed to identify potential metabolites (Weckwerth 2003).

1.2 Modularity in biological networks

Uncovering the topology and dynamics of biological networks can provide useful insights into how a cell responds to a specific external perturbation, when executing complex biological processes. Outlined below are some of the notable characteristics of biological networks that enable us to understand their function.

1.2.1 Modularity

One prominent characteristic of biological networks are their embedded modular structure (Barabási & Oltvai 2004). A modular system is composed of subsystems which each perform specific functions autonomously. A biological network - which is sparsely interconnected - exhibits such modularity, which in turn facilitates specific biological functions. Specifically, densely populated sets of nodes (genes or proteins) - that are linked functionally or physically and which regulate a signaling or metabolic pathway - are called hubs (Blais & Dynlacht 2005). These densely colonized hubs display a modular structure, sharing common biological functions and showing similar expression patterns in response to external perturbations to the subnetwork. For example, those groups of genes that are co-regulated with respect to time govern the different stages of the cell cycle (Simon et al. 2001).

Hierarchical modular architecture is an extension of modularity that delineates how biologically related functional modules are organised within the network. Many biological networks - ranging over metabolic, protein and genetic interactions - show signatures of hierarchical topology, in which functional modules do not independently coexist, but combine in a hierarchical fashion for governing entities of biological process (Ravasz & Barabási 2003; Yu & Gerstein 2006).

1.2.2 Network Robustness

Biological systems are robust, responding to various external and internal perturbations, whilst still being able to perform their biological functions (Barabási & Oltvai 2004). In a topological sense, environmental perturbations and other effects cause mutations of genes under which the networks continuously evolve. To cope with the effect of these perturbations, biological networks adapt their robustness in order to attain phenotypic stability. The mechanisms through which the networks are rewired to resist these changes and

restore stability are redundancy (i.e. duplication of the genome), positive and negative feedback control, and degeneracy (i.e. different biological entities of the network performing the same function in order to yield the same effect or output) (Barabási & Oltvai 2004; Blais & Dynlacht 2005). It has been argued that modules facilitate this adaptation of robustness, as they are able to maintain a cellular function despite the malfunctioning of genes under specific external perturbations. For example, the mutation of many single genes by deletion in *Saccharomyces cerevisiae* has had an insignificant effect on the organism's growth rate (Breslow et al. 2008).

1.3 Network graphs

A network is represented by a graph in mathematical terminology. A graph G with no multiple edges and loops is a pair of sets ($V(G)$, $E(G)$) where $V(G)$ represents a set of nodes or vertices, and $E(G)$ represents a set of edges, each of which links two nodes.

1.3.1 Directed and undirected graphs

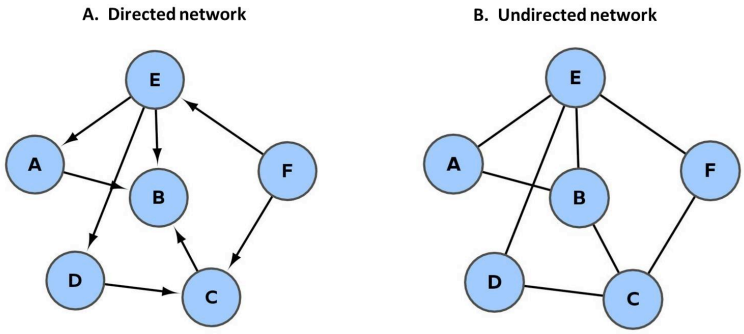


Figure 1.2. Sample directed (A) and undirected (B) networks with 6 nodes (A, B, C, D, E and F) representing genes (or proteins). Edges represent the directional interaction between two genes/proteins and their functional relationship.

A directed graph is one in which edges have specific directions or arrows, whereas in an undirected graph, edges have no directions. A directed edge indicates a causal relationship between two nodes if an edge exists. A sample directed and undirected graph is shown in Figure 1.2(A-B), where each node corresponds to a gene and edges corresponds to the relationship between two genes.

1.3.2 Weighted graphs

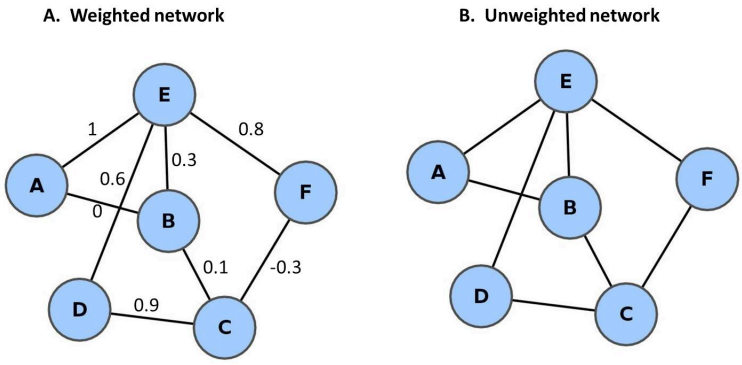


Figure 1.3. A sample weighted (A) and unweighted network (B). The network attributes are the same as in Figure 1.2.

A weighted graph is one where each edge has an associated weight, reflecting the strength of the connection between the two nodes. The weights can be either positive or negative numbers, indicating whether the edge represents activation or inhibition respectively. By contrast, an unweighted graph has no weights associated with its edges. A simple weighted and unweighted graph is shown in Figure 1.3(A-B).

1.4 Microarray datasets

The generation of high throughput data has become increasingly prevalent over the last decade. Microarray technology, in particular, has enabled expression levels to be measured

for large number of genes. The underlying principle across all microarray experiments is the same. A microarray consists of a silicon chip or glass slide that carries a large number of immobilized short single-stranded DNA sequences (ssDNAs) - more commonly known as probes. Hybridization experiments are carried out with labeled mRNAs, which attach to the probes with a reverse complementary sequence. Gene expression levels are quantified by a counting the number of labeled mRNAs bound to each probe using a scanning device. The most common microarray platforms are the single channel experiment and the two channel experiment (Ness 2006), both shown in Figure 1.4.

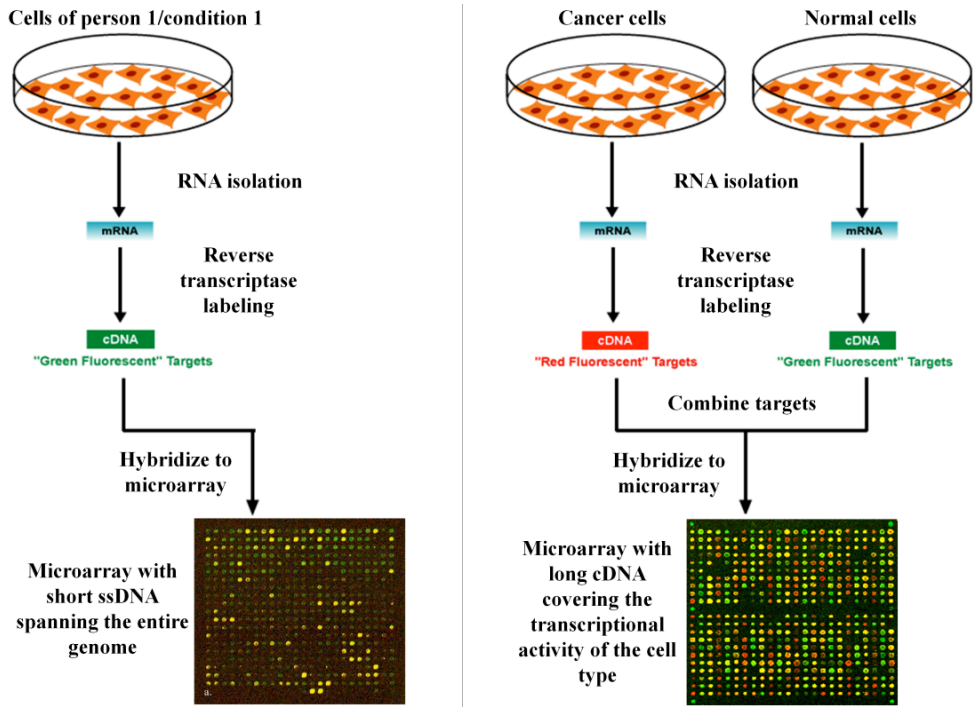


Figure 1.4. Schematic illustrating the flow process of a single channel microarray (left panel) and a two-channel microarray (right panel). This figure was adapted from (Serra 2011).

1.4.1 Single channel microarray experiments

A single channel experiment is also known as an *oligonucleotide microarray*. This means that in one experiment, only one target sample is analysed. In this platform, genes are represented

by a set of short ssDNA carrying probes - oligonucleotides (i.e. 25 mer probes). Target mRNAs are labelled fluorescently and probe-target hybridization is quantified by the detection of fluorescence signals using a scanning device (Figure 1.4-left panel). These arrays provide raw measures of expression for each individual gene (i.e. absolute expression levels). Popular single channel arrays are Affymetrix Gene Chips. A key advantage of oligonucleotide microarrays is their high specificity. For example, during the design process of the oligonucleotide sequence for a particular gene, each gene of the target gene sequence perfectly complements another; concomitantly, its partner sequence is deliberately designed to have a single base mismatch in its centre. This minimises the effects of non-specific binding.

1.4.2 Two channel microarray experiments

Two channel microarrays are also known as cDNA (complementary DNA) microarrays. These use single-stranded cDNA sequences as probes. This platform allows the sampling of mRNA from two different conditions within the same experiment, labelled with two distinct types of dyes – Cy3 (green) and Cy5 (red). Essentially, one of the labelled dyes is used as a control and the other as the experimental condition of interest (for example – disease, time point, etc.), as shown in the right panel of Figure 1.4. Target mRNAs are labelled with fluorescent dyes and expression levels are quantified by two scanning devices that detect Cy3 and Cy5 signals respectively. These arrays measure the relative difference in gene expression levels.

1.4.3 Steady-state microarray experiments

Steady-state microarray experiments sample the expression of all mRNAs at a single time point following the perturbation of a target gene (Wang et al. 2013). Here, perturbation refers

to the genetic manipulation of the genome (knock-out, knockdown or over-expression). Steady-state data does not capture the dynamics of the biological system, but it provides information as to how the expression levels of all the genes are influenced by that of a particular gene.

1.4.4 Time series microarray experiments

Time series microarray data is used to explore the dynamics of biological systems when exposed to environmental perturbations (e.g. chemical stress, heat shock, and drug treatments) (Wang et al. 2013). Here, all mRNAs are sampled at consecutive time points, from the time the external signal is introduced into the system. Time series data captures the dynamics of the experiment, and it allows delineating directional interactions between genes to understand the cause and effect relationship. The profile obtained by plotting gene expression against sampling time then quantifies the expression dynamics (Androulakis et al. 2007).

1.5 Reconstruction of gene networks

The reconstruction of GRNs based on gene expression data is known as network inference or reverse engineering. GRN reconstruction primarily uses RNA expression levels measured by microarray experiments across different experimental conditions (Figure 1.5). Typically two types of data are used: steady state and time series (see 1.4.3 and 1.4.4 above). GRN reconstruction has two main aims: locally, to determine how one gene's activity affects another gene’s activity; and globally, to determine how genes collectively respond to a perturbation. The inferred interactions can, for example, be TF-gene interactions or gene-gene interactions (Hecker et al. 2009).

In the past few years, several network inference algorithms have been developed. However, identifying GRNs in an accurate and robust manner still remains a challenge (Penfold & Wild 2011). These algorithms are broadly graded into two classes: 1) algorithms that attempt to uncover "physical interactions" - these aim to identify protein-gene interactions (i.e. TF binding on the *cis*-regulatory region of a target promoter genomic DNA sequence); and 2) algorithms that attempt to uncover "influence interactions" by identifying the regulatory relationship between genes based on expression dynamics (i.e. gene-gene interactions). Here, both classes are referred to collectively as "regulatory interactions".

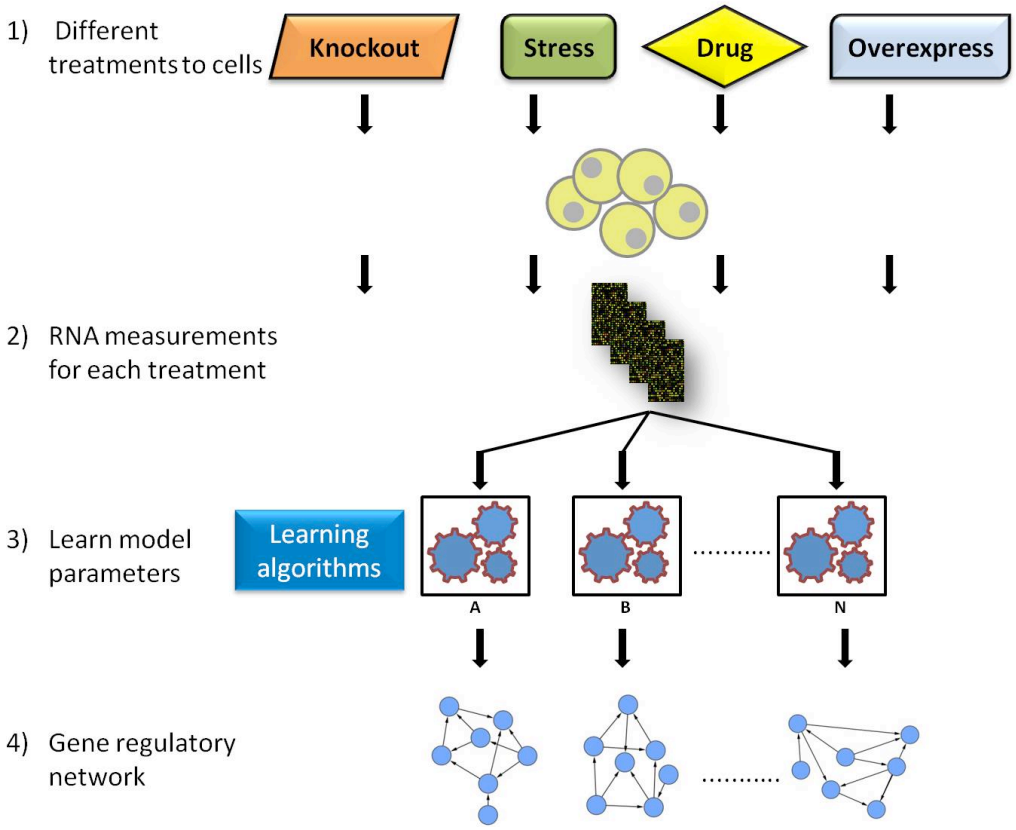


Figure 1.5. The flow process for gene regulatory network using reverse engineering approaches. A and B represents different network inference algorithms and N indicates the number of algorithms used.

1.6 Motivation behind the study

Recent evidences have suggested that when different inference methods are applied to the same microarray dataset, inconsistent predictions occur (De Smet & Marchal, 2010; Maetschke, Madhamshettiwar, Davis, & Ragan, 2014; Marbach et al., 2010) which is not surprising - as depicted in Figure 1.5, a gene pair interaction (edge) predicted from one network inference algorithm may not always be necessarily predicted by the other. One way of dealing with this discrepancy in predictions is to combine the results obtained from different inference algorithms, thereby forming an ensemble that delivers more robust predictions. Furthermore, this is an intuitive step for better coverage of gene interactions, consequently, it increases sensitivity. There are two ways to build up such an ensemble: conservative (qualitative) or profitable (quantitative). The conservative method provides a simple way of combining predictions to deliver a consensus output, based on the consistency of patterns or topology, without much importance given to numerical values. That is, the common predicted edge interactions by network algorithms used. Despite their simplicity, a major drawback of conservative methods is that they fail to provide quantifiable measures and so important interactions can be missed. The profitable method combines the predictions obtained from independent studies using statistical techniques (Borenstein & Rothstein 2007). This meta-analysis approach has been successfully applied in diverse areas - from genomic research for detecting differentially expressed genes by combining multiple gene expression profiles (Chang et al. 2013) to medical research for integrating the results of independent clinical trial studies (DerSimonian & Laird 1986). Therefore, the success of meta-analysis approaches in other disciplines motivated us to investigate its potential to produce more robust, and accurate networks that solves network inference problem.

In recent years, there have been several studies exploring meta-analysis techniques for combining results in the field of network inference (Tseng et al. 2012; Steele & Tucker

2008). However, most studies focused on combining predictions from multiple expression datasets using a single inference method, as shown in Figure 1.6A. For instance, Wang *et al* (Wang et al. 2006) combined networks from multiple time series microarray experiments performed in different conditions to construct a GRN using linear programming. Similarly, Niida *et al* (Niida et al. 2010) built a cancer transcriptional network using a conservative meta-analysis approach. More specifically, a meta-network was deduced after superimposing consistent networks that were predicted, after EEM based algorithm was applied to each of the several cancer microarray experiments. In another study, Steele *et al* (Steele & Tucker 2008) applied statistical meta-analysis approach to construct a consensus Bayesian network by combining edge interactions using results obtained from single Bayesian inference algorithm using multiple microarray datasets. They implemented inverse-variance weighted method (IVWM) (DerSimonian & Laird 1986) as a meta-analysis approach that allowed to aggregate statistical confidence measure attached to each edge from different predicted Bayesian networks. In a recent study, Marbach *et al* (Marbach et al. 2012) built a community-based consensus network by combining the networks predicted by a variety of inference methods, for different microarray datasets measured in diverse model organisms. They employed a vote counting meta-analysis approach, using the Borda count election method (BCEM) to combine the ranks obtained from the different predicted networks.

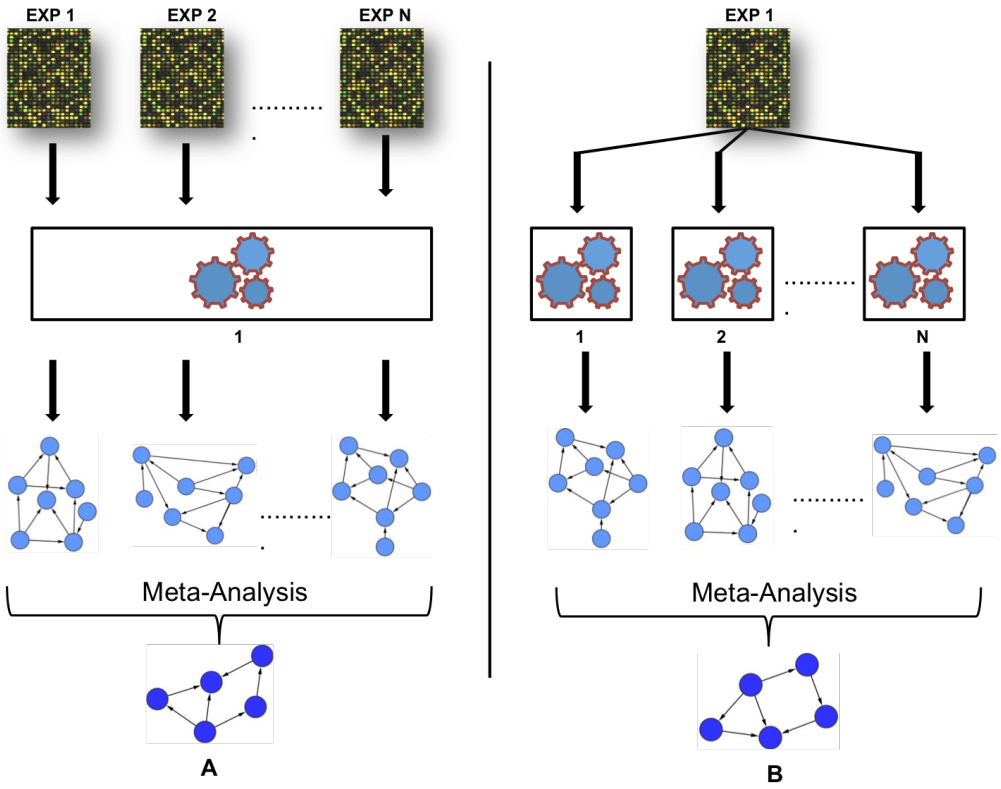


Figure 1.6: A). Common approach used to build a consensus network from multiple microarray experiments using a single inference method. B). Proposed approach to build a consensus network for one experiment using multiple inference methods. EXP indicates microarray experimental conditions; 1, 2 represents different network inference algorithms and N indicates the number of algorithms used.

By contrast to the well-established single inference method approach used in these studies, building a quantitative consensus network from multiple network inference algorithms using a single microarray experimental condition is still in its infancy (Figure 1.6B). Mendoza *et al* (Mendoza & Bazzan 2012), explored the benefits of consensus networks which includes BCEM to optimize the reverse engineering of GRNs on the same expression dataset. However, they focused on building consensus networks from only two network inference algorithms; Boolean networks and Bayesian networks. Indeed, generating an accurate consensus network, and more robust to experimental noise that in this fashion is the most discussed topic in laboratories (Bilal et al. 2015). There has been no detailed

quantitative analysis of consensus networks which requires further investigation. This is where we start this work, thus motivating us to combine the predictions generated from each network algorithm statistically to form an ensemble that delivers robust predictions for one experiment. The collective knowledge obtained by integrating multiple inference methods (the *"Wisdom of crowds")* is greater than that conferred by any individual method (Marbach et al. 2012). Considering the advantages and disadvantages of each inference method is a critical part of this procedure.

1.6.1 Fishers combined probability test

Fisher's combined probability test (FCPT) was proposed by R.A. Fisher (Fisher 1932) for combining *p*-values from a group of independent statistical tests - usually from multiple studies under the same null hypothesis. The null hypothesis states no treatment effect, and the *p*-values of each individual study are independent, uniformly distributed random variables that represent the probability of the observed significance level being attained in the experiment under the null hypothesis. FCPT has been previously employed in genomic research for combining *p*-values. For example, Hess *et al.* (Hess & Iyer 2007) successfully used this method to combine *p*-values from the probe level test of significance for detecting differentially expressed genes from Affymetrix microarray gene expression data.

The FCPT is defined below in the context of combining edges from multiple networks:

$$F_i = -2\sum_{j=1}^{n} \log\left(P_{ij}\right) \approx X_2^{2n} \tag{1.1}$$

Here, F_i signifies the combined *p*-value for a particular edge *i*, P_{ij} represents the edge weight (*p*-value) for the *j*th hypothesis test (i.e. the *j*th network algorithm), and *n* corresponds to the number of independent tests performed (i.e. the number of network algorithms applied). The

score for a candidate edge is calculated by taking the product of the *p*-values computed from each network algorithm, then applying the negative logarithm (Fisher 1932). This measures the approximate chi-square distribution on scaling by a factor of two, with 2*n* degrees of freedom, X_2^{2n}.

Despite its simplicity, the FCPT has the potential to combine extreme probability values generated from independent tests, to deliver robust predictions (Hess & Iyer 2007). In addition to its simplicity, the major advantage of this method is that it allows the gene interaction weights to be standardized to a common scale, and provides probabilistic measures to detect if a gene interaction is significant. This motivated us to investigate its potential to solve the network inference problem.

1.7 Aims and Objectives

In this thesis, we aim to explore the use of consensus learning approaches as a means to enhance the quality and robustness of the predictions made by network inference algorithms for GRN reconstruction. The broader aim of this thesis is to provide a theoretical framework to evaluate some of the more popular qualitative and quantitative consensus techniques used for combining edge predictions from independent inference algorithms. More specifically, the novel contributions of this thesis are outlined below:

1. We developed a new network inference method, referred to as the quantitative consensus network method. This uses FCPT to combine the significance values assigned to each network edge by the inference algorithms to produce a consensus network. We provide evidence in this thesis that FCPT provides a robust and efficient inference method by applying it to a variety of *in silico* benchmark datasets (Chapter

3) and also to some real experimental datasets (Chapter 5). The development of the quantitative consensus network method involved the following:

- A non-parametric based random sampling algorithm was derived, in order to convert the statistical scores associated with each network edge to significance values (*p*-values) (Chapter 3).
- In order to control false positives, the single hypothesis testing strategy from FCPT was then further enhanced using a multiple hypothesis testing strategy - the False Discovery Rate (FDR) control for edge prediction (Chapter 3).

2. We also proposed two new scoring methods: module score and model score. Module score quantifies biologically meaningful modular networks that show statistically significant association of its genes to a biological process, while model score quantifies the ability of a network algorithm to predict biologically relevant modular networks from *in silico* data (Chapter 4) and real experimental data (Chapter 5)

1.8 Thesis Overview

This thesis is organized into six chapters. An abstract is included at the beginning of each chapter summarising its content.

Chapter 1 provides a general introduction to the field of biological networks, which includes the motivation for the study and a description of its novel contributions.

Chapter 2 provides background and literature review of GRNs, whilst introducing the different reverse engineering techniques used in reconstructing GRNs. This chapter is further extended to review the existing qualitative and quantitative consensus learning methods - used to combine multiple network predictions - and to discuss statistical meta-analysis approaches in the field of bioinformatics and medicine.

Chapter 3 investigates a new network inference approach referred to as the quantitative consensus network. This is built on using the FCPT to integrate the predictions obtained from multiple inference algorithms. In this chapter, a new non-parametric algorithm was also presented which uses a random sampling approach by permutation analysis to transform the statistical scores associated with each network edge into significance values (*p*-values) in order to convert all predictions into a common metric. Furthermore, the consensus network by FCPT was tested and validated using a variety of *in silico* expression datasets for different experimental scenarios. We assess and discuss the potential advantages of consensus networks over individual networking methods, and compare existing qualitative and quantitative consensus techniques for robustness and efficiency.

Chapter 4 presents module and model scores by examining existing network inference algorithms for their ability to produce biologically meaningful hierarchical modular networks when tested with *in silico* expression data. Furthermore, the assumptions and limitations surrounding these scores were also described.

Chapter 5 examines the application of the new consensus network algorithm by FCPT to identify genome-wide regulatory interactions from real high-throughput expression data from a simple eukaryote. The performance measures achieved from FCPT were compared against those identified from other qualitative and quantitative consensus methods and other individual networking methods. Furthermore, this chapter presents modular and model scores for biologically meaningful hierarchical modular networks with real data.

Chapter 6 concludes the thesis. The limitations of this present study and the direction of future works are also discussed, relating back to the claims made in Chapter 1.

Chapter 2

Background and literature review

Abstract

This chapter provides some background on the various types of existing network inference approaches currently used to study GRNs, further extending to provide a brief overview of the existing qualitative and quantitative consensus approaches (meta-analysis) currently applied for combining predictions from multiple network inference methods.

2.1 Modelling gene regulatory networks

The primary objective of modelling a gene regulatory network (GRN) is to identify the following types of interactions: 1) Physical (protein-gene) interactions – these occur between a transcription factor (TF - a protein) and its target genes, i.e. TF binding to a sequence motif on the promoter of a target gene; 2) Influence (gene-gene) interactions – such interactions encapsulate a causal relationship between two genes by relating the expression of a gene *i* to that of a gene *j* (Bansal et al. 2007).

In the last decade, many network inference approaches have been developed which are used to reconstruct GRNs from microarray gene expression data. The methods predominantly used are broadly classified into the following categories:

1. Information theory models
2. Bayesian network models
3. Differential equation models

Depending on the type of gene expression data available, the network inference algorithm can be chosen accordingly to predict regulatory interactions, as shown schematically in Figure 2.1.

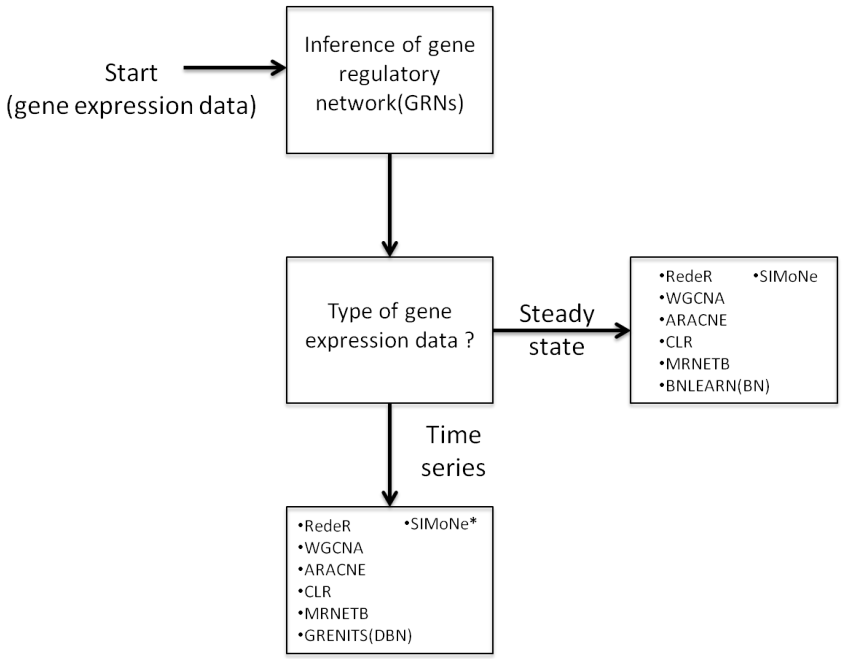


Figure 2.1: Flow chart for choosing a suitable network inference algorithm depending on the type of gene expression data used. (BN): Bayesian network; (DBN): Dynamic Bayesian network; (*): Algorithm that requires to change parameters depending on the type of the data.

2.1.1 Information theory models

2.1.1.1 Correlation networks

One of the simplest network modelling approaches is the correlation based network (Stuart et al. 2003). Here, the interaction between each pair of genes is weighted using the Pearson or Spearman correlation coefficients computed from their expression profiles, resulting in an undirected network. Two genes are characterised as connected only if the correlation coefficient between their expressions is above a specified threshold. The value of the threshold determines the sparseness of the network. These networks are also known as co-expression networks and capture the linear dependence between genes. A correlation coefficient close to zero is a strong indicator of independence between any gene pair.

The Pearson Correlation Coefficient (PCC) r_{xy} is calculated between gene x and target gene y as shown in equation (2.1):

$$r_{xy} = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2}} \quad (2.1)$$

In this equation, n represents the number of experimental sampled measurements of gene x and gene y. Correlation coefficients ranges between +1 and -1. A high positive correlation indicates high similarity between the expression profiles of two genes (x and y). While, high negative correlation indicates the expression profiles of both genes are in opposite direction.

Spearman's Rank Correlation Coefficient ρ_{xy} instead uses ranked expression profiles to calculate the distance measure between genes x and y using PCC as specified in equation (2.2)

$$\rho_{xy} = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \quad (2.2)$$

Here, d_i signifies the difference in rank order between genes x and y over n sample measurements.

The correlation based benchmark algorithms used for consensus analysis are RedeR and WGCNA that are described below.

RedeR

The RedeR algorithm reconstructs a hierarchical nested network using gene expression data (Castro et al. 2012). It manages and organizes network data structure using mixed graphs in two different layers. In the first layer, a directed acyclic graph (DAG) is defined where each

node has one parent, multiple branches, and no feedback cycles. The second layer connects the DAG components to produce a hierarchical topology in an undirected graph (UDG), as illustrated in Figure 2.2.

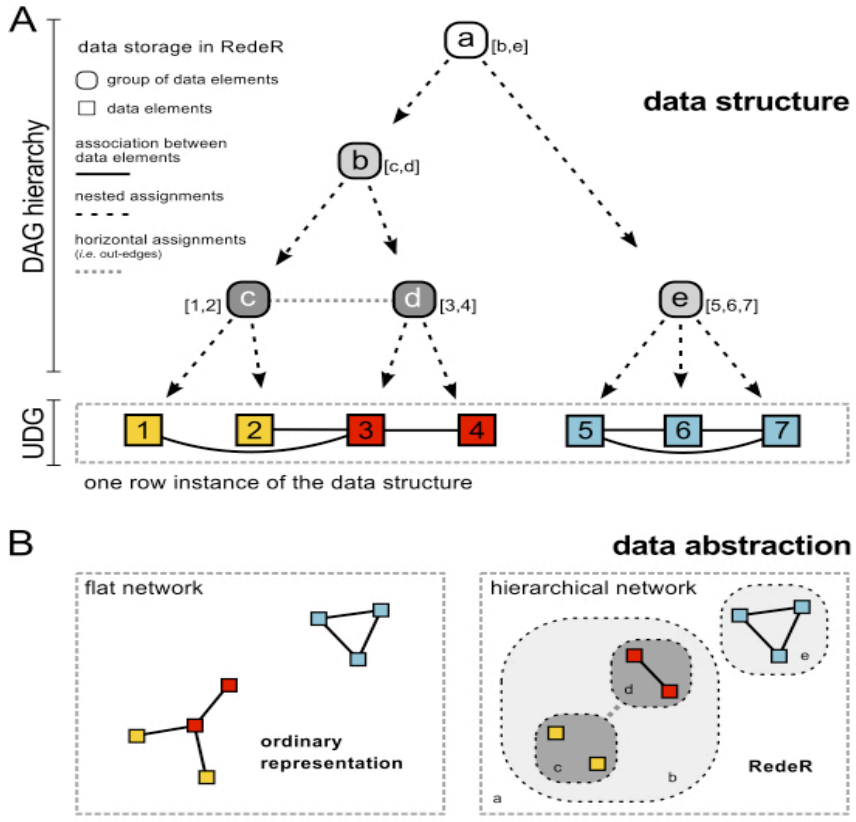


Figure 2.2: Hierarchical modular network structure from RedeR. A) Data structure connecting a directed acyclic graph (DAG) and an undirected graph (UDG) in two different layers. Here, letters a,b,c,d and e denote modules and numbers 1 to 7 represents genes. B) Data abstraction outlines the hierarchical network (right) in contrast to the flat network (left). The figure was adapted from (Castro et al. 2012).

Euclidian distance was calculated to derive a dendogram from complete linkage clustering for reconstructing the hierarchical network. The associations between co-expressed genes were calculated using Spearman's rank correlation coefficient, ρ, as defined in equation (2.2).

WGCNA

The WGCNA (Weighted Correlation Network Analysis) algorithm produces a modular network of highly correlated genes using an unsupervised clustering technique in a two step process (Langfelder & Horvath 2008).

In the first step, the signed co-expression similarity network S_{ij} is defined and utilized as an intermediate quantity to calculate a weighted adjacency matrix A_{ij}. The S_{ij} network is computed using co-expression (using Pearson Correlation Coefficient) measures that identify interacting patterns between gene *i* and gene *j*. The weighted adjacency matrix is calculated by raising the co-expression similarity S_{ij} to a soft power β:

$$\mathrm{S}_{ij} = \frac{1 + corr(x_i, x_j)}{2} \tag{2.3}$$

$$A_{ij} = \mathrm{S}_{ij}^{\beta} \tag{2.4}$$

The values of A_{ij} range between 0 and 1, denoting minimum and maximum edge strength respectively between node *i* and node *j*. β was fixed at its default value of 6 in our analysis. The correlation coefficients were transformed back to derive PCC values using the modified equation below.

$$r_{ij} = 2A_{ij}^{(1/\beta)} - 1 \tag{2.5}$$

The second step identifies functional modules associated with the co-expression network using a hierarchical clustering method. The dendogram associated with these hierarchical clustering branches correspond to modules. To derive the desired number of modules, the hierarchical tree was cut at a desired height and correspondingly the optimized Module Eigen dissimilarity threshold was fixed (Langfelder & Horvath 2008).

WGCNA has successfully been used to determine cluster modules from microarray gene expression data in the yeast cell cycle, the human brain and the mice liver (Langfelder & Horvath 2008).

2.1.1.2 Mutual information

Mutual information based networks rely on the entropy scores (known as Shannon's entropy) computed from gene expression measurements that indicates how much information obtained from the expression profile of one gene predicts the behavior of the other gene (Steuer et al. 2002). Like correlation analysis, mutual information determines the degree of statistical interconnection between two random gene variables. However, mutual information captures the degree of non-linear dependence between two genes based on their discretised expression profiles. Given two random variables X_i and X_j representing the expression levels of two genes *i* and *j*, the mutual information (MI) between gene *i* and gene *j* is defined as

$$MI_{ij} = H_i + H_j - H_{ij} \tag{2.6}$$

where

$$H_i = \sum_x p\left(X_i = x\right) \log_2 p\left(X_i = x\right) \tag{2.7}$$

is the entropy for the expression of gene *i* - a measure of information content in the distribution pattern of expression levels across measurements – and

$$H_{ij} = -\sum_x \sum_y p\left(X_i = x, X_j = y\right) \log_2 p\left(X_i = x, X_j = y\right) \tag{2.8}$$

is the joint entropy for genes *i* and *j*. Entropy is calculated using discrete probabilities, and therefore applies histogram techniques. The entropy is higher when the distribution of gene expression is more randomly distributed and reaches a maximum when distribution is

uniform. From this definition, the two random gene variables X_i and X_j are statistically independent if the MI is zero, i.e. if the joint entropy $H_{ij}=H_i+H_j$, meaning that $p(X_i=x,X_j=y)=p(X_i=x)p(X_j=y)$. A higher MI indicates that the two gene variables are non-randomly associated.

The mutual information derived statistic scores by weight are applied by network inference algorithms like ARACNE (Algorithm for the Reverse engineering of Accurate Cellular Networks) (Margolin et al. 2006), CLR (Context Likelihood to Relatedness) (Faith et al. 2007), and MRNETB (Maximum Relevance Minimum Redundancy Backward) (Meyer et al. 2010) to study large scale regulatory networks.

ARACNE

ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks) identifies the transcriptional regulatory network (TRN) between genes and their products using microarray gene expression data (Margolin et al. 2006). ARACNE predicts the association between genes through statistical dependency in two main steps.

In the first step, ARACNE derives a MI matrix, $M_{ij}=MI_{ij}$ for all input pairs of genes i and j in the expression dataset using the definition in equation (2.6). The gene expression data is continuous, so is discretized with the equal width binning method. The empirical probability distribution estimator for the assessment of a mutual information score is applied using the function build.mm with the number of bins set to $\sqrt{n}$, where n denotes the number of experimental samples (Meyer et al. 2008).

The second step is a pruning procedure based on the Data Processing Inequality (DPI). The DPI is formally defined using a triplet of nodes $\{X_i;X_j;X_k\}$, where gene X_i interacts with gene X_j through gene X_k ($X_i \rightarrow X_j \rightarrow X_k$) then the edge that is weakest, which is considered as an indirect interaction, say $\{X_i;X_j\}$ is removed if the mutual information weight

is below $\min\{M_{ik},M_{jk}\}$-*eps*, where *eps* is a numerical threshold that is set to 0.15. The *eps* was relaxed from the default (0.05) as it was observed to be too stringent in our study. ARACNE has been successfully applied to study TRNs in human B cells and has outperformed Bayesian networks and several other inference methods (Margolin et al. 2006).

CLR

The CLR (Context Likelihood to Relatedness) (Faith et al. 2007) algorithm is built upon a MI-based relevance algorithm that is primarily used for clustering (Butte & Kohane 2000). The CLR algorithm has two main steps. In the first step, it calculates the MI matrix, M_{ij}, for all input pairs of genes *i* and *j* in the expression dataset (as in ARACNE). In the second step, the algorithm eliminates false interactions by computing Z-scores. For each input pair of genes *i* and *j*, a Z-score, Z_{ij} is calculated from an empirical MI density for all regulators of the target gene Z_j and an empirical MI density for all targets of the regulator gene Z_i. CLR identifies possible interactions whereby MI values are significantly above the empirical distribution of MI values. That is, instead of considering mutual information values M_{ij} for random gene variables X_i and X_j, it calculates Z-scores, $Z_{ij} = \sqrt{Z_i^2 + Z_j^2}$ where,

$$Z_i = \max_k \left(0, \frac{M_{ik} - \mu_i}{\sigma_i}\right) \qquad (2.9)$$

In equation (2.9), μ_i represents the mean and σ_i the standard deviation of the empirical distribution of mutual information values $\{M_{ik};\ k=1,\ldots,n\}$. The CLR algorithm has been successfully applied to decipher the *E.coli* TRN (Faith et al. 2007) .

MRNETB

The MRNETB (Maximum Relevance Minimum Redundancy Backward) algorithm is an improved version of MRNET that depends on the feature selection strategy known as MRMR (Maximum Relevance Minimum Redundancy). That is, to select genes, a sequence of supervised learning is applied by MRMR, wherein each gene is played as a regulator. For example, consider a supervised learning task, where X is a set of input variables and Y is the output. A score, S is used to sort X by rank, calculated using the difference between maximum relevance (MI of output gene variable Y) and minimum redundancy (mean MI of the penultimate ranked gene variable X). The higher ranked variables indicate direct interactions whereas lower ranked variables are considered as indirect interactions. Specifically, MRMR starts by selecting a variable X_k that has the highest mutual information M_{kj} to the target X_j. It then selects the variable X_i that has high mutual information M_{ij} to the target X_j and at the same time has low mutual information M_{kj} to the previously selected variable. A major limitation of MRNET is that it used forward selection strategy that strongly depends on the first variable selected (i.e. variable having the highest MI with the target gene). If the first variable selected is not a true target then maximizing MRMR may not be advantageous. In contrast, MRNETB uses a backward elimination combined with sequential search to rank all candidate edges (Meyer et al. 2010).

MRNETB infers edges in a two-step process. In the first step, it estimates MI values same as in ARACNE and CLR. In the second step, MRNETB initiates the selection of edges from a set X_{Sj} through backward elimination employing the MRMR principle to rank features using the score S_j containing all variables ($X_{Sj} \subseteq X \setminus X_i$) and then removes X_i iteratively that actuates maximal increase of the X_{Sj} score until the termination criteria is reached i.e. when the relevance term, is greater than the redundancy term.

The enhancement of the process is achieved by sequential replacement, where at each step, the status of selected and non-selected variables is swapped so that the maximal increase in the objective function (i.e. X_{Sj} score) is reached. MRNETB algorithm was implemented using the minet package (Meyer et al. 2008), and it has been previously applied to study SynTReN derived and DREAM4 challenge benchmark datasets (Meyer et al. 2010).

2.1.1.3 Graphical Gaussian models

Graphical Gaussian models (GGMs) are another class of information theory models, which are generally known as covariance graph models, and have recently become quite popular when studying gene regulatory networks using expression data (Edwards 2000). These models use partial correlation measures to identify conditional dependency between any two pairs of genes. A partial correlation determines the relationship between two random gene variables, by removing the effect of other gene variables. Unlike the correlation coefficient, partial correlation is computed using a concentration matrix, which is calculated by taking the inverse of the correlation matrix. These values provide a strong measure of dependence between genes and if the edge is missed by GGMs, it relates to conditional independence (Markowetz & Spang 2007).

SIMoNe

The Statistical Inference for MOdular Networks (SIMoNe) algorithm employs GGMs to infer edges through latent clustering (Chiquet et al. 2009). It has been used to investigate modularity in GRNs from gene expression data from breast cancer patients (Chiquet et al. 2009). SIMoNe is a mixture version of a graphical lasso as it favors network sparsity. Latent network structure is adapted to enhance estimation accuracy. In addition, the inference of a modular network is driven by latent network structure through penalization procedure of

nodes to be connected. The SIMoNe algorithm uses either steady state or time series expression data, which is defined in the *type* parameter of *simone* function. In the function, the number of penalties parameter, which is used for penalizing the edges, was fixed to 50 instead of default (100) to avoid memory crashes. The Bayesian information criteria (BIC) scoring function is adapted to select a particular network structure to identify cluster modules. The cluster size was determined by fixing the *cluster.qmin/qmax* parameter to number of cluster modules desired. Other parameters were kept as default.

2.1.2 Bayesian network models

Bayesian networks (Friedman et al. 2000; Hartemink & Gifford 2001) are probability distribution based graphical models which are able to capture properties of conditional dependence between random gene variables (nodes), X = X_1,..X_i,..X_n.. The edge connecting two such nodes, $X_i \rightarrow X_j$ indicates probabilistic dependence illustrated using a directed edge. The dependency strength (edge) between variables for each node (X_i) is measured by conditional probability distributions $P(X_i \mid pa(X_i))$ under the assumption that the variables are discrete. Here, $pa(X_i)$ indicates parents of node, X_i. For example, if there is an interaction (edge) from node X_i to node X_j, then node X_i is considered to be the parent of node X_j and node X_j is the descendent or child of node X_i. BNs provide a powerful modeling approach as they are able to encapsulate the type of interaction (activation or inhibition) between gene variables and its relative strength (Yu et al. 2004). BNs also have the potential to distinguish between direct and indirect interactions, providing a key advantage over correlation based approaches (Werhli et al. 2006).

The set of random variables in BNs is represented by joint probability distributions (JPD). Considering *n* gene variables in BN from X_1 to X_n, the JPD is indicated by

$$P\left(X_1 = x_1, X_2 = x_2, ..., X_n = x_n\right) \sim P\left(x_1, x_2, ..., x_n\right)$$

By applying chain rule from probability theory the local probability distribution is described for each node.

$$\begin{aligned} P\left(x_1, x_2, ..., x_n\right) &= P(x_1) \times P(x_2 \mid x_1) ...., P(x_n \mid x_1, ..., x_{n-1}) \\ &= \prod_{i=1}^{n} P(x_i \mid x_1, ..., x_{i-1}) \end{aligned} \tag{2.10}$$

For all the other nodes that is conditional independent given its parents can be factorized using joint probability distributions *P(x)* shown below

$$P(x) = \prod_{i=1}^{n} P(x_i \mid pa(x_i)) \tag{2.11}$$

Here, $pa(x_i)) \subseteq \{X_1,\ldots,X_{i-1}\}$ denotes the parent state vector that corresponds to regulatory activity measure of a gene where x_i indicates current state vector. This approach reconstructs the regulatory network by calculating a product of conditional probabilities using Bayes' theorem:

$$P(X_i \mid X_j) = \frac{P(X_i)P(X_j \mid X_i)}{P(X_j)} \tag{2.12}$$

Where, $P(X_i \mid X_j)$ represents conditional probability, $P(X_j)$ and $P(X_j|X_i)$ indicate prior probability and likelihood respectively.

Bayes' rule is applied to understand the stochastic nature of a gene regulation. The advantage of this inference method is that different datasets can be combined; it also takes into account prior biological knowledge for reconstructing the GRN. Other advantages include the avoidance of over-fitting the training data to the model whilst also being able to incorporate hidden variables (e.g. TF activity) and noisy measurement data. Essentially, there are three

imperative steps to learn BNs. The first critical step is selection of the model that defines network structure through directed acyclic graphs (DAGs) that shows regulatory relationships between nodes. The selection of the model is generally deduced by heuristics for instance, greedy-hill climbing, simulated annealing or Markov Chain Monte Carlo (MCMC) to learn BNs efficiently. Second, is fitting of parameters for each node in graph using conditional probabilities given a discrete and continuous gene expression values. Finally, evaluating the fitness of each model using a score function. The model that yields the highest score is inferred as better gene network model.

In their simplest form, BNs can only infer DAGs (i.e. no feedback loops). Dynamic Bayesian Networks (DBNs) overcome this restriction by enabling time course dependent expression data to be used (Kim et al. 2003; Zou & Conzen 2005). BNLEARN (Scutari 2009) and GRENITS (Morrissey 2013) are Bayesian algorithms which infer static and dynamic gene networks using steady state and time series gene expression data respectively.

2.1.2.1.1 Static Bayesian Networks

It is common practice to apply BNs in order to study network structures using steady state microarray gene expression data. Accordingly, BNs were compared to other networking inference methods that use such data. A search and score based approach was employed to learn the Bayesian network structure. This class of heuristic optimization algorithms scores each network by exploring all possible systems in the search space. Finally, a network with the highest objective function score is selected. A variety of score based learning algorithms exist; however; due to its simplicity, greedy hill climbing was implemented in the study using the BNLEARN package (Scutari 2009). As recommended by the author, prior to learning the structure, the gene expression data was discretized. In order to score each candidate network, the Bayesian Information Criterion (BIC) was implemented, as it considers the number of

perturbation experiments (samples) present in the gene expression dataset. All other parameters were fixed at their default values.

2.1.2.1.2 Dynamic Bayesian Networks

The BNs are further extended to DBNs which typically use time series microarray data for modeling GRNs. (Zou & Conzen 2005). The advantage of DBNs is that they are capable of constructing cyclic networks with feedback loops. Thus, different network inference algorithms using time dependent microarray datasets were compared against DBNs to evaluate their performance. DBNs were studied using the Bioconductor package GRENITS (Gene Regulatory Network Inference using Time Series) (Morrissey 2013). GRENITS predicts regulatory interactions by calculating posterior probabilities using MCMC (Markov Chain Monte Carlo) simulations. MCMC is an efficient, fair algorithm for sampling from high dimensional probability distributions using random numbers drawn from uniform probability in a certain range. Essentially, the objective is to figure out a global optimum from the posterior probability distribution. Generally, posteriors are not well formulated and often inconsistent. By using MCMC, integration statistics is summarized over random numbers generated from Markov chains as a means of illustrating model convergence by assessing visual diagnostic plots. A Markov chain is a mathematical model for stochastic systems, with discrete or continuous states governed by a transition probability. The current state in a Markov chain is dependent only on the most recent state (Borovkov 2003). Two Markov Chains are generated during the simulation process, and the network is deduced based on the convergence of link probabilities. Furthermore, a network probability matrix is generated which estimates probability scores for each gene as a regulator.

2.1.3 Differential equation models

Differential equation models describe the changes in the pattern of gene expression over time, taking into account environmental conditions and the expression levels of other genes (Hecker et al. 2009). Therefore, modeling GRNs using this approach reproduces the dynamic behavior of networks in a quantitative manner.

The general ordinary differential equation (ODEs) model for studying the dynamics of expression data as shown in equation (2.13):

$$\frac{dx}{dt} = f(x, p, u, t) \tag{2.13}$$

Here, $x(t)$ = $(x_1(t),x_2(t),x_3(t),x_4(t)....x_n(t))$ represents the expression values of genes $\{1,2,3,...,n\}$ at time t, f represents the function defining the rate of change of each state variable x_i for parameter set p and u represents the external signal or environmental perturbation. Determining the parameters p for the function f from the measured signals u and x using an optimization algorithm leads to network inference. Typically, the gene regulatory process is characterized by non-linear dynamics. However, most of the inference approaches are based on linear ODE models.

ODE models are deterministic, unlike information theory models and Bayesian network models, which apply conditional probabilistic approaches. ODE based approaches result in directed networks and are used to study both time series and steady state expression data. Microarray Network Identification (MNI) (di Bernardo et al. 2005) and Time Series network Identification (TSI) (Bansal et al. 2006) use ODEs to infer GRNs from steady and time series gene expression data respectively.

2.1.4 Other approaches

2.1.4.1 Boolean networks

Boolean networks were first proposed by Kauffman (Kauffman 1969), and since then they have been used to investigate GRNs. This modeling approach employs discrete dynamics using binary variables, $X_i \in \{0,1\}$ which define the state of gene i as active (X_i=1) or inactive (X_i=0), i.e. as on or off. Discretization is an important step in Boolean network modeling, where continuous gene expression data is transformed (discretized) into a binary format. These networks show directed graphs and the edge between two nodes as a function of Boolean operations (i.e. OR, AND, NOT), rather than a statistical score.

REVEAL (REVerse Engineering Algorithm) (Liang et al. 1998) was one of the first Boolean algorithms proposed to predict the transition between a gene's state at t and its state at t+1, although Liang *et al*. did not specifically use microarray gene data in their study.

2.1.4.2 Clustering Algorithms

Clustering, is not a network inference algorithm *per se*, but is a technique used to visualize and identify those genes possessing similar expression profiles across different experimental perturbations or time points (Bansal et al. 2007). The underlying assumption of this method is that the genes have similar expression dynamics (are co-expressed) within a cluster, being possibly regulated by a common TF or perhaps belonging to the same pathway or having the same biological function (Richards et al. 2008). The function of known genes within a cluster and of un-annotated genes can be characterized using statistical methods like gene ontology (GO) enrichment analysis and/or promoter sequence enrichment analysis; these provide a powerful approach to identifying biological processes and hidden regulatory variables of uncharacterized genes within individual clusters. Although many clustering algorithms exist, those that are most widely used for analyzing gene expression data are broadly classified into

partition based, or hierarchical clustering algorithms (D'haeseleer et al. 2005; Costa et al. 2004). Partition based algorithms employ an unsupervised learning approach to partition genes into a predefined number of cluster structures without any hierarchy. The predominant algorithms of this class are *k*-means and the self-organizing map (SOM) (Tavazoie et al. 1999; Tamayo et al. 1999). Hierarchical algorithms divide sub-clusters into smaller clusters which form a hierarchal architecture (Eisen & Spellman 1998). Hierarchical agglomerative clustering (HAC), and Hierarchical divisive clustering (HDC) is a classic example in this class.

2.2 Consensus methods

In statistics, the aim of meta-analysis is to estimate the combined or overall effect of independent studies possessing similar research hypotheses. The major advantages of using meta-analysis approaches are to increase the power or sensitivity over individual studies to resolve uncertainties where prediction results disagree, and to improve the estimate of effect size (Borenstein & Rothstein 2007). Effect size is a quantifiable measure that allows comparing effectiveness of individual study against another study. In this section, we will discuss popular statistical meta-analysis methods used in medical and genomics research for consensus decision-making.

2.3 Qualitative approaches

Qualitative approaches seek to build a naïve consensus by uncovering the pattern or trends of regulatory interactions from different inference algorithms using binary decision making, without much emphasis on numerical values. The most common qualitative approaches in

this context are the intersection and union method which use the principle of the naïve Venn-diagram to combine predicted networks (Joshi et al. 2014).

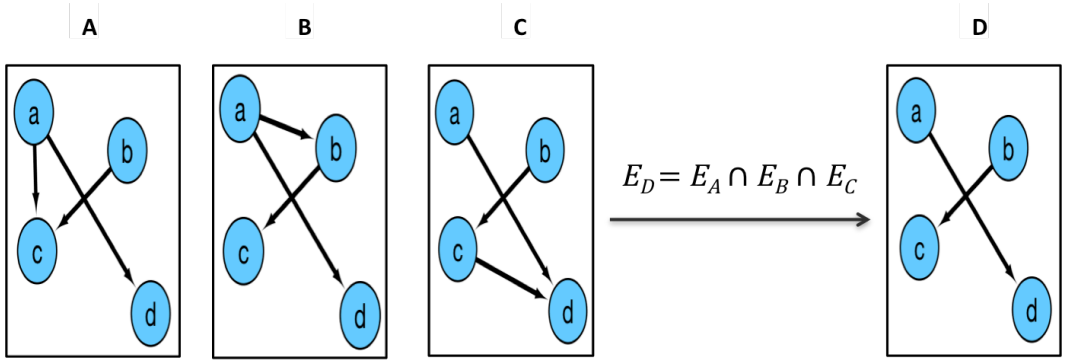
2.3.1 Intersection method

The intersection method utilises the traditional community based-approach (De Smet & Marchal 2010) to infer network structure from the set of common predicted edges generated by individual inference algorithms.

Edge selection strategy

The interactions are chosen to be part of the naïve consensus network only if the edge is predicted to exist by all the network algorithms for a single expression dataset. Figure 2.3 demonstrates this edge selection strategy for an example network comprising four genes, a, b, c and d, where the interactions are predicted by three network inference algorithms, A, B and C. For each inferred network, a corresponding adjacency matrix encodes the connectivity between genes, where 1 denotes a connection between two nodes and 0 denotes no connection. The final naïve consensus network D contains only edges predicted by all of the network inference algorithms. For simplicity, in the example presented here, we have considered directed interactions and not signed interactions.

The advantage of this approach is that it simplifies the analysis by overlapping interactions, thereby delivering a consensus network in the most conservative way. However, the disadvantage of this strategy is it results fewer interactions and does not provide a quantifiable consensus edge measure, as different network algorithms give heterogeneous edge scores, resulting in lower sensitivity.



	a	b	c	d
a	0	0	1	1
b	0	0	1	0
c	0	0	0	0
d	0	0	0	0

	a	b	c	d
a	0	1	0	1
b	0	0	1	0
c	0	0	0	0
d	0	0	0	0

	a	b	c	d
a	0	0	0	1
b	0	0	1	0
c	0	0	0	1
d	0	0	0	0

	a	b	c	d
a	0	0	0	1
b	0	0	1	0
c	0	0	0	0
d	0	0	0	0

Figure 2.3: A sample intersection consensus network D and corresponding adjacency matrix, derived from the networks A, B and C predicted by different inference algorithms. a, b, c and d denote network nodes (genes). Edges (E) signify their regulatory relationship.

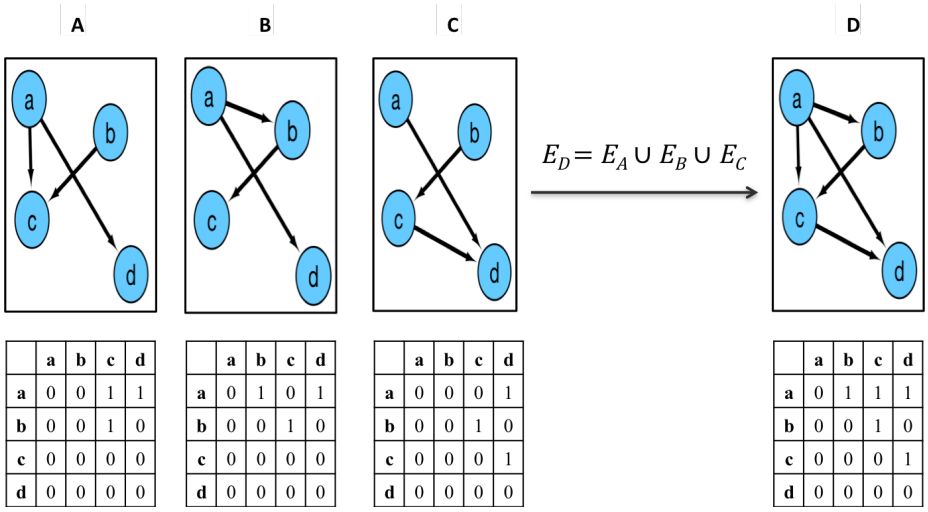
This means, the sensitivity measures the fractions of consistent edges that are actually true. Low consistent edge predictions from different network inference algorithms can be attributed to lower sensitivity. The naïve consensus approach has been previously applied by Steele *et al* (Steele & Tucker 2008) to generate a consensus Bayesian network by identifying consistent interactions across different gene expression datasets for yeast and *E.coli*.

2.3.2 Union method

The union method is an alternative to the naïve consensus approach. The resultant consensus network consists of edges that are predicted by any of the inference algorithms.

Edge selection strategy

Figure 2.4 demonstrates the edge selection strategy for this approach (see Figure 2.3). An edge is chosen to be in the final consensus network if it is present in any one of the predicted networks A, B or C.



	a	b	c	d
a	0	0	1	1
b	0	0	1	0
c	0	0	0	0
d	0	0	0	0

	a	b	c	d
a	0	1	0	1
b	0	0	1	0
c	0	0	0	0
d	0	0	0	0

	a	b	c	d
a	0	0	0	1
b	0	0	1	0
c	0	0	0	1
d	0	0	0	0

	a	b	c	d
a	0	1	1	1
b	0	0	1	0
c	0	0	0	1
d	0	0	0	0

Figure 2.4: A sample union consensus network D and corresponding adjacency matrix derived from the networks A, B and C predicted by different inference algorithms. The other plot attributes are the same as in Figure 2.3.

The advantage of this approach is that none of the predicted interactions are missed in the final consensus network, yielding increased sensitivity. However, a drawback of this approach is that many false positives may occur due to the higher proportion of extracted interactions. It also does not provide quantifiable measures for evaluating the strength of each interaction. Consequently, despite being minimally conservative, this method is not popular among the network community. The union method was, however, recently applied by Joshi *et al.* (Joshi et al. 2014) to integrate the predictions generated from the CLR algorithm using multi-expression datasets of several different *Drosophila* species.

2.4 Quantitative approaches

Quantitative methods generate weighted consensus networks for decision-making (Mendoza & Bazzan 2012). An example network is shown in Figure 2.5 where numerical values are assigned to a predicted network edge (a→d) by three network inference algorithms A, B and C. The edge, a→d in the consensus network, D requires a numerical value to derive a weighted consensus that combines heterogeneous weights associated with the same edge from the networks A, B and C. To combine edge weights statistically, many quantitative meta-analysis approaches can be applied which are discussed in the next section.

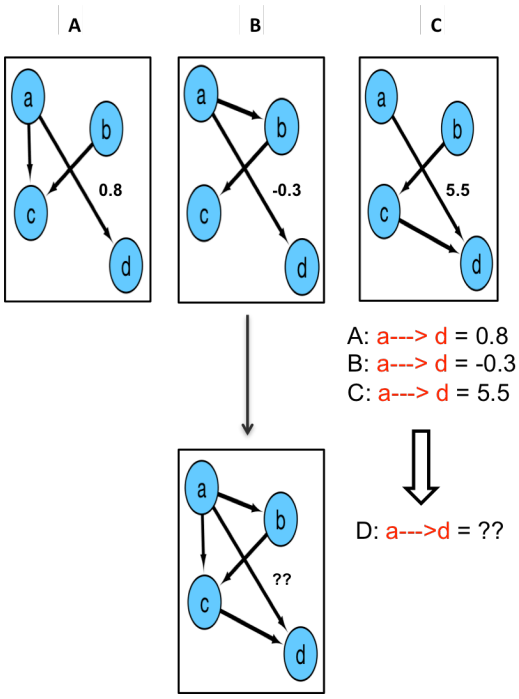


Figure 2.5: A sample quantitative consensus network D derived from the networks A, B and C predicted by different inference algorithms. a, b, c and d denote nodes (genes). Edges (E) signify regulatory relationships; the value associated with an edge (e.g. a→d) signifies the strength of the interaction whilst the sign indicates the type of interaction (positive – activation; negative – repression).

2.4.1 Combining p-values

In statistics, integrating information by combining *p*-values has a long history (Tseng et al. 2012). A *p*-value indicates the probability of the observed experiment under the null hypothesis. In the context of the network inference, many network algorithms do not deliver *p*-values as edge weights. In contrast, they provide heterogeneous edge scores. Therefore, the challenge to implement this strategy to derive weighted consensus network requires consistency of edge scores (i.e. estimated *p*-value for each edge score). However, the advantage of this approach is that it allows heterogeneous edge scores obtained from independent studies (network algorithms) to be converted into a common metric (*p*-values). That is, when the edge scores from independent algorithms are not binary (Figure 2.5), *p*-values can still be calculated. However, effect sizes may not be well defined. The major advantage of combining *p*-values over other popular meta-analysis is that its simplicity and extensibility to various kinds of edge scores. Some of the popular methods used for combining *p*-values are discussed below. It should be noted that Fishers combined probability test (FCPT) is the most popular such method, and was described in Chapter 1.

2.4.1.1 Inverse-variance weighted method

The inverse-variance weighted method (IVWM) (DerSimonian & Laird 1986) is a meta-analysis approach that like FCPT, integrates the statistical significance estimates T_{ij} attached to each edge e_{ij} to generate a final aggregated edge score $\overline{T}_{ij}$, over the networks predicted by n inference algorithms (Steele & Tucker 2008).

$$\log \overline{T}_{ij} = \frac{\sum_{k=1}^{n} w_k\left(e_{ij}\right) \log T_{ij}}{\sum_{k=1}^{n} w_k\left(e_{ij}\right)} \tag{2.14}$$

In the above, $w_k(e_{ij})$ equals to the number of networks that predicts the existence of edge e_{ij}

2.4.1.2 Stouffer's Z-score

Stouffer's method (Stouffer S.A 1949) adopts the inverse normal transformation procedure. Unlike Fisher's method, it avoids log-transformation by using Z-score statistics to combine *p*-values. The Z-score approximates the standard normal distribution under the null hypothesis. The advantage of the Z-score is that it allows incorporating study specific weights. Traditionally, the weights characterize the sample size or the effect size of a particular study (Li & Ghosh 2014; Zakin 2011). This means, the weights allow us to determine how informative a particular study is. However, this method is only appropriate for one-sided right-tailed *p*-values.

The Z-score is an overall meta-analysis measure calculated by taking the average of the Z_i values obtained from *n* independent tests. It is calculated as

$$Z \sim \frac{\sum_{i=1}^{n} Z_i}{\sqrt{n}} \tag{2.15}$$

Where, $Z_i = \varphi^{-1}(1 - p_i)$, ϕ^{-1} is the inverse cumulative distribution function of the standard normal distribution on the common null hypothesis and p_i represents the significance value (*p*-value) for the *i*th hypothesis test. Stouffer's Z-score method has been implemented to combine the *p*-values of a probe set using acute lymphoblastic leukemia and osteoarthritis gene expression data (Geistlinger 2008).

Since our motivation is aligned more towards forming an ensemble of predictions that incorporates heterogeneous edge weights, rather than determining the effect size of independent studies (network algorithms), we choose not to employ Stouffers Z-score. Furthermore, incorrectly assigning study specific weights can reduce the power of

combination yielding unreliable results. For the normal distribution, FCPT and Stouffer's test yield similar power when the number of independent tests is equal (Chen 2011; Zakin 2011), and Stouffer's test shows high linearity with FCPT when the number of independent studies are initially small (less than 5) (Ion Mandoiu 2008).

2.4.1.3 Combining effect sizes

An alternative way of combining multiple studies using meta-analysis is based on the assumption of standardized effect size. The effect size is a quantitative measure indicating the strength or magnitude of empirical research findings. If all independent studies demonstrate equally precise results, then one could simply compute the average of the effect sizes. However, in reality some studies are more precise than others and one would want to assign more weights to the studies that carried more information (Borenstein & Rothstein 2007). Meta-analysis solves the latter issue by computing a weighted average of effect sizes, where more weight is assigned to some studies and less to others. The two popular approaches for combining effect sizes are the fixed effect and random effect models. These are discussed in turn below.

2.4.1.3.1 Fixed effect model

The fixed effect model (FEM) makes the assumption of homogeneity between the results obtained across all the independent studies and therefore has one true effect size shared by all the studies. In FEM, a weight w_i is assigned to every individual study i that is inversely proportional to its variance V_i:

$$w_i = \frac{1}{V_i} \tag{2.16}$$

In some studies, inverse variance is approximately equal to the sample size although it is considered a nuisance measure and serves to reduce the variance of the combined effect (Borenstein & Rothstein 2007). The mean weight $\overline{T}$ is calculated as

$$\overline{T} = \frac{\sum_{i=1}^{n} w_i T_i}{\sum_{i=1}^{n} w_i} \qquad (2.17)$$

Here, T_i denotes the estimated effect size in study *i,* given by $T_i = \mu + \varepsilon_i$, where μ indicates the common true effect across all the studies, ε_i represents the measurement error for study *i* and *n* corresponds to the number of independent tests.

Steele *et al* (Steele & Tucker 2008) used the inverse variance weighted method to combine the statistical confidence estimates attached to each edge over a set of bootstrapped Bayesian networks to build a consensus network.

2.4.1.3.2 Random effect model

The random effect model (REM) allows the true effect to vary randomly across all the studies (i.e. it assumes that heterogeneity exists), rather than assuming that the true effect is identical in all studies. The weights in REM are estimated similarly to FEM, but use an additional variable, τ^2 to adjust those weights in order to incorporate variability between studies as well as variation within each study itself (Goldstein 2005). The adjusted weight w_i^* is calculated as

$$w_i^* = \frac{1}{(1/w_i) + \tau^2} \qquad (2.18)$$

Where τ^2 estimates the variability between studies. The estimated mean weight is shown in equation (2.17), but with adjusted weight w_i^* in place of w_i. Likewise, the variance of adjusted weight is given by

$$\frac{1}{\sum_{i=1}^{n} w_i^*} \tag{2.19}$$

where n corresponds to the number of independent tests. Choi *et al* (Choi et al. 2003) were the first to employ this REM meta-analysis approach, combining multiple microarray gene expression datasets, taking into account inter study variation.

2.4.2 Combining ranks

One of the issues of combining effect sizes or *p*-values is that the results can contribute to outliers. This can lead to a significant problem when one needs to analyse many noisy independent experiments concomitantly. Robust rank statistics methods are used to alleviate this problem as they calculate the ranks of each variable under each independent study. Calculating the product and mean of ranks across studies is becoming one of the most popular statistical tests for detecting differential gene expression from multiple microarray experiments (Tseng et al. 2012). The Borda vote-counting method has proved to be an effective way of combining ranks across studies in genomics research.

2.4.2.1.1 Borda counting

A commonly used strategy that combines ranks using vote counting is the Borda Count Election Method (BCEM). This ranks true gene interaction and non-interaction pairs in order of their probabilities for each statistically independent inference method (independent studies). A community consensus network is then constructed based on the best average rank predictions.

The application of the method to an example 4-gene true network is shown in Figure 2.6.

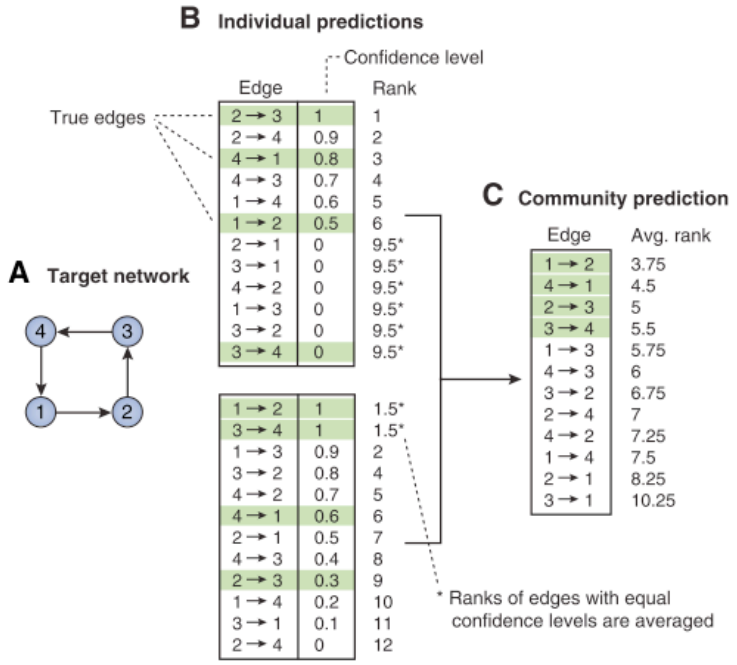


Figure 2.6: An example 4-gene true network (A) used for building a community consensus network (C) from two sets of predictions (B) using the Borda count election method. The true edges from the target network are highlighted in green. This figure was adapted from (Marbach et al. 2010; Marbach et al. 2012).

Here, the edges are ranked from lowest to highest based on the corresponding confidence scores: a high confidence edge is assigned a low rank and a low confidence edge is assigned a high rank. If two or more edges possess the same confidence score, then their average rank is calculated and assigned to each edge. Once the edges are ranked for each predicted network, the ranks associated with each interaction I are averaged across the number of predicted networks (n) resulting in the final score $r_{Borda}(I)$ for I:

$$r_{Borda}(I) = \frac{1}{n}\sum_{j=1}^{n} r_j(I) \qquad (2.20)$$

Here, r_j denotes the rank assigned to I by the jth algorithm. In Figure 2.6, the edges that are highlighted in green signify true interactions. Threshold is applied to the edges that are associated to have Borda scores less than mean of average ranks to obtain final consensus network. For instance, in the example Figure 2.6 (C), threshold is mean of average ranks (i.e. 6.45). This algorithm was programmed in R for comparative analysis.

The Borda count election method has been implemented (Marbach et al. 2012) to build community based consensus networks from microarray data for different model organisms (*E.coli, S.cerevisiae*) (Marbach et al. 2010; Marbach et al. 2012).

2.4.3 Directly merging raw data

Despite there being heterogeneity in the results of independent studies, many meta-analysis approaches attempt to standardise the different quantitative measures generated in each study (i.e. network inference algorithm) to a common scale (Tseng et al. 2012). Although this approach has been used to directly combine multiple microarray experiments from the same, or similar platforms (e.g. multiple Affymetrix platforms or a single Affymetrix U133), it remains a major issue in the field of network inference.

2.5 Discussion and Conclusion

This chapter discusses and reviews various existing network inference algorithms for reconstructing GRNs from gene expression data. These algorithms can predominantly be classed as: 1) Information theory models; 2) Bayesian models; and 3) Differential equation models. Each of these methods attempts to identify biologically plausible GRNs, but it is still a challenge to generate a robust network. In this chapter, we further discussed the various existing qualitative and quantitative techniques used for combining results from independent

studies. Despite the availability of powerful statistical tools, network inference problems can be solved with qualitative approaches that rely on naïve Venn-diagram based methods. Although the Venn-diagram provides a useful way of visualising an intersection or union of predictions, it does not perform an integration of real information, but rather only illustrates the consistencies of features. By contrast, the success of statistical meta-analysis approaches in many quantitative disciplines for combining predictions from independent studies under the same null hypothesis motivated us to apply these methods to network inference problems.

Vote counting is useful when combining studies using raw data, or when the *p*-values are not available. In statistical meta-analysis however, vote counting is considered as a last resort when combining studies as transforming quantitative measures into ranks, a significant loss of information is likely to occur. Combining effect size is well defined when the outcome variables of the independent studies are binary (Tseng et al. 2012). However, in the field of network inference, different numerical values are attached to a network edge by different inference algorithms. Furthermore, a major issue with this approach is that all effect sizes should be in a common metric in order to justify effect sizes from different independent studies with the same treatment effect (Morris & DeShon 2002). Because some studies control for sources of different types of bias, by introducing moderator test that examines experimental design of the effect size. However, they are likely to over or under estimate the treatment effect. A possible remedy may be to reduce the bias by aggregating the effect size across the studies. However, this aggregation of effect across the studies should be justified empirically.

Combining *p*-values across studies provides an important statistical technique for the effect size free ranking method. In particular, we proposed to apply FCPT to combine *p*-values for two major reasons. Firstly, owing to its simplicity, it has the potential to combine extreme probability values attached to a network edge from each network inference

algorithm. Secondly, *p*-values can be calculated non-parametrically using permutation-based analysis to generate a common metric, which we will discuss in depth in the next Chapter.

Chapter 3

A consensus approach to predict regulatory interactions

Abstract

Exploiting microarray gene expression data to predict regulatory interactions has become a key challenge in recent years, for which many network inference algorithms have been developed. Combining predictions of multiple algorithms qualitatively to produce a consensus network has been previously implemented. In this chapter, we propose and investigate a quantitative consensus approach, based on combining regulatory interactions using the Fisher's combined probability test (FCPT). Edge significance values of different network inference algorithms were combined statistically to determine whether the edges should be included in a resulting consensus network. We validated and tested our approach with a variety of benchmark networks, including datasets from the DREAM4 challenge. We have evaluated our algorithm against static & dynamic Bayesian networks, individual networking methods and other popular existing consensus methods - the Borda count election method (BCEM) and the inverse-variance weighted method (IVWM). The results demonstrate that in many cases, consensus networks outperform individual and other consensus methods in predicting regulatory interactions and are more robust. A part of this chapter was published in conference proceedings (Mohammed et al. 2014).

3.1 Introduction

Genetic and physical interactions are essential to regulate cellular machinery. Identifying these interactions and the mechanisms by which they work is an important target of systems biology research (Hecker et al. 2009). A genetic interaction represents a functional/molecular relationship between two molecules, whilst a physical interaction represents the interaction between a molecule and its products. Because the majority of such interactions are temporal in two aspects - stress response and time dependence - molecular activity such as gene expression or metabolite abundance is important and unique in identifying temporal interactions. This becomes particularly important if a large-scale interaction network requires investigation. A whole-cell or genome-wide molecular activity measurement is of great benefit and the advent of high throughput technology such as microarrays has enabled the global responses of a cell to specific perturbations to be measured. Specifically, microarrays provide snapshots of thousands of gene expression profiles under various experimental conditions. Thus, using microarrays has provided data for developing algorithms to construct gene regulatory networks (GRNs). Technically, the construction of a GRN has two aims: locally, to determine how one gene's activity affects other genes' activity, and globally, to determine how genes collectively respond to a perturbation. For example, the inferred interactions can be transcription factor - target gene including gene-gene, protein-gene and protein-protein interactions (Hecker et al. 2009). A predicted GRN facilitates the understanding of the underlying mechanisms of the cellular complex, leading to the ultimate goal of systems biology research - disease control and prevention - as well as targeted drug development (Cassman et al. 2007).

An abundance of network inference algorithms have been developed in the last decade and used to construct GRNs using microarray gene expression data (Penfold & Wild 2011). Those using Bayesian learning mechanisms (referred to as Bayesian networks) have

the potential to encapsulate various types of relationships between gene-pairs, such as the direction of the relationship and the type of interactions (activation or inhibition) (Friedman et al. 2000; Needham et al. 2007). However, Bayesian networks were originally designed to infer directed acyclic graphs. Dynamic Bayesian network inference algorithms were subsequently developed to construct networks using time dependent gene expression data and are able to also reveal cyclic interactions between genes (Zou & Conzen 2005). Although Bayesian networks have many advantages, they work on a small number of genes which demand large-scale experimental data and require prior knowledge of the network of interest (Beal et al. 2005; Gevaert et al. 2007; Steele et al. 2009). Therefore, Bayesian network algorithms are normally computationally expensive. Correlation and mutual information based algorithms are able to capture linear and non-linear relationships between random gene variables respectively. Furthermore, their low computational costs make these frequency-based statistical algorithms more effective when investigating large-scale regulatory networks with fewer experimental samples (Langfelder & Horvath 2008; Meyer et al. 2007; Sales & Romualdi 2011; Faith et al. 2007). We are therefore motivated to continue the consideration of correlation and mutual information based statistical network construction algorithms in this thesis.

An interaction which was not predicted by all individual network inference algorithms using single expression data may be an important one and should not be missed. The qualitative and quantitative consensus approaches are able to make the decision as to whether an interaction (edge between two genes) is valid for delivering a final network. There have been several studies that have addressed this topic with studies by Steel *et al* (Steele & Tucker 2008) and Marbach *et al* (Marbach et al. 2012) demonstrating that combining predicted outcomes from different inference methods can improve the breadth and accuracy of network construction. Steele *et al* (Steele & Tucker 2008) focused on resource diversity when studying the sub-

network of yeast and *E.coli*, and built up an ensemble based on the predictions obtained from different datasets using the same network inference algorithm - the Bayesian network (Steele & Tucker 2008). Both conservative (qualitative) and profitable (quantitative) approaches were employed in the study. The qualitative approach combined common predicted interactions generated from all datasets used, whilst the quantitative approach bootstrapped predictions using the inverse-variance weighted method (IVWM) (DerSimonian & Laird 1986). It was found that such an approach is biased in favour of finding specific local solutions according to the interest of the researcher in the whole network (De Smet & Marchal 2010). Marbach *et al* (Marbach et al. 2012) focused on species diversity and reconstructed GRNs using community-based consensus networks on different model organisms (*E. coli*, *S. cerevisiae*) built using microarray data. They employed the Borda count election method (BCEM) to integrate results from predicted networks from different species. A major drawback of BCEM is that it does not satisfy the majority rule (Erdmann 2011). For example, if an edge is most preferred (i.e. top ranked) by a majority of network inference methods, it is not necessarily the case that this edge is considered significant in the final consensus edge list. In addition, BCEM is vulnerable to teaming. That is, when more candidate edges have similar confidence measures, then the probability of one of these edges being significant increases. Furthermore, transforming quantitative edge scores to average ranks can perhaps make the edge measure less credible as an important feature of unique edge score is lost.

Generating a robust consensus network based on predictions from different network inference algorithms for one species or one experiment is the most asked question in laboratories. We start this work by considering the Fisher combined probability test (FCPT) as a means for addressing algorithm diversity in order to remove the confusion generated by inconsistent edge interactions (Fisher 1932). However, two issues arise. Firstly, it is almost

certain that the network construction has a multiple hypothesis-testing problem where the null hypotheses are: these interactions are insignificant. Therefore, a single Fisher combined probability test cannot answer this question. As a result, we considered false discovery rate (FDR) control (Dabney A, Storey JD 2013). The next important issue is the consistency of meta-data (*p*-values) for applying the FCPT. Not every frequency-based statistical network inference algorithm delivers *p*-values. We have therefore developed a permutation approach for converting frequency-statistics to *p*-values and from this the FCPT can be used for testing the null hypothesis that an interaction is insignificant.

We have used five popular frequency-based network inference algorithms in order to construct our consensus networks. These algorithms include RedeR (Castro et al. 2012); Weighted correlation network analysis (WGCNA) (Langfelder & Horvath 2008); Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE) (Margolin et al. 2006); Context Likelihood of Relatedness (CLR) (Faith et al. 2007); and Maximum Relevance Minimal Redundancy Networks Backward (MRNETB) (Meyer et al. 2007; Meyer et al. 2010). We validated our approach using a variety of benchmark networks for which the ground truth (gold standard network) is known, including the DREAM4 challenge datasets. Our results from the benchmark datasets provided an insight into the variation in the performance measures for each network inference algorithm. We demonstrate that our FCPT-based consensus approach outperforms many individual methods and the existing consensus approaches on several benchmark gene expression datasets. In particular, for large-scale (500 genes) SynTReN derived datasets consisting of small (10 samples) experimental samples, and DREAM4 challenge steady-state based small (10 genes) and medium (100 genes) sized datasets.

3.2 Methods and Material

3.2.1 Benchmark algorithms

In this thesis, we employed five benchmark algorithms[1] for reconstruction of consensus network.

3.2.2 Generating a consensus network

A consensus network inference approach was proposed for building a GRN, by combining results generated from a variety of inference algorithms. Unlike the naïve qualitative consensus approach of selecting a set of common predicted edges from the network algorithms used, we employed a new quantitative scoring system for evaluating and selecting edges utilizing the correlation based (RedeR, WGCNA) and mutual information based (ARACNE, CLR and MRNETB) network inference algorithms.

3.2.2.1 Edge selection strategy

The edge selection strategy involves taking into account the significance values (*p*-values) of the entire connectivity matrix (edges) from each network algorithm, without applying a particular threshold for identifying regulatory interactions. Applying a particular threshold leads to a trade-off between false positive rate (FPR) and true positive rate (TPR). We therefore applied the classic Fisher's combined probability test (FCPT) to score all candidate edges as described in equation (1.1).

[1] These benchmark network inference algorithms are described in more detail in Chapter 2.

3.2.3 False Discovery Rate control

In order to control the rate of false positives, the single hypothesis testing strategy from FCPT was then further enhanced using a multiple hypothesis testing strategy - the False Discovery Rate (FDR) control (Dabney A, Storey JD 2013) for edge prediction. We computed *q*-values from combined *p*-value generated using FCPT. The *q*-value is an FDR analogue of a *p*-value which provides a measure of the proportion of incurred false positives in a particular test. Accordingly, for correcting multiple testing errors, the qvalue package (Dabney A, Storey JD 2013) was implemented to compute *q*-values for FDR level equal to 5% by bootstrapping the combined *p*-values generated by FCPT using default parameters. A significance cut-off of $q<0.05$ was applied to determine statistically significant edges.

3.2.4 Network Validation

In order to validate our consensus approach and compare it with other networking methods, we used a variety of benchmark *in silico* datasets of different dimensions which are illustrated in Table 3.1. These artificially generated (simulated/synthetic) datasets approximate real gene expression data, making it possible to quantitatively evaluate different network inference algorithms on the basis of their prediction accuracy. The flow process of this *in silico* validation framework is shown in Figure 3.1. The *in silico* networks (true networks) and associated *in silico* expression datasets were generated from SynTReN (Van den Bulcke et al. 2006) and GeneNetWeaver (GNW) (Schaffter et al. 2011). GNW was employed to generate DREAM4 challenge datasets.

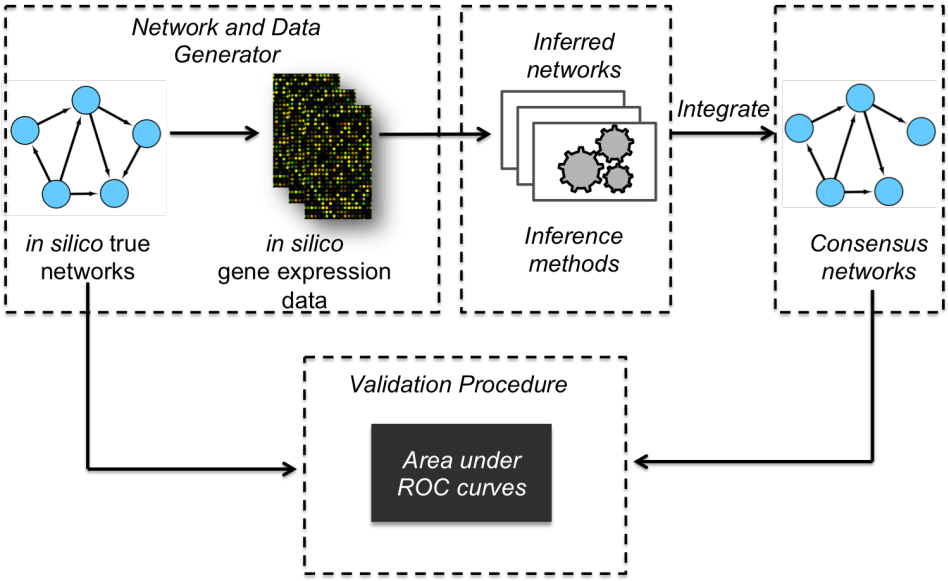


Figure 3.1: Flow process of the validation framework using benchmarked *in silico* datasets to assess the performance of consensus and other network inference methods.

3.2.4.1 *In silico* dataset generation

In this section we describe the details of *in silico* dataset generation from SynTReN and DREAM4 challenge that were used in this study.

3.2.4.1.1 SynTReN

Synthetic Transcriptional Regulatory Network (SynTReN) (Van den Bulcke et al. 2006) is a Java-based tool for generating synthetic TRNs that subsequently produces *in silico* gene expression datasets that approximating biological reality. Datasets are generated using the following steps. First, by selecting a sub-network from the known source networks of *S. cerevisiae or E. coli,* we obtain a network topology. Second, transition functions and corresponding parameters are assigned to the edges of the selected sub-network, specifying the regulatory interactions between genes. Finally, using Michaelis-Menten and Hill equation

kinetics, gene expression levels are simulated for various conditions. A normalized dataset of synthetic microarray measurements is returned, following the optional addition of experimental noise. SynTReN was implemented to generate different simulated steady state datasets under the neighbour addition random sampling method, in order to create subnetworks using default parameters (this included 10% experimental noise, approximated with a lognormal distribution). The algorithm was applied for different network sizes and sample sizes, as summarized in Table 3.1. Auto-regulatory interactions (i.e. self-loops) were removed during the subnetwork extraction process for simplicity.

Table 3.1: Descriptions of the benchmark *in silico* networks and corresponding datasets used in the validation framework. The following acronyms are used: KO-Knockout; KD-Knockdown; MF-Multifactorial; TS-Time series. *Size* represents the number of network nodes (genes) and *Samples* denotes the number of perturbation experiments. A denotes steady state data and B denotes for time series data.

Source	Topology	Type	Size	Samples
SynTReN	*S. cerevisiae*	A	100	10,100,500
SynTReN	*S. cerevisiae*	A	500	10,100,500
DREAM4-KD	*E. coli/S. cerevisiae*	A	10,100	100
DREAM4-KO	*E. coli/S. cerevisiae*	A	10,100	100
DREAM4-MF	*E. coli/S. cerevisiae*	A	10,100	100
DREAM4-TS	*E. coli/S. cerevisiae*	B	10	105
DREAM4-TS	*E. coli/S. cerevisiae*	B	100	210

3.2.4.1.2 DREAM4 Challenge

We used *in silico* gene expression data generated for the DREAM4 (Dialogue on Reverse Engineering Assessment and Method) challenge to extend our validation process, as these

have been used previously to assess more than 30 network inference algorithms (Marbach et al. 2010; Greenfield et al. 2010). The aim of the DREAM4 challenge is to reverse engineer gene regulatory networks using different sizes and varieties of simulated steady state and time series datasets (see Table 3.1). DREAM4 provides three *in silico* sub-challenges, described below:

1. Insilico_size10 - Insilico_size10 networks, which consist of five 10-gene gold standard networks, and five corresponding expression datasets for each experiment (KD-Knockdowns, KO-Knockouts, MF-Multifactorial, and TS-Time series). We describe the different experiment types in the next section. Each individual experiment comprises a single simulation of a fixed (gold standard) network. The five networks in each experiment vary in their topology. This means there are 25 simulated experiments in total.
2. Insilico_size100 - Similar to size 10, Insilico_size100 consist of five 100-gene networks and five corresponding datasets from different simulated experiments (Knockouts, Knockdowns and Time series) although Multifactorial perturbations were not included.
3. Insilico_size100_Multifactorial - Similar to size 100, Insilico_size100_Multifactorial consists of five 100-gene networks, and five corresponding datasets generated exclusively from Multifactorial perturbation experiments.

The different experiment types were simulated as follows:

- A Knockout experiment provides steady state expressions levels under single gene deletions for each of the genes in the network. Setting the transcription rate of each gene to zero in turn simulates this experiment.

- A Knockdown experiment provides steady state expressions levels under single gene knockdowns for every gene in the network. This is simulated by halving the transcription rate of each gene in turn.
- A Multifactorial experiment provides steady state expressions levels following random multifactorial perturbations to the genes in the network, simulated by increasing or decreasing the basal activation rates concomitantly.
- A Time series experiment provides a time course that simulates the response of the network to an initial perturbation at time t=0, followed by removal of the perturbation at time t=500, where the gene expression levels revert back to the original state (i.e. unperturbed state). The perturbation is applied to a third of the genes in the network, whose basal transcription rates are strongly increased or decreased accordingly.

Subnetworks were extracted from TRNs of *E.coli* or *S. cerevisiae* which possessing no self-loops, with all datasets corresponding to mRNA levels. All the DREAM4 *in silico* datasets were downloaded from DREAM project website (http://dreamchallenges.org/) that were originally generated from *GeneNetWeaver*[2] (Schaffter et al. 2011) using stochastic differential equations, which included experimental noise simulated as a mix of normal and lognormal random variables.

3.2.5 Scoring Method

A binary decision problem is well established for evaluating gene networks as it classifies a predicted edge between a pair of nodes as either existing or not existing. Thus, a positive or negative label is assigned if the edge exists or doesn't exist respectively for each respective pair of nodes. A positive label assigned by a network inference algorithm to a predicted gene

[2] *GeneNetWeaver* (GNW) is open-source Java-based software for generating of *in silico* benchmark networks. GNW was widely employed in DREAM challenges to test the accuracy of network inference methods.

interaction (edge) can be a True Positive (TP) if the corresponding interaction is present in the true network (gold standard), or else a False Positive (FP) if that corresponding interaction is absent. Similarly, a negative label assigned by a network inference algorithm can be a True Negative (TN) if the corresponding interaction is absent in the true network, or a False Negative (FN) if the corresponding interaction is present. A confusion matrix is used to summarise and classify the predicted positive and negative labels as shown in Table 3.2.

Table 3.2: Summary of the confusion matrix used to classifiy edge predictions. TP (True Positive) - an edge present in both the predicted and true network. FP (False Positive) - an edge present in the predicted network but not the true network. TN (True Negative) - an edge absent from both the predicted network and the true network. FN (False Negative) - an edge absent in the predicted network but present in the true network.

Edge	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

To measure the performance of the consensus approaches and compare with that of other individual networking methods, we employed classic Receiver Operator Characteristic (ROC) curves. This is common practice and generally recommended for the evaluation of binary decision problems (Science et al. 2011). The advantage of using ROC curves is that they evaluate the performance of classification methods without choosing any particular discrimination threshold, which means there is no bias present (Fawcett 2006). ROC curves are viewed graphically by plotting FPR (False Positive Rate) on the X-axis against TPR (True Positive Rate) on the Y-axis for each choice of threshold value, A key property of ROC

curves is that they measures the robustness of a classifier; that is, its sensitivity to signal and insensitivity to noise.

The *True Positive Rate*, or *Sensitivity* (also known as *recall*), measures the fraction of true interactions, which are correctly identified:

$$\text{True Positive Rate} = \frac{\text{TP}}{\text{TP} + \text{FN}} = Sensitivity \tag{3.1}$$

The *False Positive Rate* measures the fraction of interactions which are incorrectly identified:

$$\text{False Positive Rate} = \frac{\text{FP}}{\text{FP} + \text{TN}} = 1 - Specificity \tag{3.2}$$

where

$$Specificity = \frac{\text{TN}}{\text{TN+FP}} \tag{3.3}$$

In ROC space, a perfect network prediction (i.e. identical to the true network) will have TPR=1 and FPR=0 (i.e. the prediction will sit in the top-left corner). In our *in silico* experiments, for each network inference algorithm, we plot ROC curves where each point in the plot corresponds to different level of statistical significance (threshold).

A global numerical measure, the Area Under ROC Curve (AUROC) is more commonly utilised in decision-making problems for evaluating the performance measures of individual network inference methods. The AUROC ranges between 0 and 1. The overall performance of a networking method is considered better if its AUROC is closer to 1 and further away from 0.5, as an AUROC of 0.5 is equivalent to random guessing, indicated by a diagonal in the ROC curve between the coordinates (0,0) and (1,1).

3.2.6 Estimation of significance values

In order to combine multiple network inference algorithms using Fishers Combined Probability Test (FCPT), it is imperative that each of the network inference algorithm produces an edge score with a common metric, namely significance *p*-values. However, many network inference algorithms do not deliver *p*-values, but rather provide unique statistical edge scores. In this section, we discuss a new algorithm we have developed for transforming these unique statistical scores into *p*-values, using a non-parametric random sampling approach via permutation analysis.

3.2.6.1 Parametric approach to calculate *p*-values

Parametric statistical tests are those that assume that the data is normally distributed, and depend on statistical measures such as means, standard deviations and variances (Harris et al. 2008). Here, the *p*-values from correlation coefficients (CC) generated from RedeR and WGCNA were calculated using the R programming function *cor.prob* (Venables 2000). As an example, to illustrate the distribution of correlation coefficients, we applied RedeR and WGCNA to size-100 simulated gene expression data generated from SynTReN. Histograms of the corresponding CC distributions are shown in Figure 3.2A (RedeR) and Figure 3.2B (WGCNA). It should be noted that the distribution of CCs cannot be exactly Gaussian as it varies between -1 and 1. A normal (Gaussian) curve calculated based on the mean and standard deviation of each CC distribution is shown by a solid red line superimposed over the histogram in each case. The plots reveal that the CC distributions are approximately normal. However, many CCs that are almost exactly -1 or 1 are grossly underestimated by the Gaussian approximation.

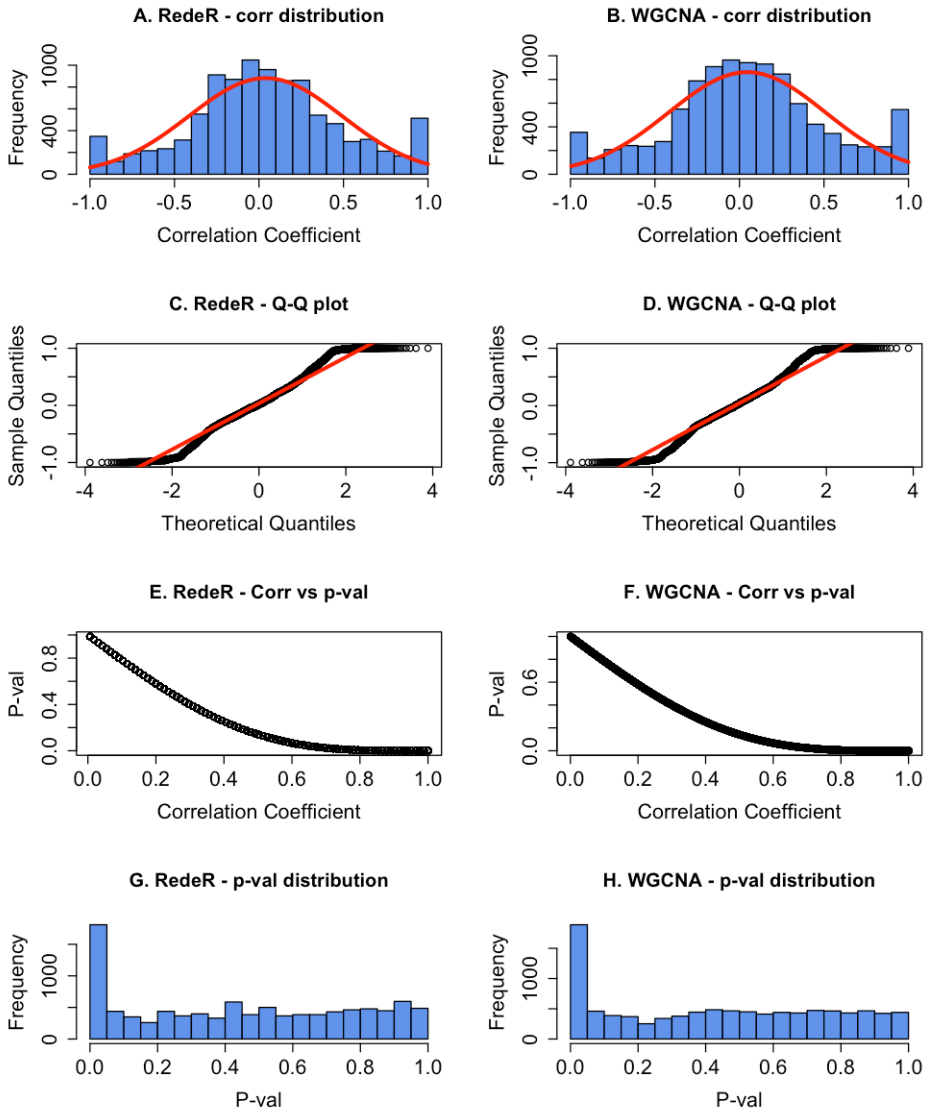


Figure 3.2: The distribution of frequency statistics calculated using a parametric approach. A, B: Correlation coefficient histograms obtained by applying RedeR and WGCNA to benchmark *in silico* data for size 100 genes. The red line indicates a normal curve fitted to the histogram. C, D: Quantile-Quantile (Q-Q) plots for the distribution of correlation coefficients. The linearity of the quantile points across the diagonal red line suggests that the histogram approximates a normal distribution in each case. E, F: Scatter diagrams displaying the correlation between correlation coefficient and calculated *p*-values. G, H: Distributions of transformed *p*-values.

We further tested for normal distributions using Q-Q (Quantile-Quantile) plots as shown in Figure 3.2C (RedeR) and Figure 3.2D (WGCNA). Q-Q plots are used to compare sample quantiles against theoretical quantiles. More specifically, quantiles generated from CC are compared against theoretically calculated quantiles from normally distributed data. The red fitted line along the diagonal of the quantile points suggest that the CC approximate a normal distribution for both RedeR and WGCNA. *p*-values were calculated from the CC using the R function *cor.prob*, with Figures 3.2E and 3.2F, showing the corresponding relationship between CC and *p*-values for RedeR and WGCNA respectively. The distribution of calculated *p*-values for RedeR and WGCNA are shown in Figures 3.2G and 3.2H respectively, and approximate a uniform distribution in each case. The flat uniform distribution of *p*-values ranging between 0 and 1 attribute to null distribution (Bland 2013), where the null distribution is the probability estimates of a test statistic when the null hypothesis (H_0) is true. The spike at low *p*-values (i.e. close to 0) is associated with alternative hypothesis being true which includes false positives. If the *p*-value is less than or equal to 0.05 (5% significance level), then null hypothesis can be rejected.

3.2.6.2 Non-parametric approach to calculating *p*-values

When the normality assumption is violated i.e. the data does not approximate a normal distribution - then parametric tests may not be meaningful or useful. In this case, non-parametric statistical tests offer an alternative solution for significance testing, as they make no prior assumption of the normal distribution of the statistical data.

Permutation tests are non-parametric based multiple comparison statistical tests which provide a formal way of quantifying the statistical significance of a test by delivering *p*-values (Knijnenburg et al. 2009). A permutation test, which can also be called a randomization test, calculates test statistics by randomly re-arranging (i.e. permuting) the

labels of a dataset. Here, labels refer to experimental effect or condition (samples). Statistical significance is assessed by comparing the test statistic derived from the permuted values against the original (unpermuted) values under the null hypothesis. If the null hypothesis is true, the permuted labeled data would reflect original data under any condition suggesting there is no experimental effect. This means, the permuted test statistic would look like the original test statistic (Nichols & Holmes 2002). The, *p*-values are calculated by estimating the proportion of values generated by permuting the labels that are greater than, or equal to, the values calculated from the original (unpermuted) data. Significantly, if low *p*-values are attained, this indicates that the labels (samples) are not random and that the configuration of the original label is relevant to the data.

Here, ARACNE, CLR and MRNETB do not provide *p*-values for edges. Instead, they deliver mutual information values as relational values, which are not consistent across algorithms. In order to derive a significance value for each edge between two genes, we proposed the following null hypothesis:

H_0: the relation between two nodes is by chance.

If the null hypothesis cannot be rejected, an edge between two genes cannot be considered significant. Otherwise, a pair of genes is believed to have a strong correlation. We drew samples by permuting the gene expression data across each gene N times across experimental labels with replacement as shown in the Figure 3.3(A). In this diagram, E_0 represents an example original gene expression dataset comprising 4 genes and 4 experimental samples. ($E_1,..,E_N$) indicate expression datasets obtained by randomly permuting E_0 a total of N times (N=1000 in this study). Each network inference algorithm was run using the original and sampled gene expression as input, as shown in Figure 3.3B.

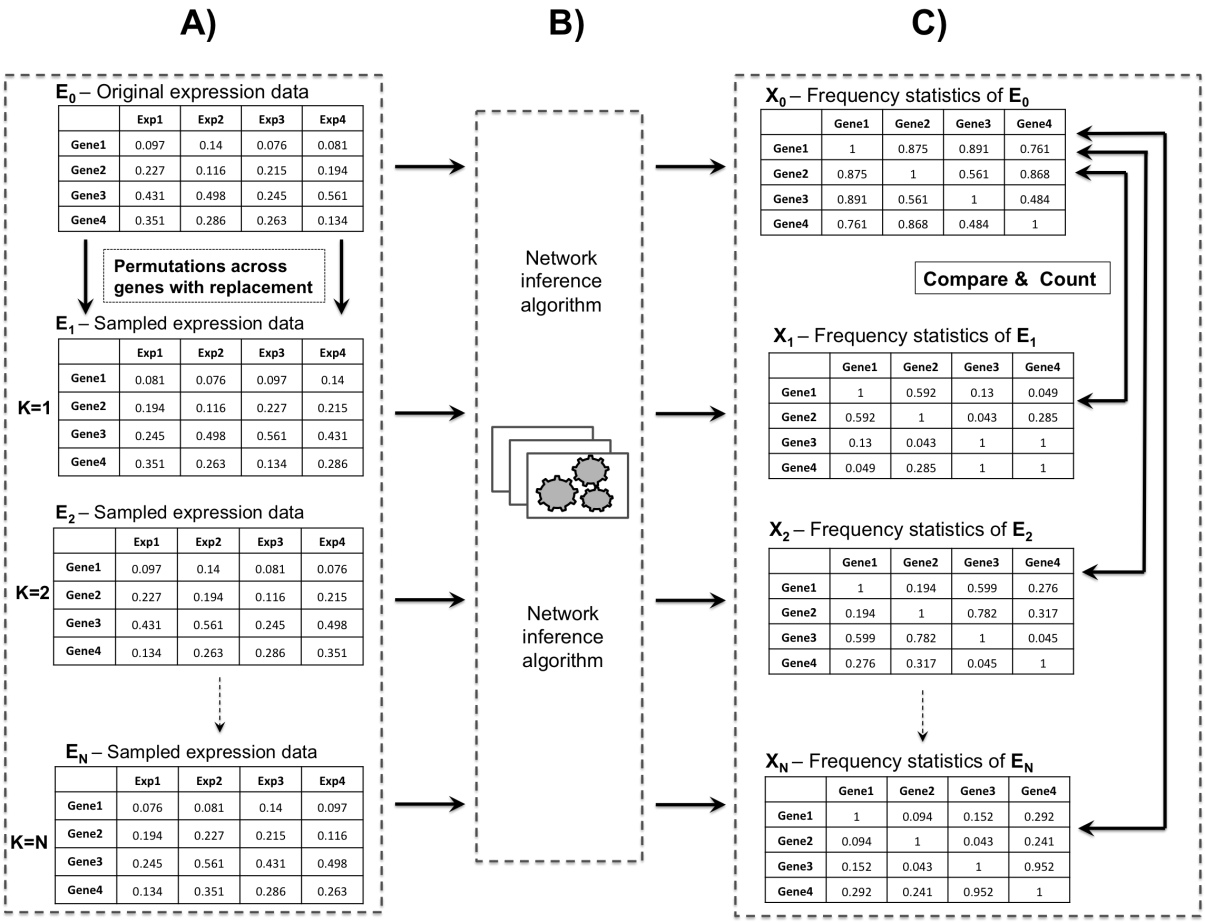


Figure 3.3: Workflow illustrating the process of calculating *p*-values from an example gene expression matrix consisting of 4 genes and 4 experimental samples. A: Original (E_0) and random sampling of gene expression data across genes (E_K) by permutation analysis. B: Network inference algorithms that takes the original and sampled expression data as input and output frequency statistical scores. C: Frequency statistics matrices from the original $X_0[i,j]$ and sampled gene expression data $X_k[i,j]$.

Thus $P[i,j]$ (*p*-values) for corresponding mutual information weights are calculated by comparing and counting (Figure 3.3C) the number of times that a particular edge weight under sampled data $X_k[i,j]$ is greater than, or equal to, the original edge weight $x_0[i,j]$, then dividing the count by the sampling number N, as shown in the equation below (Knijnenburg et al. 2009):

$$P[i,j] = \frac{\sum_{k=1}^{N} I(X_k[i,j] \geq X_0[i,j])}{N} \quad (3.4)$$

Here, $I(.)$ is the indicator function. When a $P[i,j]$ value is smaller than a critical p-value, the null hypothesis is rejected. The pseudocode for the permutation-based algorithm to derive p-values is indicated in Figure 3.4.

Significance estimate
Input: *Network with frequency statistics attached to each network edge, $X_0[i,j]$ with gene expression data*
Output: *Network with statistical significance estimates (p-values) attached to each network edge [i,j]*

N=number of random samplings by permutation with replacement
For *each pair of nodes [i,j]* **do**
count=0
Predict frequency statistics with randomly sampled gene expression data N times – $X_k[i,j]$
If *the edge weight, $X_k[i,j]$ with permuted data ≥ edge weight $X_0[i,j]$ with original data* **then**
count=count+1
end
P.value[i,j]=count/N
end

Figure 3.4: Illustrates a new permutation-based algorithm to estimate significance values from frequency statistics generated by a network inference algorithm.

3.2.6.2.1 Validation of the non-parametric approach

The proposed random sampling approach by means of permutation analysis was investigated for its validity using a benchmark gene expression dataset of size 100. In this investigation, we compared the p-values estimated from our non-parametric permutation approach against an established parametric R function, *cor.prob* using the same simulated dataset used above in Section 3.2.6.1 for parametric-based analysis, and network inference algorithm respectively. The underlying assumption is that if the p-values estimated by non-parametric approaches are similar to those estimated by the parametric approach, then a linear regression fit with a high correlation coefficient of determination (R^2) should be obtained and should

show a statistically significant score (p<0.05) (Zou et al. 2003). Furthermore, the p-values estimated by the non-parametric statistical approach should approximate a uniform distribution between 0 and 1 when the null hypothesis is true (Bland 2013; Hung et al. 1997). While the distribution of p-values is not uniform (skewed) when the alternative hypothesis is true (i.e. p-values are likely to be smaller creating a spike near 0) (Barton et al. 2013). In this investigation, we employed WGCNA as the common network inference algorithm as it outputs correlation coefficients as frequency statistics. This enabled us to apply a parametric approach to estimate p-values, as the distribution of correlation coefficients approximates a normal distribution as shown in Figure 3.5.

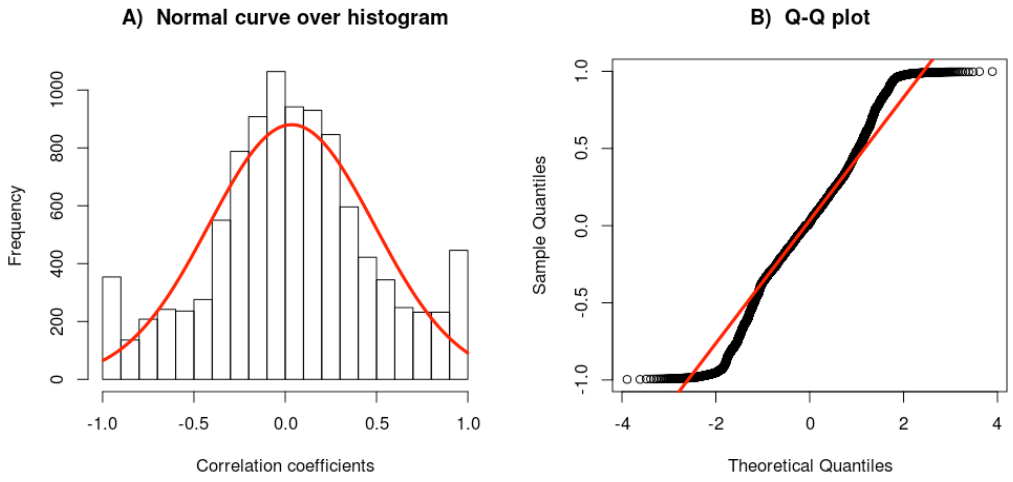


Figure 3.5: A: Histogram showing the distribution of correlation coefficients calculated by WGCNA as edge scores across all the genes of size 100. The red line indicates a normal distribution. B: Quantile-Quantile (Q-Q) plot for the corresponding correlation coefficients calculated by WGCNA. The linearity of the quantile points across the diagonal red line suggests that the histogram approximates a normal distribution.

The flow process of this investigation is outlined in Figure 3.6. The estimated p-values from both parametric and non-parametric approaches were investigated for similarity using a simple scatter plot, and the relationship between them was fitted using a linear regression

model as shown in Figure 3.7. The similarity was quantified using the squared correlation coefficient, R^2, and its statistical significance score (p-value).

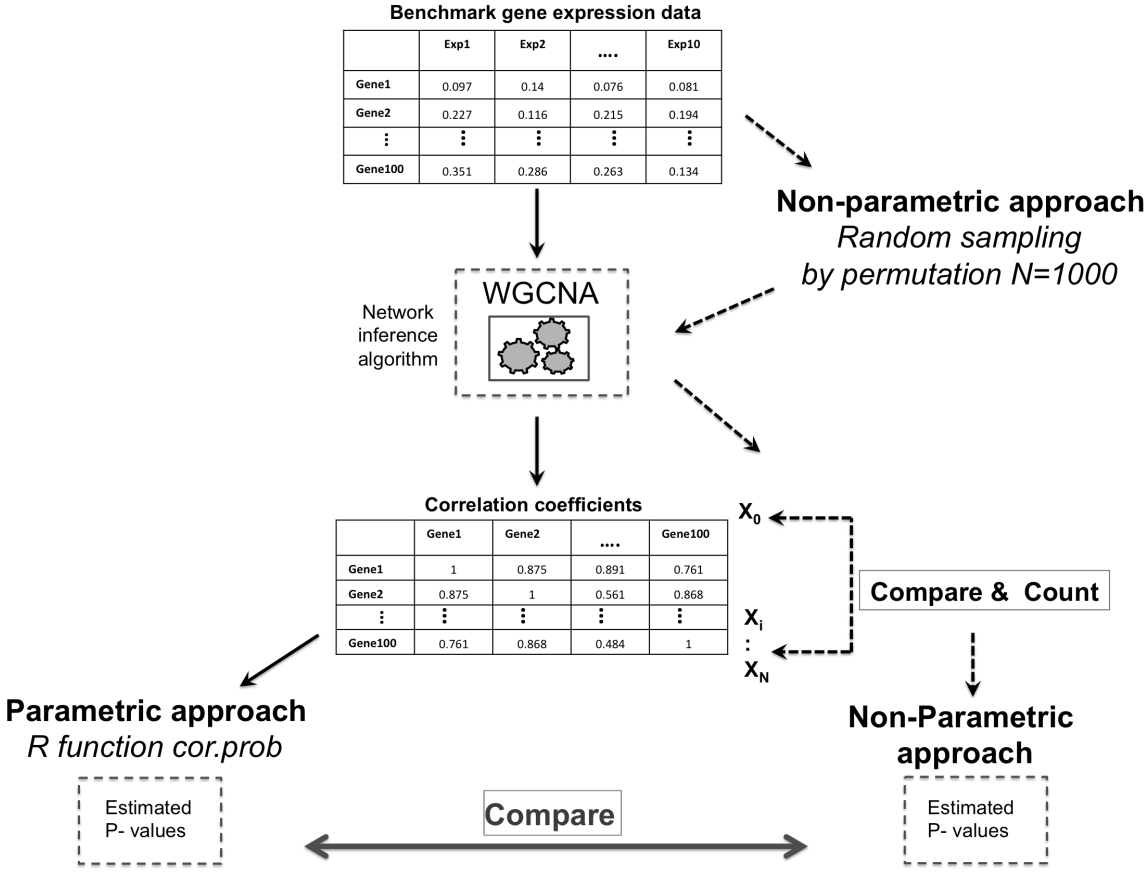


Figure 3.6: Flow process for estimating p-values using parametric and non-parametric approaches from a common benchmark gene expression data of size 100, and a common network inference algorithm. The solid and dashed arrows indicate the parametric and non-parametric workflows respectively.

Figure 3.7B reveals that the p-values estimated from the random sampling approach approximate a uniform distribution between 0 and 1 when the null hypothesis is true, and creating a spike close to 0 under alternative hypothesis (Barton et al. 2013) that matches closely with the p-values estimated using the R function as in 3.7A.

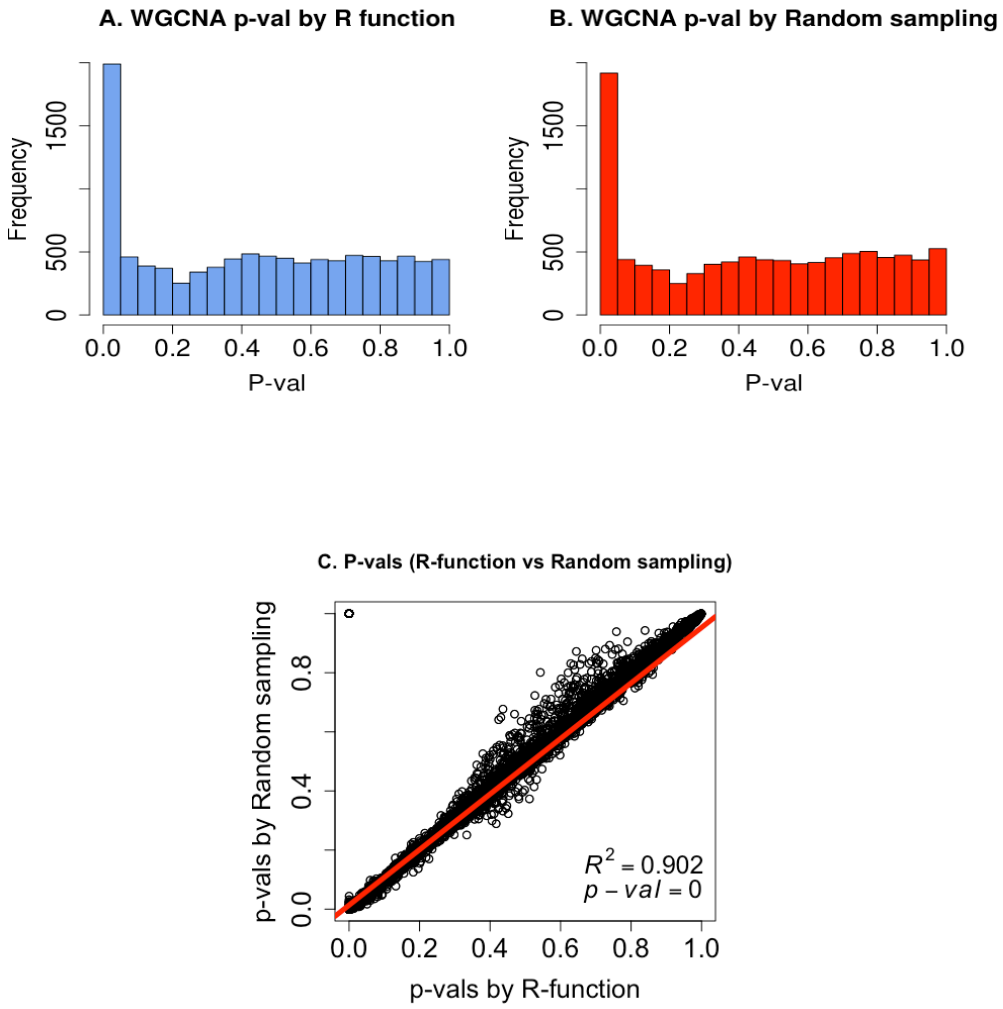


Figure 3.7: Shows the distribution of *p*-values estimated using two different approaches on the same gene expression dataset of size 100 and the same network inference algorithm (WGCNA). A: Distribution of *p*-values obtained parametric based approaches using the parametric R function *cor.prob*. B: Distribution of *p*-values from the non-parametric random sampling approach by permutations. C: Scatter plot showing the relationship between the estimated *p*-values in A and B. The red line indicates a linear regression fit.

The similarity between the *p*-values obtained using the parametric and non-parametric methods was further investigated using the scatter plot shown in Figure 3.7C and quantified by fitting a linear regression model. The adjusted squared correlation coefficient value of R^2= 0.902 and corresponding statistical significance value p<2.2e-16 suggest that a high

correlation between the two approaches exists. This case study confirms the validity of our non-parametric, permutations based random sampling approach for calculating *p*-values.

Next, we move on to calculating *p*-values from the mutual information (MI) based algorithms, ARACNE, CLR and MRNETB using our permutations based algorithm. Figures 3.8A, 3.8B and 3.8C compare the MI statistics derived using these inference algorithms for a sample gene expression dataset of size 100 and sample size 100. It is quite evident from the distribution of MI statistics that none of these scores approximate a normal distribution, showing the necessity of employing a non-parametric method. Figures 3.8D, 3.8E and 3.8F, compare scatter plots between the mutual information estimates and calculated *p*-values. It is observed that the relationship between mutual information estimates and calculated *p*-values, are inversely proportional as expected from our previous investigation (Figures 3.2E and 3.2F). The distributions from transformed *p*-values for ARACNE, CLR and MRNETB are shown in Figure 3.8G, 3.8H and 3.8I respectively. It is apparent from this plot, that the *p*-values approximate a uniform distribution in each case when the null hypothesis is true (i.e. *p*-values between 0 and 1). Notably, at the tails, the spike closer to low *p*-value (i.e. near 0) is associated with alternative hypothesis being true (Barton et al. 2013), while the spike at high *p*-value (i.e. near 1) provides stronger evidence that the null hypothesis is true (Rodwell et al. 2004). Further examination revealed that the high bar at *p*-value close to 1 is associated with those edges that have a MI score equal to 0. For example, in this illustration, ARACNE revealed 4168 edges that have MI score equaling 0 (Figure 3.8A), and the same (4168) edges were associated with having *p*-valve exactly equaling 1 (Figure 3.8G). Those edges having low MI score (or high *p*-value) indicate that the associated gene pair in an edge are mutually independent (or randomly associated), and are not biologically regulated (Margolin et al. 2006). This explains the reason for observing a spike at *p*-value close 1 in CLR (Figure 3.8H) and MRNETB (Figure 3.8I).

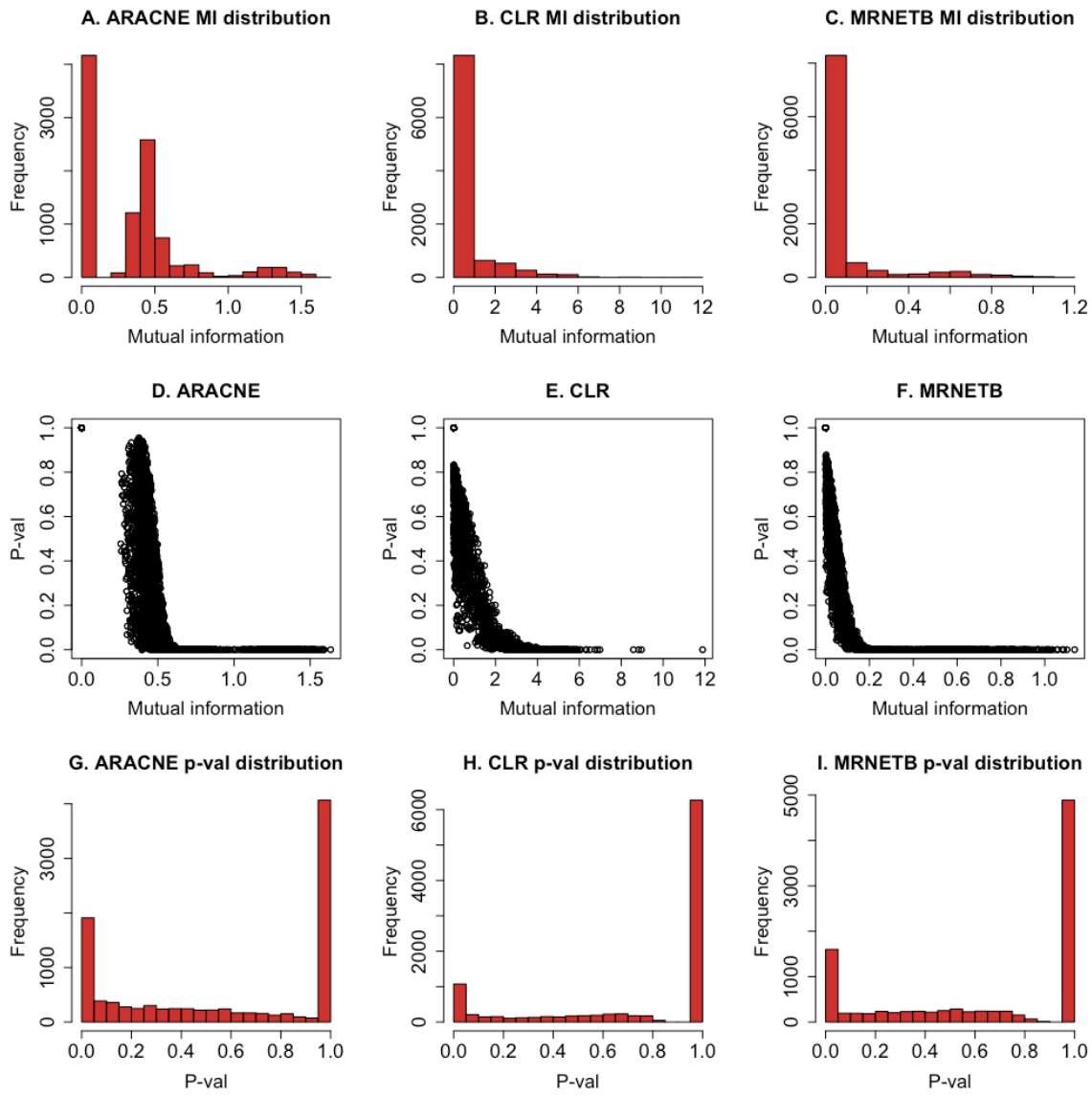


Figure 3.8: The distribution of frequency statistics for MI inference algorithms calculated using the non-parametric permutations based approach. A, B, C: Histograms showing mutual information scores from ARACNE, CLR and MRNETB using benchmarked *in silico* gene expressions datasets (size 100 and sample size 100). D, E, F: Scatter plots between the mutual information estimates and calculated *p*-values for ARACNE, CLR and MRNETB respectively. G, H, I: Histograms showing the distributions of transformed *p*-values for ARACNE, CLR and MRNETB respectively.

3.3 Results

In this section we investigate the performance measures of consensus networks generated by Fishers Combined Probability Test (FCPT) from five prominent information theory based network inference algorithms: RedeR (Castro et al. 2012), WGCNA (Langfelder & Horvath 2008), ARACNE (Margolin et al. 2006); CLR (Faith et al. 2007) and MRNETB (Meyer et al. 2007; Meyer et al. 2010). The first set of analyses focused on validating the consensus network against an *in silico* derived network, commonly known as a gold standard or target network, for which the ground truth is known. Here, we employed different dimensions of benchmarked networks, generated from SynTReN[3] as summarized in Table 3.1.

The gold standard networks from the yeast GRN of gene size 100 and 500 were used in the validation study representing medium and large networks respectively. The simulated gene expression data for both these corresponding gold standard networks of size 100 and size 500 have variable experimental samples sizes of 10, 100 and 500. The networks[4] for size 100 and size 500 are shown in Figure 3.9 drawn using an igraph module (Csardi 2010). In the graph, blue nodes indicate genes and black arrowed edges represent true interactions between genes. The size 100 and size 500 networks consist of 147 and 868 true regulatory interactions respectively.

Figure 3.10 shows the similarity measures between the networks inferred using the different reverse engineering algorithms from the size 100 *in silico* gene expression data. Here, the overlap ratio between consistently identified regulatory interactions was used to quantify the similarity. Overlap ratio ranges between 0 and 1, where 0 and 1 represents no overlap and perfect overlap respectively. For example, in predicted networks A and B, if 20

[3] See the Methods section for more details as to how *in silico* networks are generated from SynTReN.

[4] The *in silico* gene expression data for these gold standard networks are steady state.

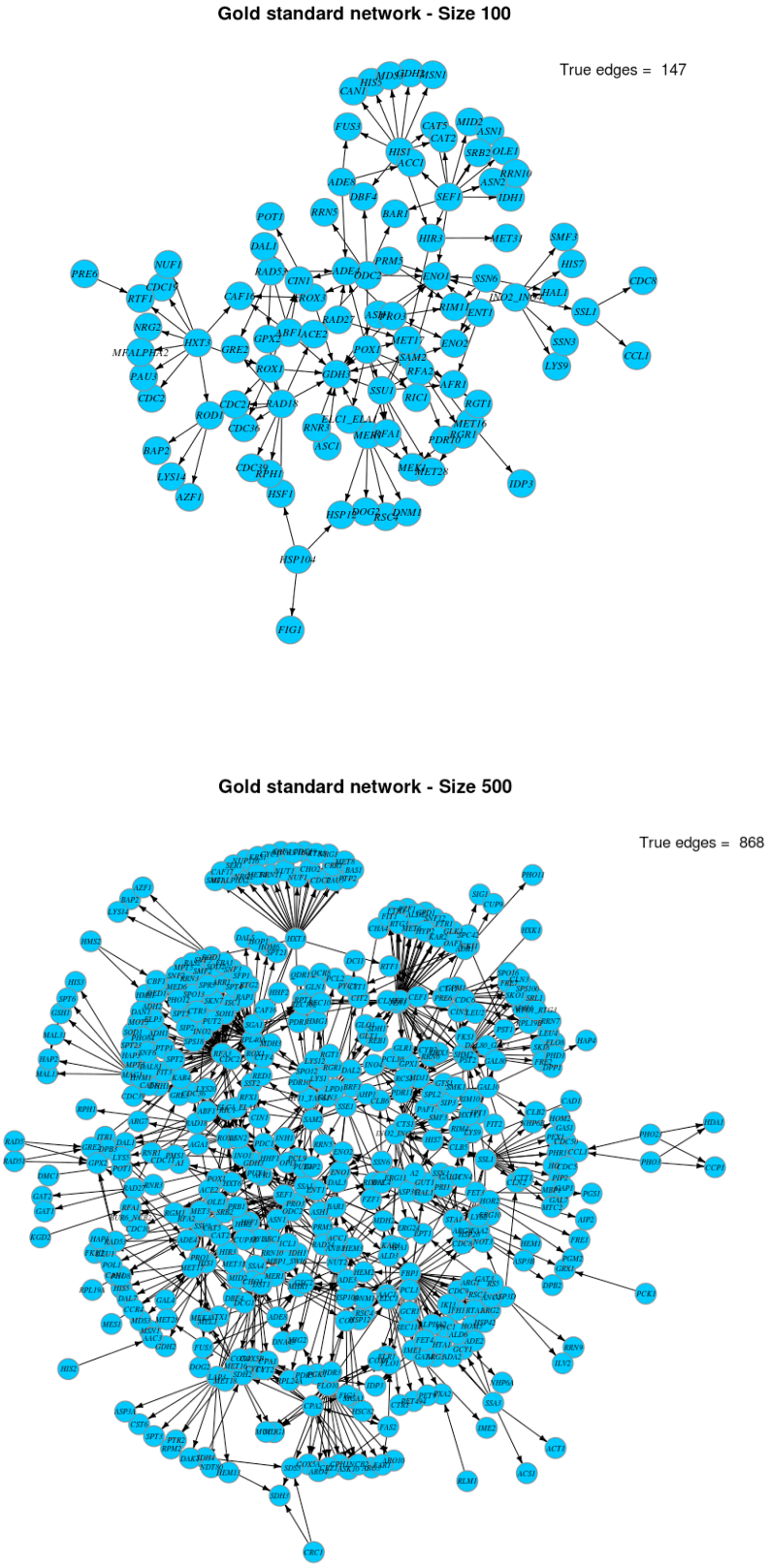


Figure 3.9: Shows gold standard true *in silico* network of size 100 and size 500 generated from SynTReN. Blue nodes denote genes and black arrowed edges represent true interactions.

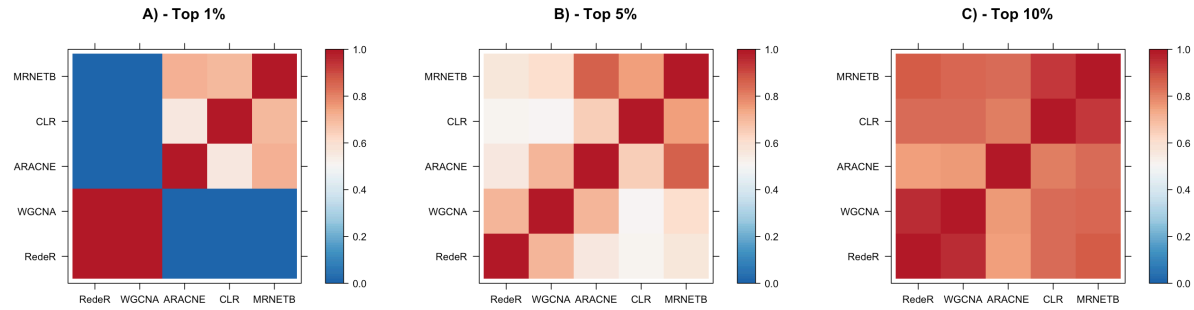


Figure 3.10: Similarity between different network inference algorithms for identifying consistent regulatory interactions with *in silico* gene expression data of size 100 at various rank thresholds, A). Top 1%, B). Top 5%, and C). Top 10%.

interactions were commonly identified in the top 1% (100 edges) ranked interactions yields an overlap ratio of 0.2 (i.e. 20/100). From this data, one can see that at lower rank thresholds (top 1% and top 5%) there is a weak similarity between correlation based (RedeR and WGCNA) and mutual information based (ARACNE, CLR and MRNETB) methods for identifying consistent features, except for when the threshold is relaxed (i.e. top 10%). These findings seem to suggest that every reverse engineering method tends to recover a different set of predictions, and thus demonstrate weak similarity. The inconsistency observed among these different network inference methods raises concerns over which inferred network yields accurate predictions when the target network is unknown. This finding of heterogeneity in the performance of different network inference algorithms is further evidenced in Appendix A.1-A.4, where results for other benchmark datasets are shown. The observed variation in predicted networks provides a clear motivation to use consensus-learning techniques in order to improve the accuracy and robustness of predictions.

Next, we move on to build consensus networks using FCPT. In order to provide a thorough comparison of consensus network performance against that of individual inference

methods, for each benchmark *in silico* dataset, we display ROC curves plotting True Positive Rate (Sensitivity) against False Positive Rate (1-Specificity) at various significance thresholds. The area under the ROC curve (AUROC) is used to quantify the performance of each algorithm. In Figure 3.11, ROC curves and AUROCs are shown for the benchmark yeast GRN datasets of size 100 and size 500. It was observed from the plot that the performance of individual methods varied, with the consensus network (highlighted as a solid black line) demonstrating the best performance overall. However, it is worth mentioning that with size 100 networks, the consensus method is still on occasion beaten by a single method when the number of samples is smaller than or equal to the number of genes (in this case, for sample sizes of 10 and 100). Furthermore, the performance of the consensus method typically improves and matches that of the best single method when the number of samples is sufficiently large (500 samples). A possible explanation for these results may be that an inadequate number of samples yields spurious edges from the single inference methods that influences the performance of the consensus method. However, for larger size networks (500 genes), the consensus method is robust, and consistently beats other single methods, independently of sample size.

It is interesting to note that the performance of each inference method improved for both sized networks with increasing sample size. The grey dashed line running across the diagonal of the ROC plots denotes the network obtained by random guessing.

For comparative analysis, each of the network inference methods applied to produce the consensus network - which are based on correlation and mutual information models - were also compared against static Bayesian networks (BNs) (Scutari 2009) and dynamic Bayesian networks (DBNs)[5] (Morrissey 2013), using steady state and. time series gene expression data respectively. Here, we used area under ROC curves (AUROC) to quantify the

[5] Static and dynamic Bayesian methods are discussed in Chapter 2.

performance measures for each corresponding ROC curves. Figure 3.12 compares the performance of the consensus network against static BNs and other inference methods for benchmark networks of different dimension and for different sample sizes. For size 100, one can see that the consensus network performs better (indicated by the horizontal dashed line) than the majority of individual methods, including static BNs. However, the correlation-based methods outperform all other approaches, providing better predictions, for these medium sized networks. Mutual information and consensus network performance improved with increased sample size. For size 500, the consensus network outperforms all single inference methods, including Bayesian networks, even when the number of samples used is small (10). These results suggest that the consensus network has better capability for predicting true interactions for large networks (size 500), with its performance improving with increased sample size. It is also worth mentioning that mutual information networks performed well compared to correlation networks, for large sized networks.

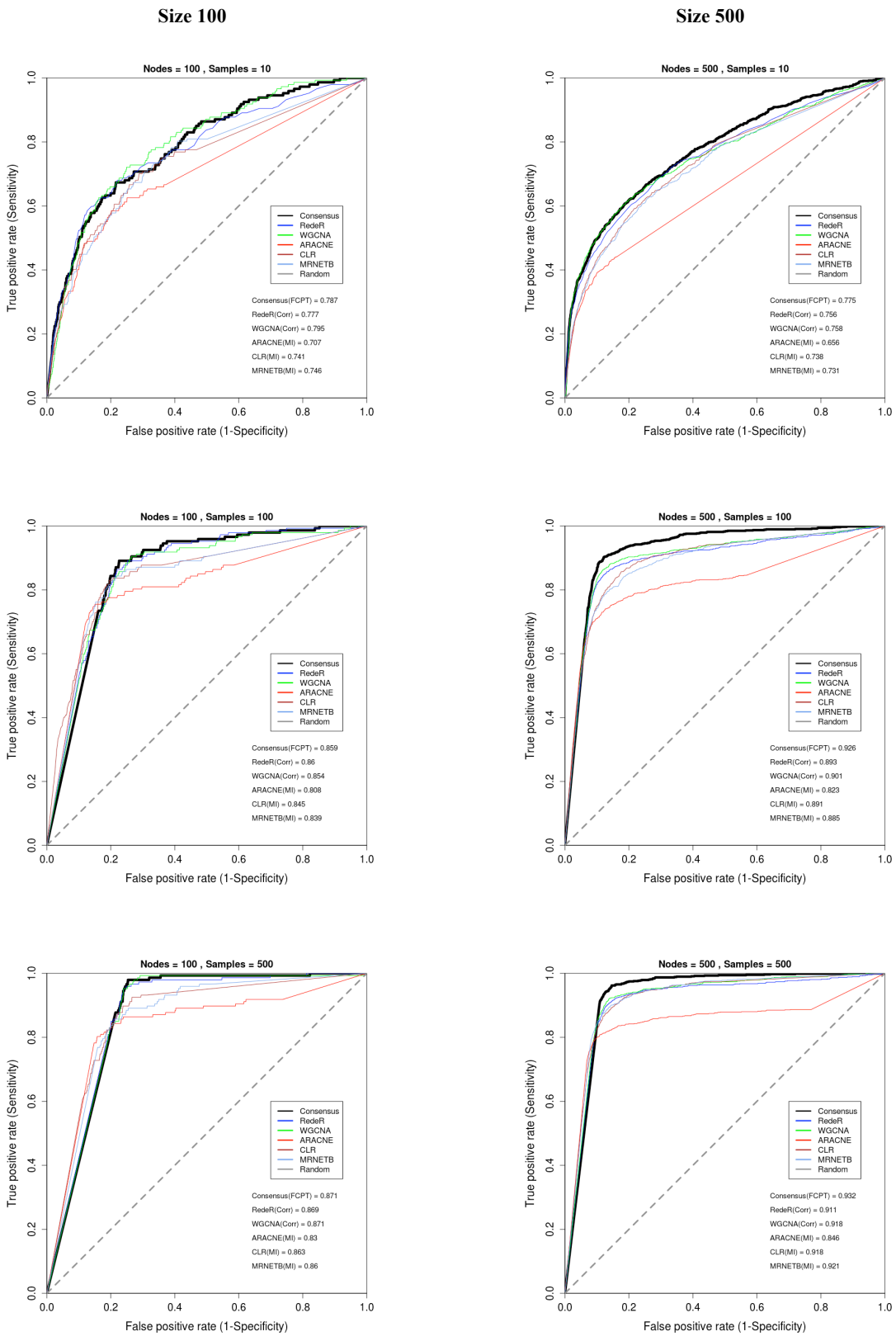


Figure 3.11: ROC curves and corresponding AUROC estimates for individual network inference methods and the FCPT-based consensus method using benchmark *in silico* datasets of size (nodes) 100 and size 500 for varying sample sizes (10, 100 and 500) generated from SynTReN. Abbreviations: Corr-Correlation; MI-Mutual information.

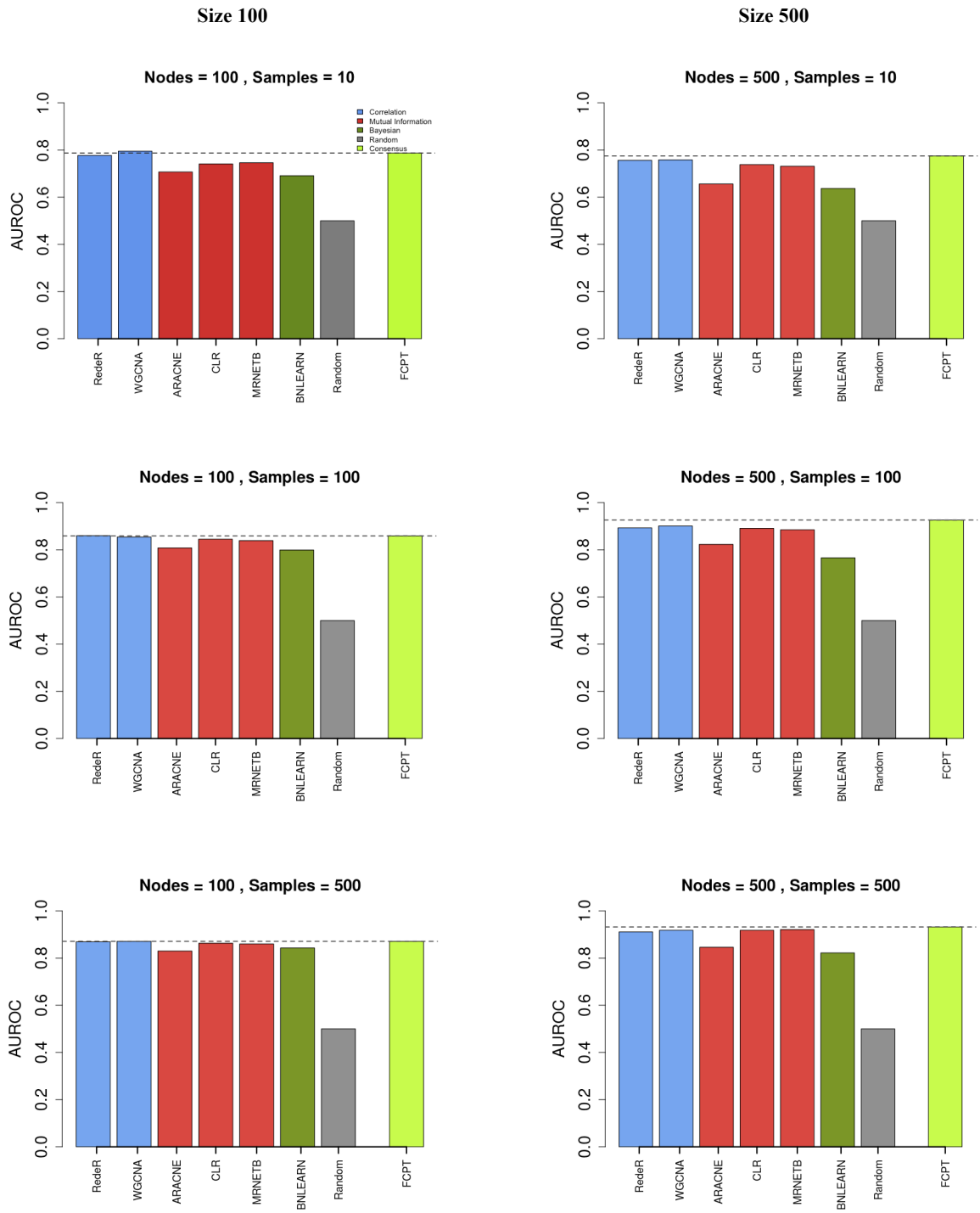


Figure 3.12: Comparing the performances of consensus network by FCPT against individual network inference approaches using AUROC measures with benchmarked *in silico* datasets of size (nodes) 100 and size 500 for varying sample sizes (10,100,500).

3.3.1 DREAM4 size 10

The boxplots in Figure 3.13 show the AUROC scores for different network inference algorithms when applied to a variety of DREAM4 challenge benchmark steady state and time series data generated from small size networks (10 genes). The steady state data used consists of different simulated knockdown (KD), knockout (KO), and Multifactorial (MF) experiments, whereas the time series (TS) data is single simulated experiment. For more details on the simulated experiments, refer to the Method section. We compared our consensus approach against static BNs and DBNs when steady state data and time series data were used respectively. It is observed in the boxplots (Figure 3.13) that the consensus network performs consistently well, outperforming many single network inference approaches in several cases. In particularly, it ranked second under steady state KD simulated data and fourth under TS simulated data. Moreover, with the KO and MF experiments, it outperforms all other approaches, with median AUROC scores of 0.73 and 0.69 respectively. It should also be noted that the performance of the consensus approach is consistently better than that of the static Bayesian and dynamic Bayesian networks for all forms of simulated data tested. These results suggest that even with small sized networks (size 10), the consensus network gives robust results in predicting known interactions from benchmark datasets.

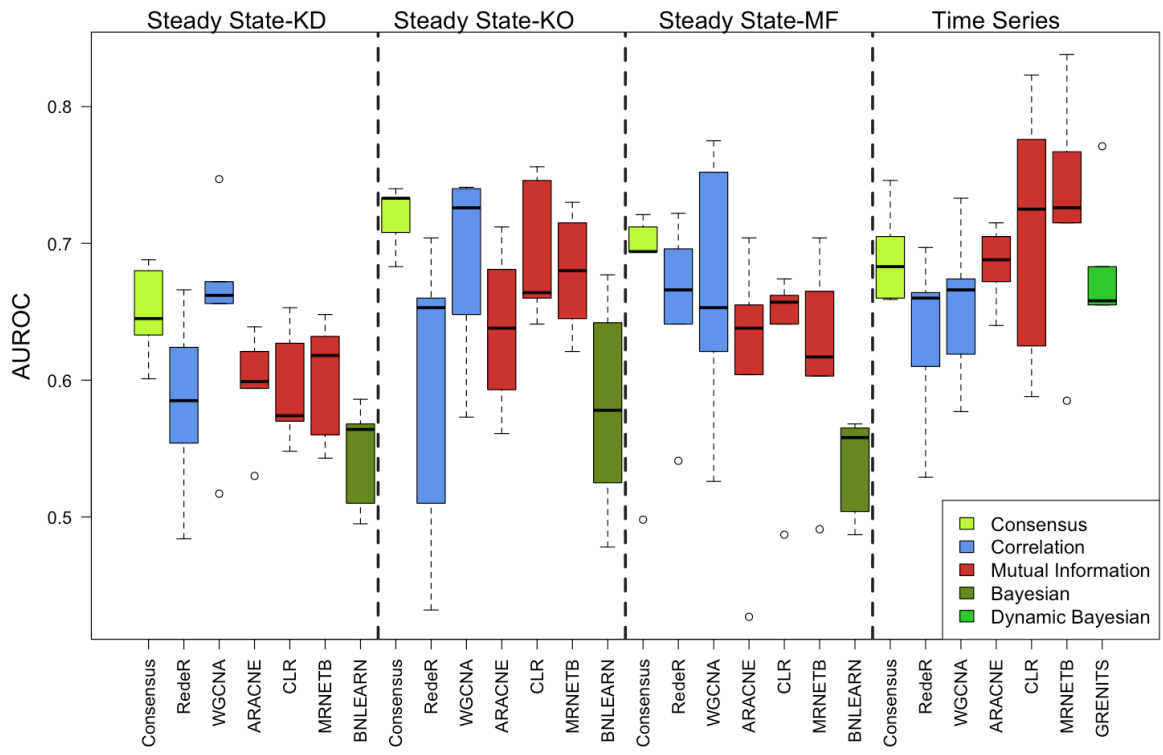


Figure 3.13: Performance scores of different network inference approaches using benchmarked DREAM4 challenge *in silico* datasets of size 10. Both steady state (KO-Knockout; KD-Knockdown; MF-Multifactorial) and time series data (Time Series) were used.

3.3.2 DREAM4 size 100

The boxplots in Figure 3.14 compare the AUROC scores obtained with different inference approaches for another medium sized DREAM4 challenge network (100 genes) under similar simulated experimental conditions as that of the size 10 network. The performance pattern of the consensus approach was similar to that obtained for the small size network (see Figure 3.13) except for the MF and TS experiments, for which it was ranked second and third respectively. It should be noted that with TS data, the mutual information based algorithms performed well compared to correlation-based methods.

Overall, these findings demonstrate that the consensus network by FCPT is consistent in capturing accurate interactions from several network inference algorithms for different types of steady state and time series gene expression data.

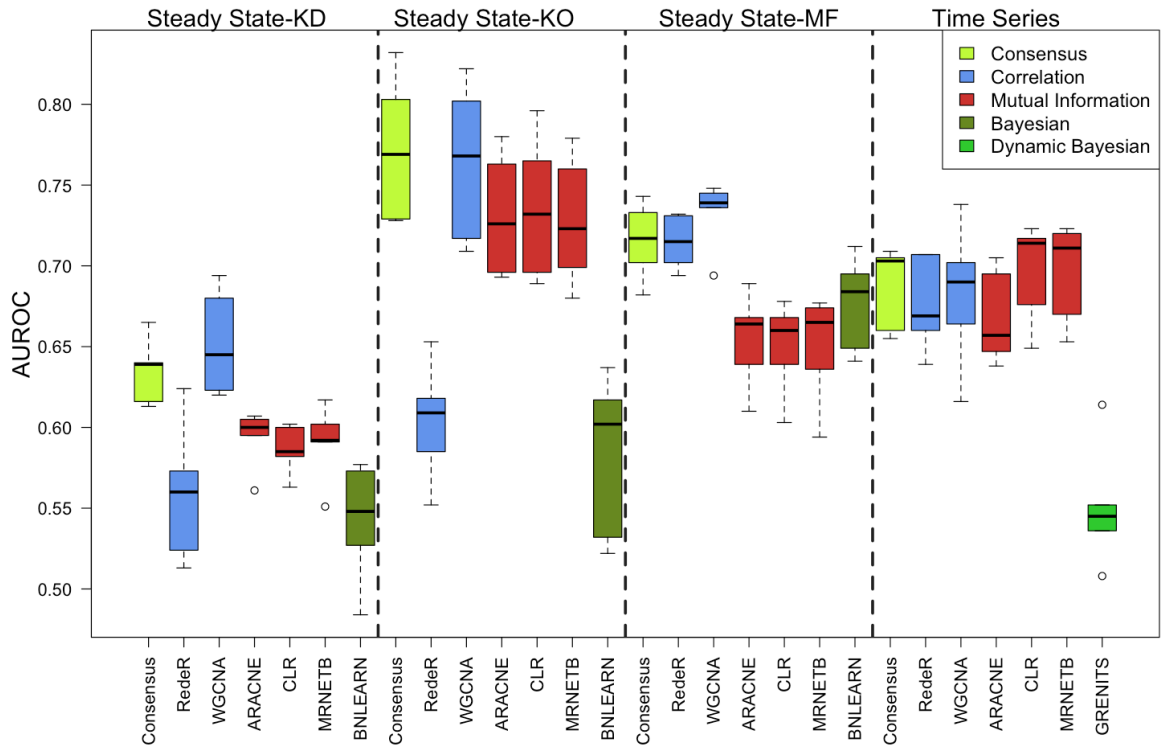


Figure 3.14: Compares performance scores of different network inference approaches using benchmarked DREAM4 challenge *in silico* datasets of size 100. Both steady state (KO-Knockout; KD-Knockdown; MF-Multifactorial) and time series data (Time Series) were used.

3.3.3 Consensus edges - overlap statistics

This section investigates the proportion of edges predicted by consensus networks (by FCPT) that overlap (are common) with the edges predicted by individual network inference algorithms. In order to perform an unbiased comparative analysis, the number of common edges was quantified using the Overlap Ratio (OR) at various rank thresholds for *in silico* benchmark networks (see Figure 3.15). The OR ranges between 0 and 1 - where 0 indicates no common edges - and 1 denotes perfect overlap. For example, in the size-100, sample-100

dataset, the top 1% of highly ranked edges from the consensus network comprises 10 edges, of which 6 edges are common with the top 1% highly ranked by RedeR; this yields an OR of 0.6 (i.e. 6/10).

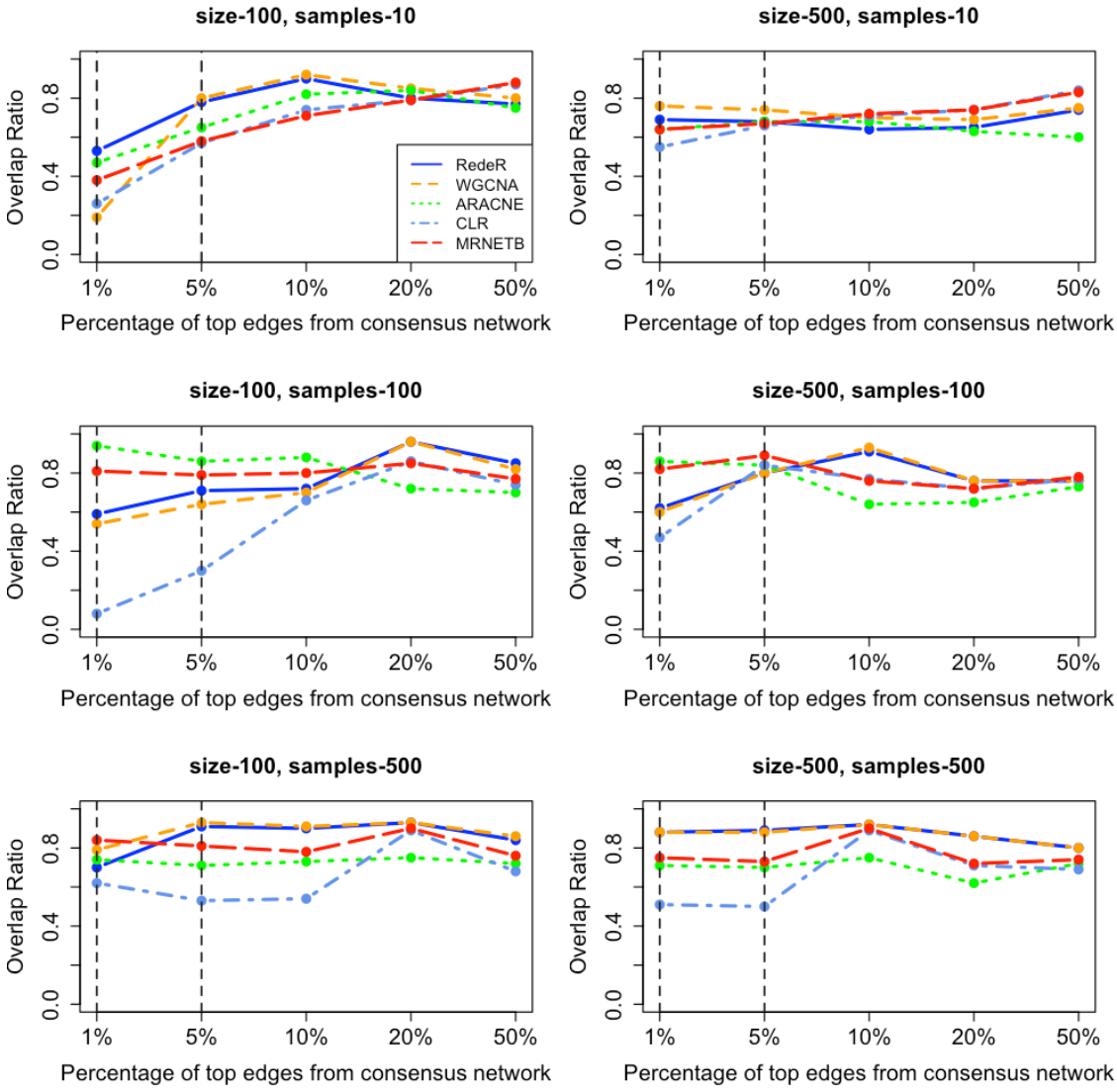


Figure 3.15: Overlap ratio of edges predicted by the consensus network method (FCPT) and individual inference methods for various thresholds of top ranked edges. Ratios were calculated from benchmark datasets of different dimension for networks of size 100 and size 500 generated by SynTReN.

One can see from Figure 3.15 that wide disparities exist in OR across datasets and networks at various thresholds. It is interesting to note that by relaxing the rank threshold, the OR

appears to converge to a common value. This trend was consistent with both networks sizes (100 and 500) across the different experimental samples tested. Overall, these results reveal that the edges included in the consensus network are not biased in favour of a particular network algorithm, but rather provide a cumulative contribution from all the networks.

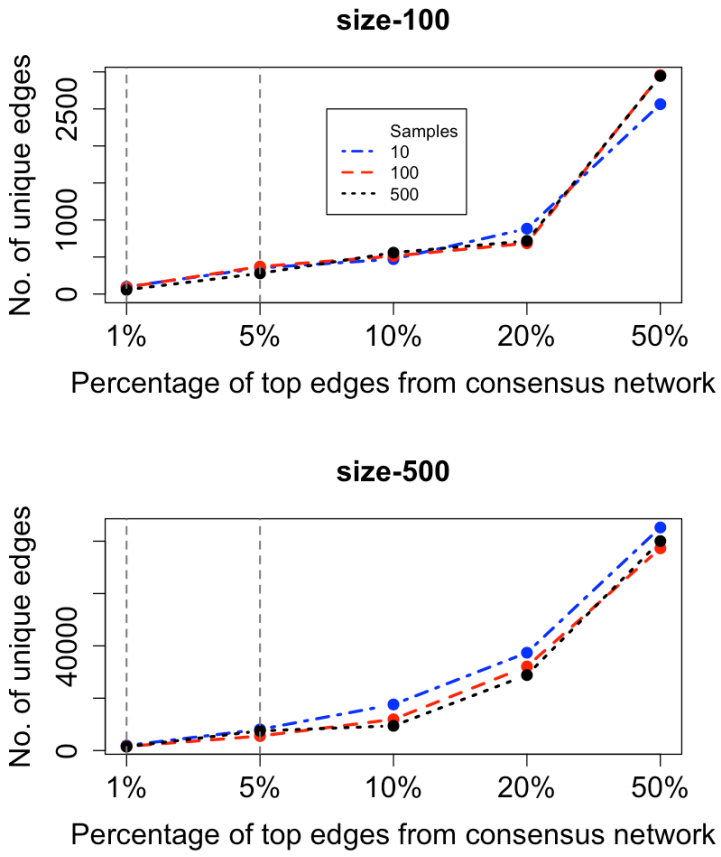


Figure 3.16: Number of unique edges predicted by the consensus (FCPT) network that are not common across individual networks for various thresholds of top ranked edges. The benchmark datesets used are the same as those described in Figure 3.15.

The number of edges uniquely determined by the consensus network (i.e. that are not common across individual inference methods) is illustrated in Figure 3.16. It is quite evident from the plot that the number of unique edges found by the consensus method is positively correlated with the percentage threshold. It is worth noting that even with a small sample

size, the consensus network is able to identify unique edges at various percentage thresholds. These results reflect the potential of consensus networks to identify edges that are missed by individual network algorithms.

The performance of the consensus and individual networks was also quantified using a statistical measure (sensitivity), as shown in Figure 3.17. This measures highlights the capacity of consensus and individual methods to predict true edges. At various fixed thresholds, the performance of the consensus method is consistent and robust for both network sizes. It should be noted that with a low number (10) of experimental samples, the sensitivity of the consensus method is on a par with the best performing individual networks over several thresholds.

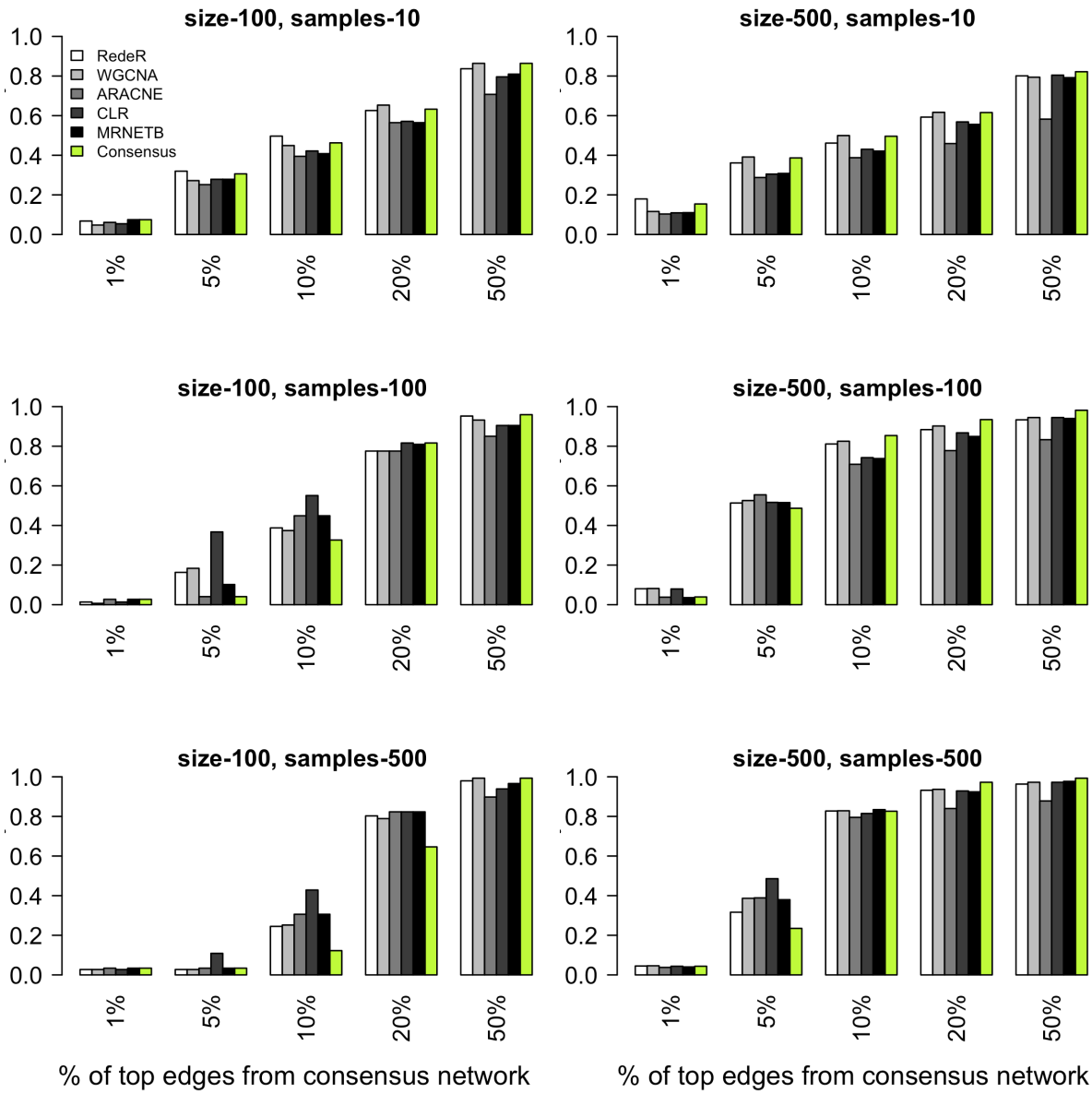


Figure 3.17: Sensitivity measures for edges predicted by consensus (FCPT) and individual inference methods for various percentage thresholds of top ranked edges. The benchmark datesets used are the same as those described in Fig 3.15.

3.3.4 Noise and Robustness

We further explored the robustness of the consensus network method by generating noisy expression data for networks of size 100 (i.e. 100 genes), and size 500 (i.e. 500 genes) with a sample size of 10 (i.e. 10 individual simulated perturbation experiments) at various noise levels (10%, 20%, 30%). Here, noise refers to experimental noise generated using the

SynTReN simulator. The performance AUROC measures for the consensus method and each individual inference methods for different noise levels are summarized in Table 3.3.

Table 3.3: AUROC scores obtained by applying different inference methods to *in silico* datasets of size 100 and size 500 with sample size 10 for different experimental noise levels (10%, 20% and 30%). For each noise level and dataset, highest scores are highlighted in bold.

	Noise-10%	***Noise-20%***	***Noise-30%***
	Size 100		
RedeR	0.777	0.759	0.747
WGCNA	**0.795**	**0.774**	0.755
ARACNE	0.707	0.675	0.67
CLR	0.741	0.74	0.736
MRNETB	0.746	0.736	0.734
Consensus-FCPT	0.787	0.771	**0.769**
	Size 500		
RedeR	0.756	0.736	0.713
WGCNA	0.758	0.74	**0.717**
ARACNE	0.656	0.628	0.606
CLR	0.738	0.655	0.662
MRNETB	0.731	0.656	0.661
Consensus-FCPT	**0.775**	**0.742**	0.712

With size 100 data - at a low noise level (10%) - the performance of the consensus network is second best, with an AUROC of 0.787. When the noise was increased to 20%, the performance of the consensus network worsened somewhat to 0.771 – the second best again. Although the consensus network AUROC decreased once more to 0.769 for 30% noise, it demonstrated the best performance in this case. Analysis with a large size network (500 genes) revealed that the consensus performed consistently well overall, yielding the best score for 10% and 20% noise, but ranked third best when noise was further increased to 30%. Overall, these findings suggest that the consensus by overall is a good alternative method for robust network inference.

3.3.5 Combining inference methods

In this section, we compared individual inference methods against the consensus networks obtained by combining different numbers of inference methods cumulatively in ascending order of their performance from weakest to strongest. For example, the two individual algorithms which yielded the lowest AUROC scores were first combined {1,2} to build an ensemble. Similarly, the worst three algorithms {1,2,3} were then combined, followed by the worst four {1,2,3,4}, until all network inference algorithms were included in the final consensus. This type of analysis is particularly useful for assessing the robustness of consensus by FCPT when a weaker performing method is combined with a better performing one. Here, we chose benchmark data from a medium sized network (size 100) with variable sample numbers.

The results obtained from this analysis are presented in Figure 3.18. The weakest performing algorithm (referred to as the individual method in the plot) was ARACNE in each case. We cumulatively integrated this algorithm with better performing inference methods using FCPT. It is apparent from Figure 3.18, that combining the weakest individual method with better performing ones improves the performance of the resulting consensus network. Furthermore, increasing the number of integrated methods monotonically increases consensus performance with any number of samples (i.e. perturbation experiments). It is interesting to note that by increasing the number of samples, the optimum consensus performance can be achieved by integrating only four methods (bottom plot of Figure 3.18). However, the integration of all five methods yielded the best results overall (AUROC- 0.871) with sample size 500. This finding suggests that the consensus network by FCPT is robust to combining inference methods of varying performance, and shows a synergistic effect. In contrast, combining the network algorithms in reverse order of performance from high to low is described in the latter section 3.3.6.4.

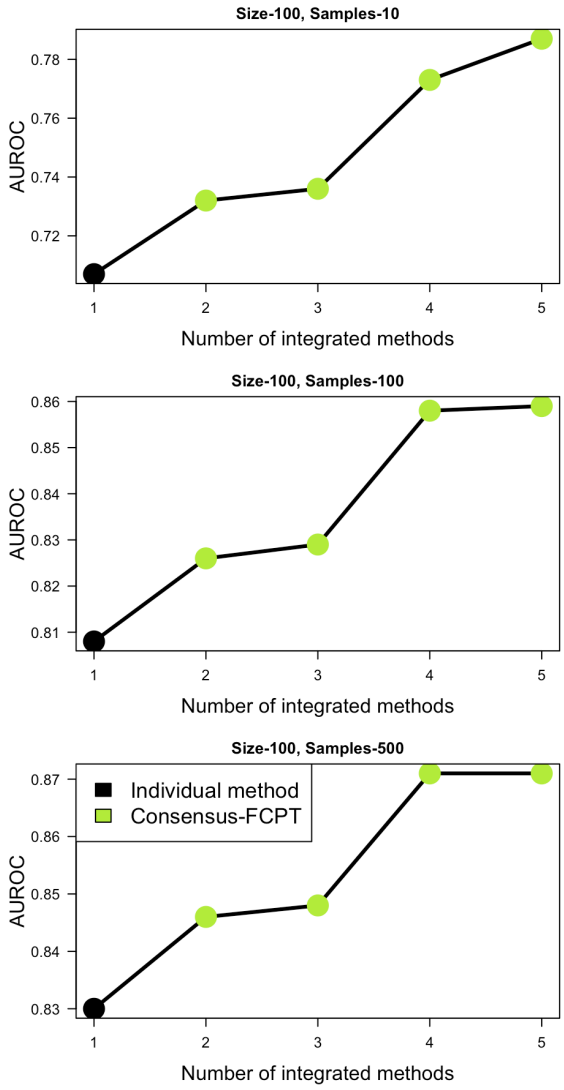


Figure 3.18: Comparative performance scores of individual and integrated network inference approaches using benchmarked *in silico* datasets generated from SynTReN of size 100 for different perturbation experiments (samples) of size 10, 100 and 500.

The order of integration of methods is shown below for the various benchmark *in silico* dataset dimensions tested:

Size 100, samples -10

1. ARACNE
2. ARACNE+CLR
3. ARACNE+CLR+MRNETB
4. ARACNE+CLR+MRNETB+RedeR
5. ARACNE+CLR+MRNETB+RedeR+WGCNA

Size 100, samples -100

1. ARACNE
2. ARACNE+ MRNETB
3. ARACNE+ MRNETB +CLR
4. ARACNE+ MRNETB +CLR+WGCNA
5. ARACNE+CLR+MRNETB+WGCNA+RedeR

Size 100, samples -500

1. ARACNE
2. ARACNE+MRNETB
3. ARACNE+MRNETB+CLR
4. ARACNE+MRNETB+CLR+RedeR
5. ARACNE+MRNETB+CLR+RedeR+WGCNA

3.3.6 Comparison with existing consensus methods

The consensus network built using FCPT was compared against popular existing qualitative and quantitative consensus methods[1] previously applied to integrate edge predictions using gene expression data. For qualitative consensus, we applied the Intersection and Union approaches; for quantitative consensus, we applied the Borda count election method (BCEM) and the Inverse variance weighted method (IVWM).

[1] More detailed descriptions of qualitative and quantitative approaches can be found in Chapter 2.

3.3.6.1 Qualitative consensus networks

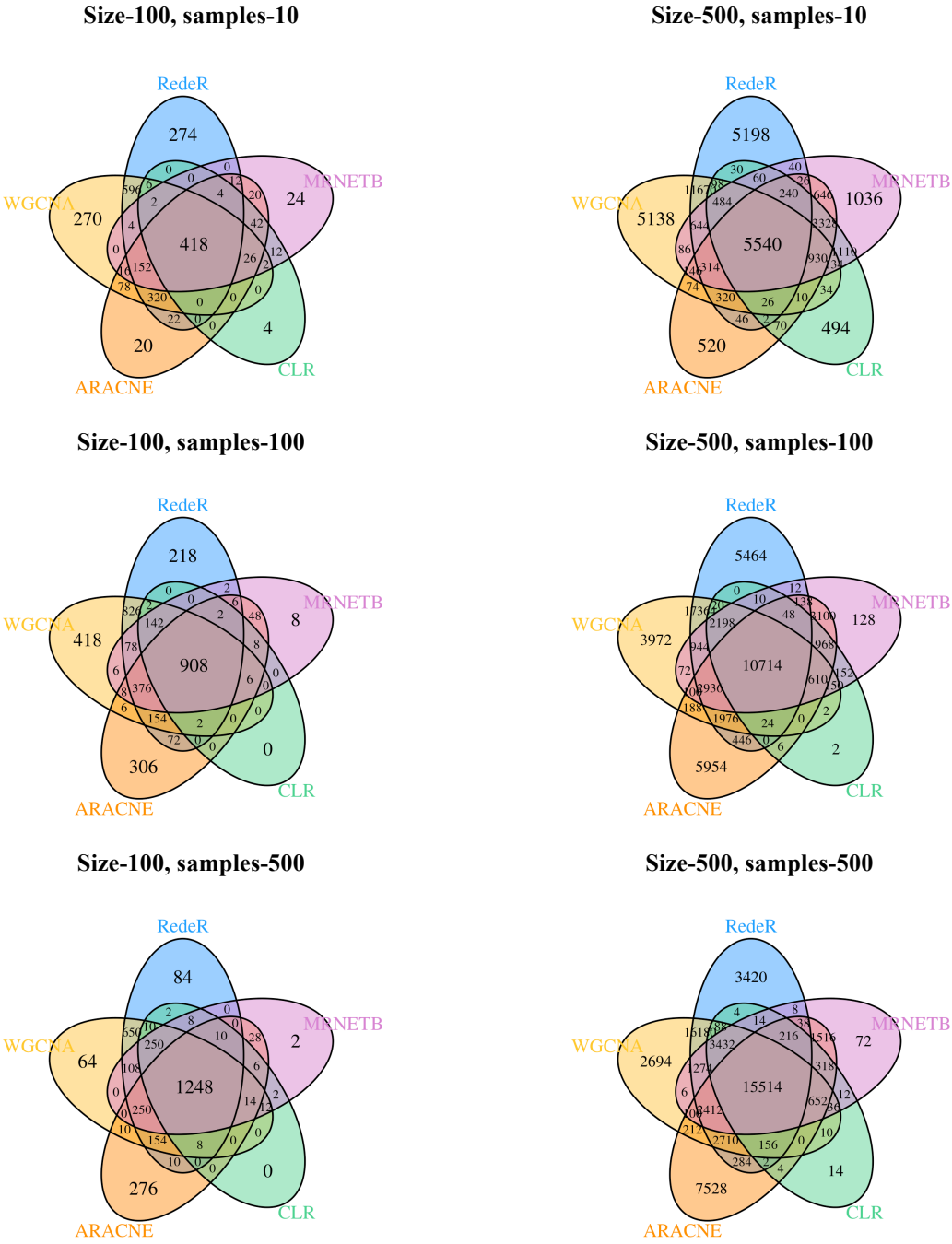


Figure 3.19: Venn diagrams showing the number of common predicted edges across different network inference algorithms at statistical significance level $p<0.05$ using the intersection consensus method. Results are shown, for datasets of various dimension generated using SynTRen from benchmark networks of size 100 and 500.

The qualitative consensus method is focused on recovering edges consistently across various networks at a particular fixed threshold. More specifically, it identifies the number of edges that are common (overlapping) across all inference algorithms. The two qualitative consensus methods (intersection and union) were implemented by extracting all the edges from different inference algorithms that had significance scores (*p*-values) less than 0.05, before constructing a qualitative consensus network using Venn diagram based approaches. Furthermore, these networks were compared against those generated by FCPT at the significance level[2] of 5% FDR (i.e. $q<0.05$). In this analysis, the consensus networks constructed via the qualitative method were compared against Erdős–Rényi (ER) random networks (P. Erdös 1959) with exactly the same number of vertices $|V(G)|$ and edges $|E(G)|$ as the network obtained using FCPT (ER graphs are constructed by connecting every pair of nodes with equal probability). Furthermore, to ensure an unbiased analysis, we generated 100 different random networks in each case and measured performance in terms of the mean sensitivity and specificity. ER networks were generated using the igraph module (Csardi 2010).

Here, the Venn diagrams shown in Figure 3.19 demonstrate the qualitative consensus networks obtained by applying the intersection method to *in silico* benchmark data of various dimensions. It can be seen that increasing the sample number increases the number of consistent edges concomitantly. This trend was observed for both size 100 and size 500 networks. It was observed (Figure 3.20) that with the fewest number of samples (10), the sensitivity of the Intersection method was poor for networks of both size 100 and size 500 compared to the Union and FCPT methods. However, the Intersection method gave the highest specificity (0.96) and as sample number increased, sensitivity improved across all the methods. Notably, the Union and FCPT methods gave a higher sensitivity (0.98) for sample

[2] See the method section as how *q*-values were calculated from *p*-values in order to control false discovery rate (FDR).

size 500 and network size 100. Correspondingly, the specificity of FCPT (0.75) was higher than that of Union (0.69). A similar trend was observed with size 500 networks. The performance of the random network was poor across all the networks tested.

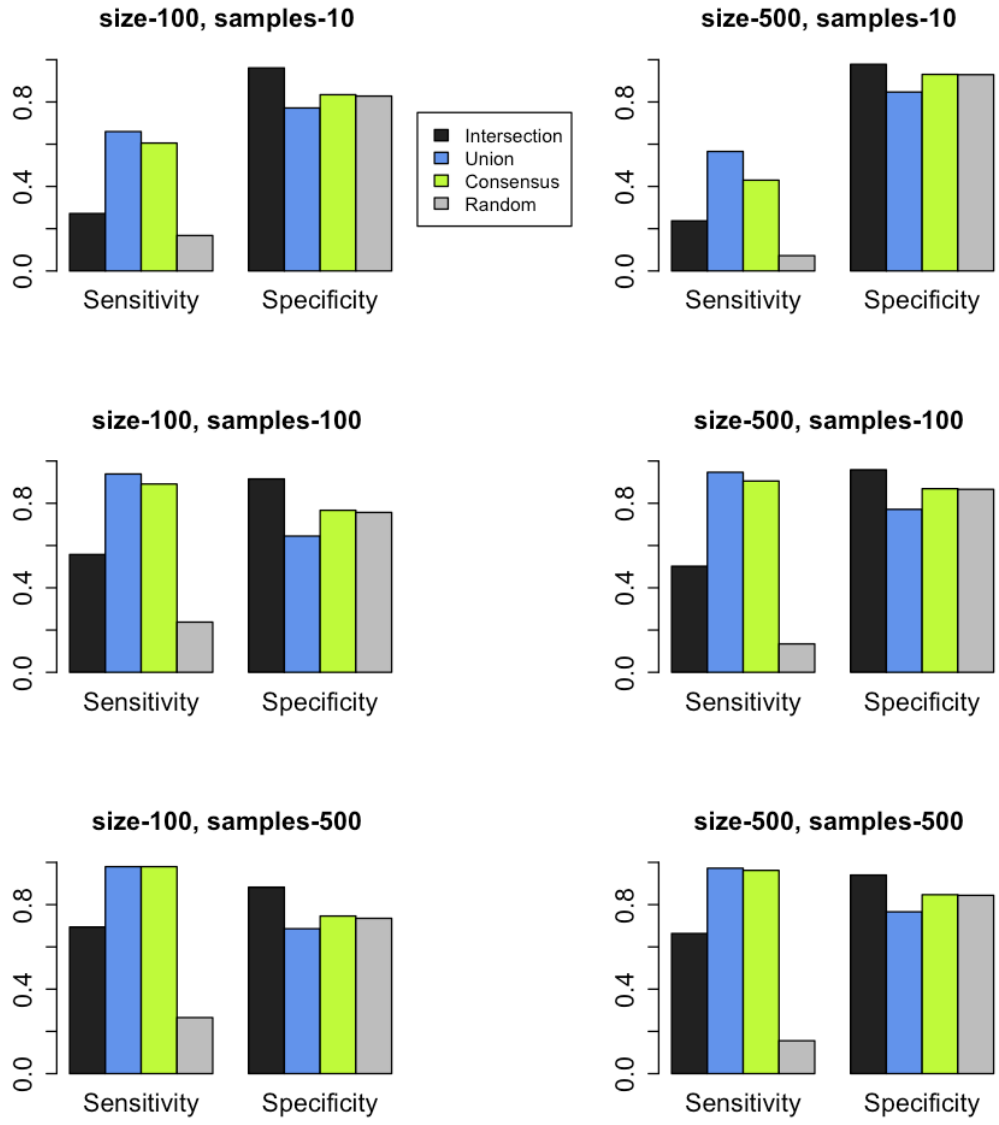


Figure 3.20: Sensitivity and specificity values obtained with qualitative consensus methods (intersection and union) at significance level $p<0.05$ compared with those obtained with quantitative consensus by FCPT at significance level $q<0.05$ and random networks.

Overall, these results suggest that the Intersection method yields the highest specificity, with a trade-off of poor sensitivity. This is because the Intersection method reduces the number of prediction, eventually missing out on many true predictions present in

the target networks. Conversely, the Union method trades off high sensitivity with lower specificity because it increases the number of predictions, eventually giving as many true positives as the Intersection and FCPT method. However, FCPT performs consistently well in terms of both sensitivity and specificity compared to the qualitative consensus methods. In particular, FCPT performance improves with increased sample size.

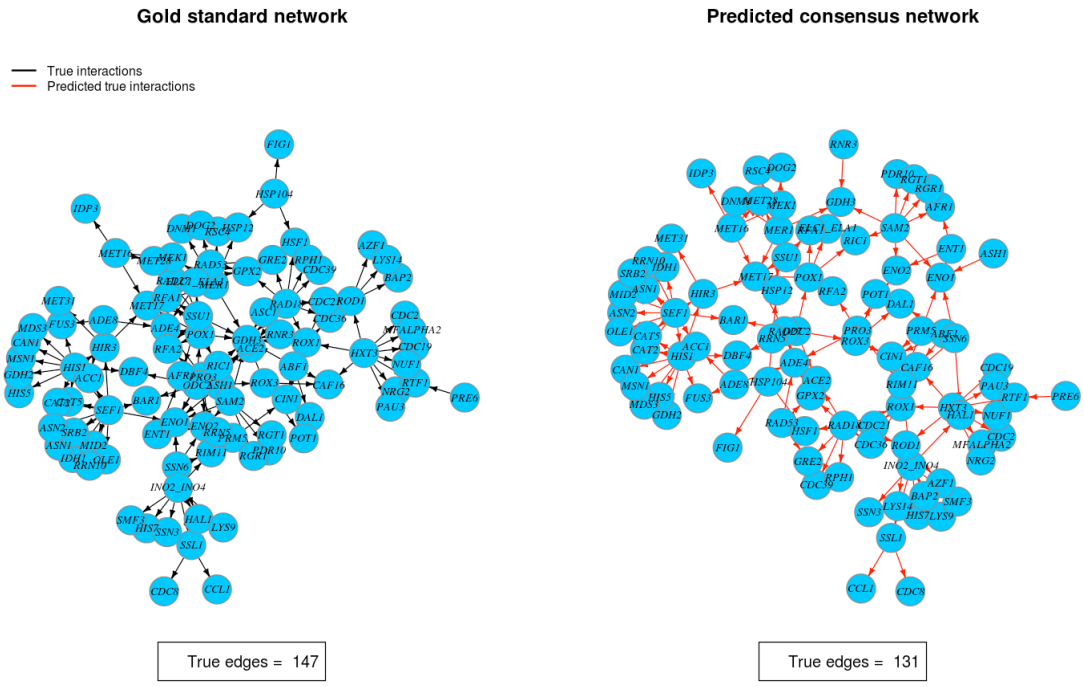


Figure 3.21: Gold standard (true) network (left plot) and predicted consensus network by FCPT at significance threshold q<0.05 (right plot) obtained from a benchmark *in silico* expression dataset of size 100 with 100 samples. Black and red edges indicate true edges of the gold standard and consensus networks respectively

Figure 3.21 compares the gold standard network and consensus network by FCPT at significance level 5% FDR (q<0.05) for size 100 *in silico* benchmark data with 100 experimental samples. In the graph, the black edges denote real edges from the gold standard network, while red edges indicate predicted true edges from the consensus network by FCPT.

The most surprising aspect of this finding is that the consensus network successfully predicted 131 out of 147 true edges, with a sensitivity of 0.89 and a specificity of 0.77.

Likewise, a similar analysis was performed for size 500 networks, which is shown in Figure 3.22. From this graph we can see that out of 868 true edges, the consensus network identified 786 with a sensitivity of 0.91 and a specificity of 0.87. It is worth mentioning that the performance of the consensus network was better with large size networks (size 500) were compared to medium size networks (size 100) in terms of both sensitivity and specificity. Taken together, these results suggest that there is a clear benefit to employing consensus learning by means of FCPT to predict regulatory interactions.

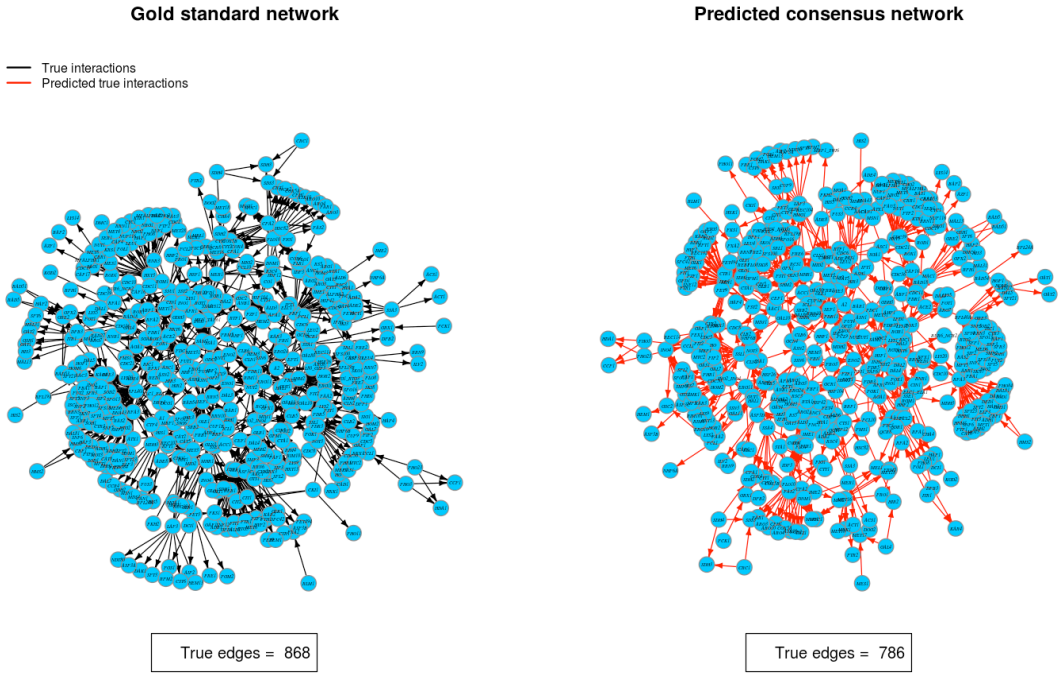


Figure 3.22: Gold standard (true) network (left plot) and predicted consensus network by FCPT at significance threshold $q<0.05$ (right plot) obtained from a benchmark *in silico* expression dataset of size 500 with 100 samples. Black and red edges indicate true edges of the gold standard and consensus networks respectively.

3.3.6.2 Quantitative consensus networks

To further investigate the potential of Fishers combined probability test (FCPT), we compared it against other quantitative consensus approaches that have previously implemented to reconstruct gene regulatory networks from microarray gene expression data (Steele & Tucker 2008; Marbach et al. 2010). The most popular methods for building quantitative consensus networks are the Borda Count Election Method (BCEM), and the Inverse Variance Weighted method (IVWM)[3]. In the current section, we focus our analysis on comparing the performance of FCPT against that of BCEM and IVWM using *in silico* gene expression datasets generated from SynTReN, and also the DREAM4 challenge datasets (Table 3.1). The performance of each of these consensus methods was evaluated using our previously established validation framework (Figure 3.1). Performance was measured in terms of the Area Under the Receiver Operating Characteristic Curve (AUROC) in order to assess the relationship between sensitivity and specificity at various possible significance thresholds.

Figure 3.23 compares ROC curves and corresponding AUROC values for medium sized networks (size 100) and large sized networks (size 500) with sample sizes of various dimensions, as described in Table 3.1. We can deduce from those ROC curves that most of the consensus methods yielded accurate results for both sized networks. The most striking result was that the FCPT outperformed BCEM and IVWM for both sized networks when the number of experimental samples was lowest (sample size 10). However, with an increase in sample size, the performance of all the consensus methods improved for both sized networks.

[3] See the consensus learning section in Chapter 2 for detailed explanation of the BCEM and IVWM methods.

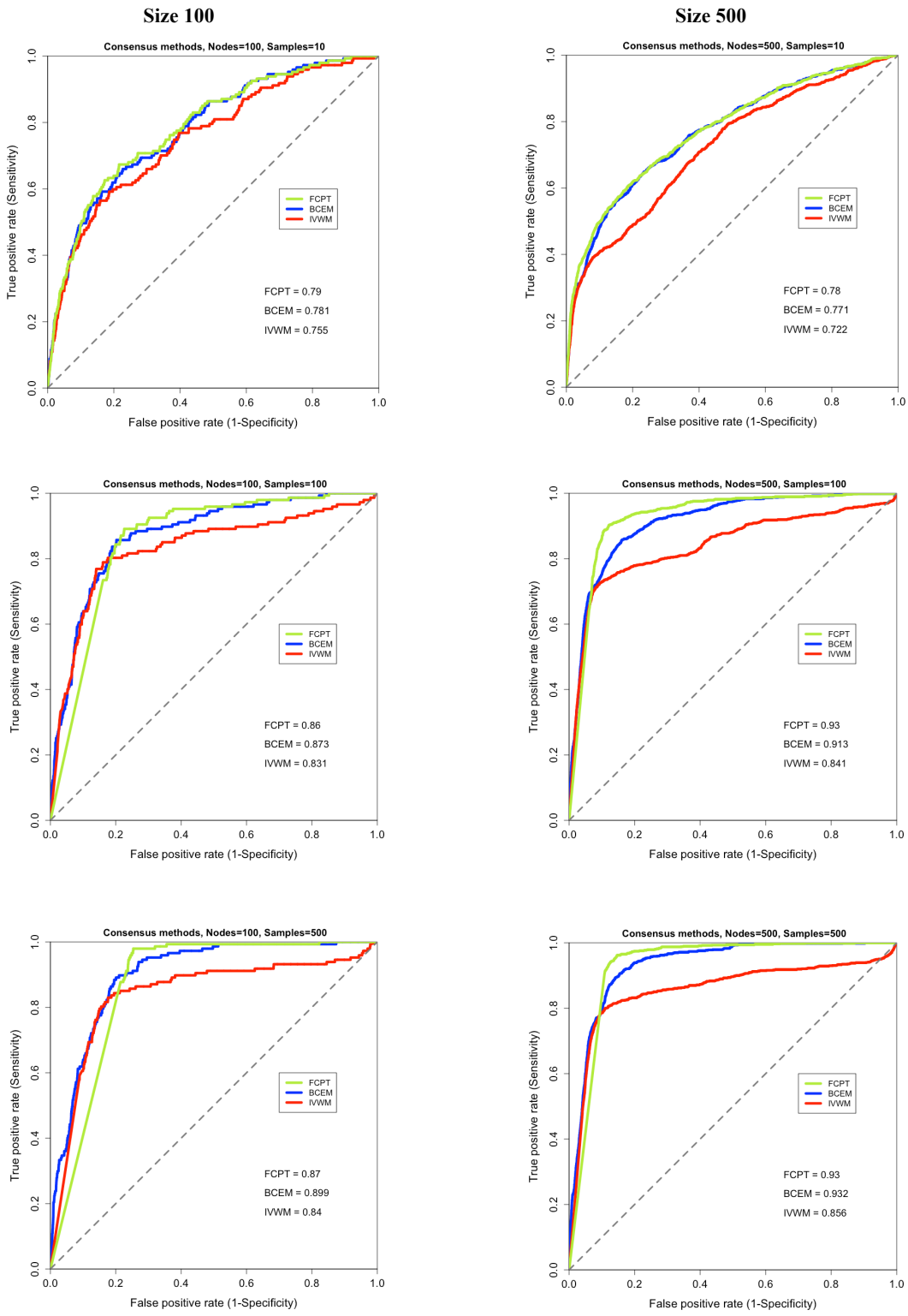


Figure 3.23: ROC curves and corresponding AUROC values for existing quantitative consensus approaches using benchmarked *in silico* datasets of size (nodes) 100 and size 500 with sample sizes 10, 100 and 500 (perturbation experiments) generated from SynTReN. Abbreviations: FCPT-Fishers Combined Probability Test, BCEM-Borda Count Election Method, IVWM- Inverse Variance Weighted Method.

In particular, the performance of BCEM modestly improved modestly when sample size increased, and outperforming FCPT and IVWM for the size 100 network. In contrast, the performance of FCPT improved, and was at par (AUROC-0.93) with BCEM for the size 500 network when sample size increased. For both sized networks, IVWM was ranked last amongst the methods. This is possibly due to the fact that IVWM calculates the weighted mean effect size using confidence scores for each edge, in order to deliver its robust combined effect, with high confidence scores influencing the weighted mean more than when low confidence scores. If the variance across the predicted edge scores from different networks is large, as in this case, the edge confidence based on consensus is weak. Overall, these findings indicate that the performance of FCPT was better than or comparable to the performance of many well-established consensus methods for the datasets tested.

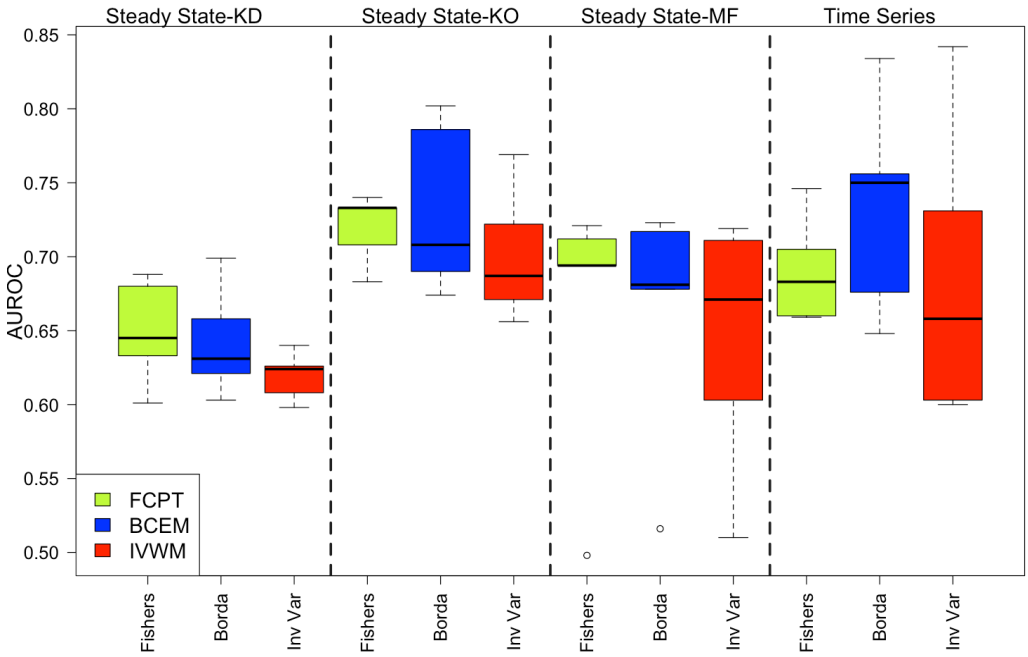


Figure 3.24: Performance scores of equantitative consensus methods using benchmarked DREAM4 challenge *in silico* datasets of size 10 (10 nodes) from different perturbation (KO-Knockout; KD-Knockdown; MF-Multifactorial) and time series (Time Series) experiments.

The comparative performance of the different consensus methods was further assessed using DREAM4[4] challenge data of size 10 (10 nodes) as shown in Figure 3.24. The results 3.24 revealed that all the consensus methods performed well across various forms of *in silico* experimental data (steady state and time series) when evaluated using median AUROC scores. What is interesting to note is that FCPT outperformed all other consensus methods for steady state data, and ranked second-best with time series data. BCEM performed second best with steady state perturbation data, and best with time series data. IVWM was found to perform poorly in comparison to FCPT and BCEM under the steady state and time series data tested.

Interestingly, with medium size networks (100 nodes), the performance of the consensus methods was found to be similar to for small size networks (10 nodes); see Figure 3.24 and Figure 3.25. However, it can be observed that the performance of FCPT improved with increased network size from 10 to 100. The most striking result to emerge from these results is that for the steady state datasets, FCPT showed improved performance to those of the other consensus methods for many combinations of network size and sample number. However, with time series data, FCPT lost first place marginally, yielding a median AUROC score of 0.71, compared to the BCEM score of 0.73. BCEM was consistently ranked second with steady state data, but outperformed all other methods for time series data. Consistent with the results for size 10 DREAM4 data, IVWM performance was weakest with all kinds of data used.

[4] The DREAM4 challenge provides benchmark data that has been used to assess more than 30 different network inference algorithms, including various steady state (KO, KD and MF) and time series *in silico* experimental datasets. See the Methods section for details.

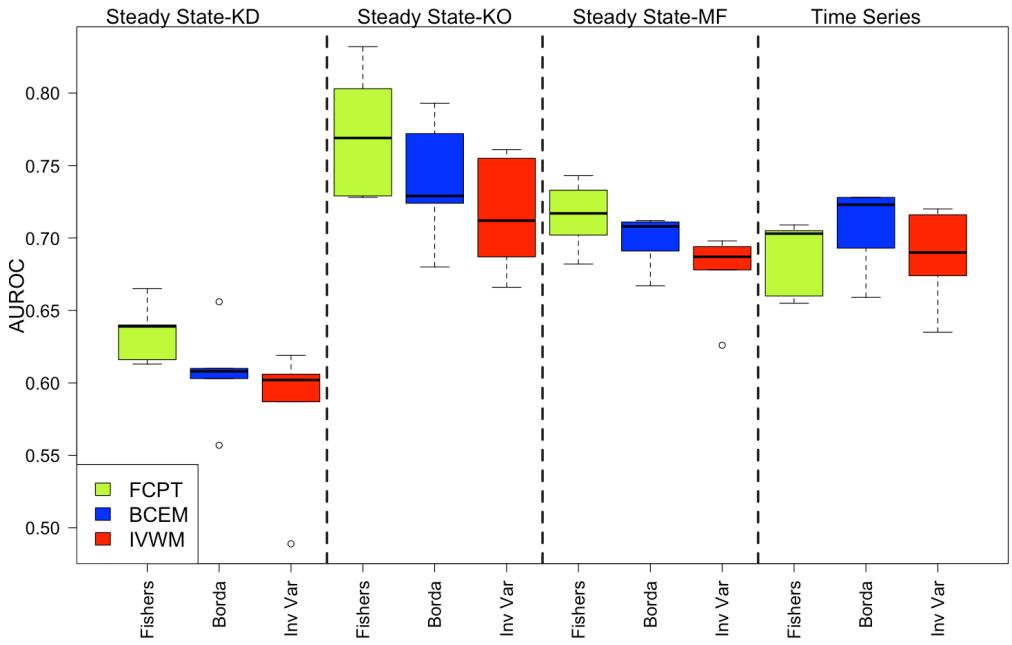


Figure 3.25: Performance scores of quantitative consensus methods using benchmark DREAM4 challenge *in silico* datasets of size 100 (100 nodes) from different perturbation (KO-Knockout; KD-Knockdown; MF-Multifactorial) and time series (Time Series) experiments.

Overall, the comparison of these consensus methods revealed that, reasonably often, FCPT outperformed BCEM and IVWM with various steady state and time series experiments, for small (10 gene) and medium (100 gene) sized networks from the DREAM4 benchmark datasets. Furthermore, there is a clear benefit to building consensus networks by FCPT, as it delivers robust predictions when tested with various experimental gene expression data types of different dimension.

3.3.6.3 Efficiency of the consensus methods

The efficiency of the consensus algorithms was evaluated based on processing time for all edge weights (predictions) to be combined. Higher processing time corresponds to higher computational cost and hence to the lower efficiency of the algorithm. Here, we compared the

efficiency of FCPT against other consensus methods, for various benchmark *in silico* datasets of size 10, 100 and 500, with variable experimental samples sizes, (see Table 3.1). One can see that FCPT and BCEM are almost equally efficient for combining predictions for small sized networks (size 10) with a processing time around 0.05s. However, for medium sized networks (size 100), the overall efficiency of FCPT was marginally behind BCEM but much better than IVWM.

Table 3.4: Average processing times (in seconds) for different consensus methods when combining predictions using benchmarked *in silico* datasets of size10, 100 and 500 with variable number of perturbation experiments (samples). The lowest time clocked for each dataset size is highlighted in bold.

Consensus algorithms	*Size 10*	*Size 100*	*Size 500*
FCPT	0.052	5.79	**250.48**
BCEM	**0.048**	**3.36**	358.39
IVWM	1.012	108.51	4084.95

However, for large sized networks, the efficiency of FCPT was best, with BCEM combining all the edges under 6 minutes (360s) and IVWM ranking last. A plausible explanation for the higher computational cost of the latter algorithm is that it performs several intermediate calculations when determining statistical parameters for each edge before returning a final consensus score. Taken together, these findings suggest that the efficiency of FCPT is consistently good for a range of network sizes. These computations were performed on a single core CPU (Intel® Core™ i7 CPU 860 @ 2.80GHz) running Ubuntu 12.04 OS with 12 GB of volatile memory.

3.3.6.4 Robustness of consensus methods

In this section, we explore the robustness of individual consensus methods when networks are constructed from noisy gene expression data. Given that noise is an inevitable property of any gene expression dataset, it is imperative to investigate how consensus methods perform when exposed to noise. Here, we compare the performance of FCPT against that of BCEM and IVWM, using AUROC measures with *in silico* benchmark data simulated under a range of experimental noise levels. The results obtained from this analysis are described in Table 3.5.

Table 3.5: AUROC scores for different consensus methods obtained from SynTReN datasets of size 100 and size 500 with sample size of 10 with different experimental noise levels (10%, 20% and 30%). Highest scores for each noise level are highlighted in bold.

Consensus algorithms	*Noise-10%*	*Noise-20%*	*Noise-30%*
Size 100			
FCPT	**0.787**	0.771	0.769
BCEM	0.781	**0.775**	**0.771**
IVWM	0.755	0.745	0.734
Size 500			
FCPT	**0.775**	**0.742**	**0.712**
BCEM	0.771	0.737	0.704
IVWM	0.722	0.672	0.638

The performance trend of FCPT was found to be consistent with the other consensus methods, with increasing in noise causing the performance to decline. This trend was found to be similar for medium and large sized networks. Nevertheless, it is worth mentioning that for large sized networks (size 500), FCPT consistently ranked first despite this decrease in performance score. For medium sized networks (size 100), however, BCEM worked

comparably well. These findings indicate that the FCPT consensus network is sufficiently robust to handle noisy data and provides higher confidence scores with larger size networks.

An alternative way of assessing robustness is to identify how well a consensus method performs when it is combined with poorer performing algorithms. More specifically, we investigated how well the performance of individual consensus methods varied when the best performing network inference algorithms were combined with weaker ones in descending order of performances. For example, the two algorithms yielded the top AUROC scores, were first combined {1,2} to build an ensemble after which the three best algorithms {1,2,3}, and then the four best algorithms {1,2,3,4} were combined, and so on, until all the network inference algorithms were included in the final consensus. The performance of the consensus networks was evaluated similarly to that of the individual networks. This type of combination approach was adapted from (Marbach et al. 2010).

Figure 3.26 shows the result of applying this approach to benchmark data from a medium sized network (100 genes) and a large sized network (500 genes) for various sample sizes (10,100 and 500) in order to provide insights into the effect of these factors on the performance of the consensus methods.

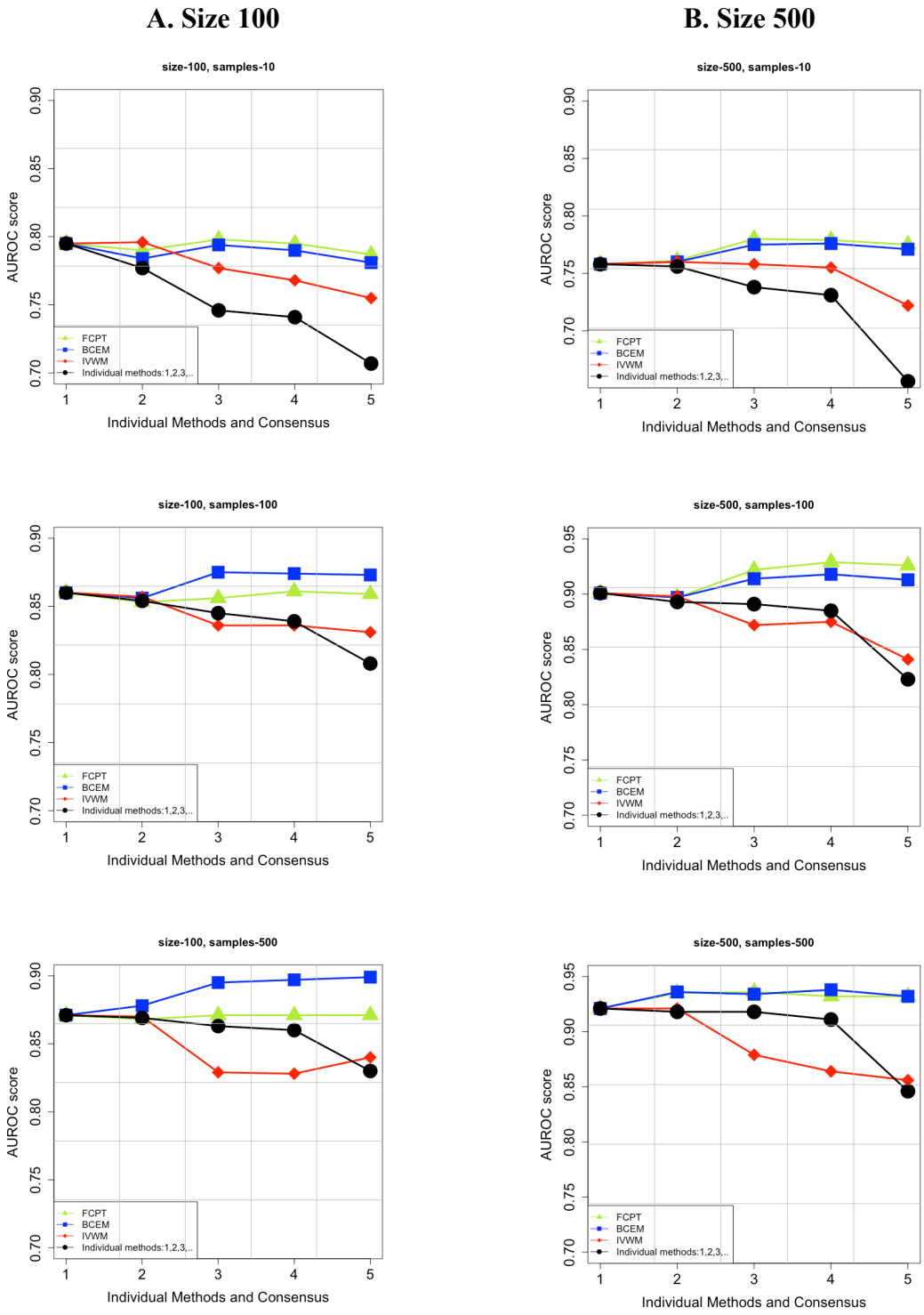


Figure 3.26: AUROC scores obtained by combining different inference methods using *in silco* datasets of various dimension generated from SynTReN. AUROC scores are also shown for the individual methods. A: Result obtained using data from the 100 gene network for sample sizes 10, 100 and 500. B: Results obtained for the 500-gene network using the same sample sizes as that of A.

It can be seen in Figure 3.26A that for medium size networks (100 genes), increasing the experimental sample size caused the performance of all the consensus methods to increase concomitantly. Here, combining the two best methods using FCPT, BCEM and IVWM yielded similar AUROC scores. Furthermore, the performance of FPCT and BCEM remained consistent when the ensemble size was increased by combining more algorithms, revealing that the FCPT and BCEM consensus methods are robust with respect to weaker methods. In contrast, it is apparent that the performance of IVWM decreases with the addition of weaker performing algorithms, suggesting that IVWM is not as robust as FCPT and BCEM to low confidence predictions. Overall, for medium sized networks (100 genes), FCPT performed well with fewer experimental samples and BCEM performed better when the number of experimental sample sizes increased. Figure 3.26B shows the results obtained from large size networks (500 genes) with various sample sizes. FCPT consistently worked well, showing superior performance in many cases, and was also the most robust of the methods used

Overall, these findings suggest that FCPT is a more robust consensus method, with the ability to handle weaker performing algorithms and still deliver consistent predictions. FCPT also shows less variation in performance with variable sample size.

3.4 Discussion

In this chapter, we proposed and investigated a novel quantitative consensus approach, which employs the Fisher combined probability test (FCPT) to combine the predictions obtained from correlation (RedeR, WGCNA) and mutual information (ARACNE, CLR and MRNETB) based reverse engineering methods. The reason we focus on these particular methods is their ability to handle high dimensional gene expression data with a lower computational cost. Overall, the findings of this study indicate that consensus network

performs consistently better in contrast with single methods when identifying regulatory interactions between a transcription factor and its targeted genes under various types of *in silico* benchmark datasets, including data from the DREAM4 challenge. The reason to choose *in silico* benchmark networks and corresponding simulated expression data is that the target network is known in advance. We validated our consensus approach using a variety of static and time series gene expression datasets (Table 3.1 in Chapter 3), as most real expression data contain both. The use time series data has an added advantage over static data. For instance, the inference methods, which use static expression data, cannot distinguish between regulators that actually have a direct causal effect on their targeted genes and genes that are co-expressed with other regulators. This problem can be partially alleviated by inferring a network from time series data that encapsulates dynamics, and contains information about direct causal effects between a transcription factor and its targeted genes (De Smet & Marchal 2010).

Deeper investigation by means of qualitative analysis of consensus by FCPT against individual methods revealed that FCPT identifies unique interactions and that there was no perfect overlap between these predicted interactions across many single inference methods, thus signifying that the FCPT method gives unbiased predictions (see Fig 3.15 and Fig 3.16). In addition, the accuracy of consensus predictions was found to be better than many single methods, in terms of sensitivity and specificity measures at various rank thresholds (Fig 3.17).

Previous studies by Steele *et al* (Steele & Tucker 2008) and Marbach *et al* (Marbach et al. 2012) have demonstrated the power of the consensus network when combining predictions using resource diversity (i.e. multiple expression datasets) and species diversity (*E.coli, S.cerevisiae*) respectively. Steele *et al* (Steele & Tucker 2008) used the inverse-variance weighted method (IVWM) (DerSimonian & Laird 1986) to quantitatively combine

bootstrapped predictions from Bayesian networks. The key issue with the IVWM is that it depends on a sample size (i.e. number of studies) which correspondingly increases its precision when a large number of high confidence interactions are combined and decreases precision when low confidence interactions are present (Deeks et al. 2008). Indeed, applying IVVM to benchmark networks in this study revealed that the method delivers robust predictions when top performing individual inference algorithms are integrated, but that the addition of any weaker algorithms deteriorates the ensemble of predictions (Figure 3.26). More precisely, IVWM delivers a robust prediction for a given edge provided all the inference algorithms predict that same edge to be statistically significant (high confidence). If any of the algorithms has predicted that edge to be insignificant, then IVWM combines the aggregates into an overall edge score to be of low confidence. In their study, Marbach *et al* (Marbach et al. 2012) employed the Borda count election method (BCEM) to integrate results from different predicted networks. However, BCEM does not satisfy the majority rule (Erdmann 2011). For example, if an edge is selected by the majority of network inference methods, then it is not necessarily the case that this edge will be included in the resultant consensus network. In addition, transforming edge probability measures to average ranks can make quantifying measures less credible, as an important feature of probability measurements are lost and an important edge can be missed in the final consensus. The most interesting finding of this study was that for larger networks of size 500, the consensus network outperforms many single inference methods including Bayesian networks, when performance is measured using AUROC scores (see Fig 3.12). This suggests that the consensus method provides accurate measurements and better value for large-scale network inference. Another important finding was that at a high level of experimental noise (30%), the FCPT consensus method demonstrated the best performance for size 100 networks and third best for size 500 networks in terms of AUROC measures (Table 3.3). Taken together, these findings suggest

that the consensus network is a good alternative method for robust network inference. It is also interesting to note that FCPT is equally computationally efficient as popular existing consensus methods (Table 3.4) for combining predictions for medium sized networks (size 100), also outperforming BCEM and IVWM reasonably often for both medium and large sized networks (size 500) with a higher computational efficiency.

Although the consensus network improves the accuracy of predictions, it is important to bear in mind that a consensus network is not always superior to the individual method, depending on the measurement data. For example, in this study the results from the MF experiment of the DREAM4 challenge for small (10 genes) and medium (100 genes) size networks (see Fig 3.13 and 3.14) showed the performance of consensus by FCPT to be best, and second best respectively. By contrast, with time series data, the consensus network ranked third compared to the other single methods for small and medium sized networks. In addition, previous findings from Marbach *et al* (Marbach et al. 2012) which implemented BCEM to generate a consensus network showed high performance measures for predicting true interactions with simulated and real *E.coli* data, but low performance measures with *S.cerevisiae* data. The authors attributed the poor correlation between mRNA levels of transcription factors and their target gene in *S.cerevisiae* as a plausible cause for this low performance. This therefore indicates that consensus networks may not be the best approach on all occasions and requires further research. Nevertheless, the consensus approach is a powerful and robust tool in reconstructing biological networks for unbiased predictions (De Smet & Marchal 2010).

3.5 Conclusions

This chapter proposes and investigates a new consensus network inference approach, which employs the Fisher combined probability test (FCPT) to integrate the predictions obtained

from five diverse, commonly used reverse engineering algorithms. Prior to employing FCPT statistics to build a consensus network, we also developed a non-parametric random sampling by permutation algorithm which converts edge statistics to *p*-values for edge consistency. FCPT - which is a single hypothesis test - was further extended by performing multiple hypothesis tests via the controlling of the false discovery rate (FDR) in order to get rid of spurious edges (Dabney A, Storey JD 2013). Consensus approaches are well known for their accuracy and robustness in decision making. Here, we explore the power of consensus by FCPT against two established strategies to build consensus networks for comparative analysis: 1) naïve or qualitative consensus, and 2) quantitative consensus. Naïve consensus methods are often too conservative and thus are not a good alternative for delivering a robust network. The FCPT, in theory, is a sound approach for quantitative consensus decision-making, as it has been successfully used in many other disciplines. We show in this chapter that predictions identified by the FCPT consensus network, provide comparable or better performance than the best single inference methods, and those existing quantitative consensus methods (BCEM and IVWM) when tested on various types of *in silico* benchmark data. Furthermore, the complementary and sub-optimal interactions identified through different network inference methods are accurately combined by consensus to generate a robust network. We also confirmed that the consensus network is not random by comparing it with the Erdős–Rényi random networks.

In summary, the findings from this chapter reveal that consensus by FCPT is a reliable means for predicting accurate gene networks that overlap closely to the target network. In the next chapter, we move on to focus our attention to addressing modularity in hierarchical networks generated from simulated gene expression data.

Chapter 4

Comparative analysis of network algorithms to address modularity

Abstract

This chapter provides a comparative analysis of different network algorithms in order to address modularity, using in silico gene expression data. Three networking algorithms were selected to study complex biological network modularity: RedeR, weighted correlation network analysis (WGCNA) and statistical inference of modular networks (SIMoNe). A new quantitative score was proposed for evaluating their ability to identify biologically meaningful modular networks. A section of this chapter has been published in conference proceedings (Mohammed 2013).

4.1 Introduction

In the previous Chapter, we focused our analysis on investigating consensus networks and their ability to accurately and robustly infer gene regulatory networks (GRNs). Previously, many studies have explored the property of modularity in GRNs. Modularity is a ubiquitous characteristic of biological networks, that serves as a building block to process biological functions at the cellular level. Modules are a group of genes, proteins or metabolites that can be co-regulated and which govern either a complex biological process or a molecular function in a highly coordinated manner (Barabási & Oltvai 2004). This underlying coordination of functional modular genes and their relationships can be organized in a hierarchical manner, forming a network that encapsulates the functional organization of a cell, as shown in Figure 4.1.

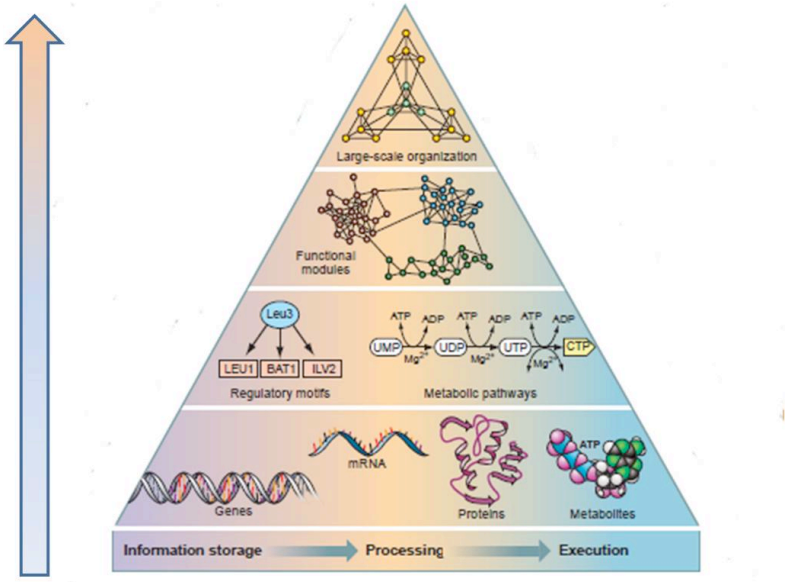


Figure 4.1: Pyramid structure of a cell's complexity. Information quantity and level of complexity gets exponentially larger and more complex as we move from the bottom to top. The figure was adapted from (Oltvai & Barabási 2002).

Network analysis is one of the central components in computational and systems biology's aim of unravelling cellular functions in biological networks. Significant efforts

have been made to infer and analyze the structure and topology of biological networks and relate these to cellular organization and function using protein-protein interaction networks, gene regulatory networks, metabolic networks, signaling networks and neural networks. In recent years, graph theory approaches to partitioning biological networks so as to find functional modules have gained wide interest (Mitra et al. 2013). Many such algorithms have been proposed for detecting functional modules in large-scale community networks. These include the spectral partitioning method (Donetti & Muñoz 2004), the modularity optimization method (Newman 2004), the betweenness based method (Newman & Girvan 2004), which is a tightly connected component method, and graph theory approaches based on cliques (Palla et al. 2005). Most generate modular network structures without hierarchy.

In the last decade, clustering algorithms have been widely employed in the exploitation of microarray gene expression data for elucidating biological function (Shi et al. 2010; Richards et al. 2008). Cluster analysis aims to reduce the dimensions of the data by grouping genes into modules (clusters) of co-expressed genes across multiple samples with distinct biological processes, thereby helping to predict uncharacterized gene functions (Shi et al. 2010). Various studies have demonstrated how clustering algorithms can be used for the inference of biological functions (Costa et al. 2004; Richards et al. 2008), and clustering has become one of the most common methods for biological data analysis. Some regularly used clustering methods for analyzing microarray gene expression data include; K-means (Tavazoie et al. 1999), self-organizing maps (SOM) (Tamayo et al. 1999) and fuzzy c-means (Kumar & Futschik 2007). All of these are unsupervised learning approaches which partition data into clusters, i.e. divide data into a pre-defined cluster structure. These methods are not well suited for module identification, but rather reveal greater insight into the global structure of expression data (Shi et al. 2010). Furthermore, they organize genes in a flat structure without hierarchy.

Hierarchical clustering divides data sequentially to form a tree-like hierarchical cluster structure. However, the hierarchy approach has received comparatively less attention when addressing modular attributes in biological network analysis. In fact, studies have shown hierarchical architecture in transcriptional regulatory and metabolic networks, when describing hierarchical modularity based on high throughput data from simple organisms (Yu & Gerstein 2006) and higher eukaryotes respectively (Ravasz et al. 2002; Tang et al. 2012).

Notably, researchers have drawn more attention to internal validation indices - such as Silhouette width or the Dunn index - as performance measures with which to validate the accuracy of cluster modules (Fattah 2013). Internal validation indices are of great benefit when there is no prior knowledge of the genes comprising the module. Furthermore these validation indices focus more on network properties such as compactness and separation distance within modules, without much emphasis on biological function and their experimental measurements. In order to address this concern, we devised an external validation measure that uses prior biological knowledge from the gene ontology (GO) database. In particular, a new scoring system was proposed to evaluate the accuracy of modules and the performance of the networking algorithms, by examining the results of statistically significant GO enriched, biologically meaningful modular networks. These statistical scores are simple but robust aids to assist biologists in choosing biologically meaningful network modules for further investigation and corresponding suitable network algorithms.

There are many benchmark algorithms for detecting modules using gene expression data (Mitra et al. 2013). In this study, we compare three networking algorithms with the ability to detect functional modules within a hierarchical architecture and also the potential to predict regulatory interactions, selecting these from the various published algorithms mentioned in previous chapters (Table 4.1).

Table 4.1: Benchmark network inference algorithms and corresponding data types they support. A stands for steady state and B stands for time series.

Network algorithm	Method	Source	Data type	Hierarchical Network/Modularity
RedeR	Correlation	RedeR-Bioconductor package	A/B	✓
WGCNA	Correlation	WGCNA- CRAN package	A/B	✓
ARACNE	Mutual Information	Minet/Parmigene-Bioconductor package	A/B	×
CLR	Mutual Information	Minet/Parmigene-Bioconductor package	A/B	×
MRNETB	Mutual Information	Minet/Parmigene-Bioconductor package	A/B	×
BNLEARN	Bayesian	BNLEARN- CRAN package	A	×
SIMoNe	Graphical Gaussian	SIMoNe - CRAN package	A/B	✓
GRENITS	Dynamic Bayesian	GRENITS-Bioconductor package	B	×

The methods selected for comparison are weighted correlation network analysis (WGCNA) (Langfelder & Horvath 2008), RedeR (Castro et al. 2012) and statistical inference of modular networks (SIMoNe) (Chiquet et al. 2009). WGCNA and RedeR provide a simplistic way of detecting functional modules by mapping to external traits in hierarchical layout. SIMoNe explores latent structure, using mixture models to encode the modular network.

4.2 Materials and Methods

4.2.1 Datasets

The *in silico* datasets used in this study comprise subnetworks of size 100 and size 500 in *S.cerevisiae* generated from SynTReN. Both subnetworks consist of 100 samples. Refer to section 3.2.4.1.1 in Chapter 3 for more details on how the simulated datasets were synthesized.

4.2.2 Performance measurements

The ability of the network algorithms to reproduce good quality modules that are biologically meaningful are evaluated using internal and external validation measures respectively.

4.2.2.1 Internal validation

The internal validation (IV) procedure essentially focuses on evaluating the quality, or goodness, of cluster modules. This technique does not make use of any prior knowledge of gene label information when assessing modules, but combines topological statistical properties such as compactness (for example, the similarity of data points in the same module), and separation (for example, how distant data points are in different modules) assesses intra-module homogeneity and inter-module separation to compute a final score (Handl et al. 2005). These scores do not reveal biological information directly, but place emphasis on how well cluster modules are separated from each other. In this study, we used different IV measures on the basis of their ability to evaluate both inter-cluster separation and intra-cluster homogeneity for cluster modules generated from each network algorithm. Here, we selected three popular non-linear IV indices, namely Silhouette width, Dunn index and Separation index. The reason we choose these measures compared to other existing IV measures is that they are established methods for evaluating the quality of gene clusters generated from post genomic data when prior biological information is unknown (Handl et al. 2005).

Silhouette Width

Silhouette width (SW) measures are widespread amongst internal validation methods in the context of evaluating cluster modules. SW allows one to identify how similar each point (datum) lies within its own cluster compared to its neighboring clusters (Handl et al. 2005; Bolshakova & Azuaje 2003). SW also known as average Silhouette width (ASW) that aggregates individual Silhouette values, *S(i)*, measured for individual modules:

$$ASW = \frac{1}{K}\sum_{i=1}^{K} S(i) \tag{4.1}$$

Here, *K* denotes the number of cluster modules, *i* represents each point (datum) as an object in the cluster module and *S*(*i*) is calculated as

$$S(i) = \frac{b_{out}^{\min}(i) - a_{in}(i)}{\max\left(b_{out}^{\min}(i), a_{in}(i)\right)} \tag{4.2}$$

where $b_{out}^{min}(i)$ indicates the average distance between datum, *i* and all other data in the closest neighboring module and $a_{in}(i)$ denotes the average distance between datum *i* and all other data within the same module. A*SW* measures the degree of cohesion and separation of cluster modules.

It should be noted that distance in all the internal measures relates to Euclidean distance. The value of ASW ranges between -1 and +1. A higher positive value close to 1 indicates a well-clustered data where each point *i* is far from other modules, but is closer to points within its own module. By contrast, a negative value close to -1 indicates a poorly clustered data where each point *i is* far from its own module and closer to other modules. Finally, a value close to 0 suggests that point *i* is between modules. For a good quality cluster, ASW is to be maximized.

Dunn Index

The Dunn index (DI) is a popular nonlinear internal validation measure (Bolshakova & Azuaje 2003). It is calculated by taking the ratio of the minimum separation distance W_{out}^{min} between any two points from different modules and the maximum separation distance W_{in}^{max} between any two points from the same module:

$$\text{Dunn Index} = \frac{W_{out}^{\min}}{W_{in}^{\max}} \tag{4.3}$$

The value of the Dunn index ranges from 0 to ∞. A higher Dunn index reflects better cluster modules, meaning the minimum distances between points are larger in different modules compared to the maximum distance between points in the same module.

Separation Index

The Separation index (SI) is an extended version of the internal validation technique, computing for each of the points in the same module the distance to the closest neighboring points from different modules. The mean is then calculated from the smallest proportion of computed distances between modules for each of the points to estimate SI. This measurement allows us to determine the appropriate number of cluster modules. The value of the index ranges between 0 and 1, where the maximum value is an indicator of good clustering.

It should be noted that all the above internal validation measurements are calculated using the cluster.stats function in the fpc R package (Hennig 2013). The distance metric used is Euclidean.

4.2.2.2 External validation

The external validation procedure involves knowing the ground truth of the gene labels within cluster modules and their functional category in a biological context. The aim of external validation measurements is to evaluate network algorithms for their ability to determine biologically meaningful cluster modules that regulate a functional process inside the cell. In order to quantify the cluster modules and network algorithms, we propose a scoring technique that uses prior biological knowledge from the Gene Ontology (GO) database to assess how likely the group of genes within modules are to be biologically related. Therefore, these scores become particularly useful as a guide for biologists to show which modules and network algorithms are of interest for a particular dataset. The underlying assumptions made when developing the scores and its limitations are discussed briefly later in the section 4.2.2.4.

4.2.2.3 Gene Ontology Enrichment Analysis

Gene ontology[10] (GO) enrichment analysis was employed to identify over-represented GO terms using known annotations from a set of input genes from predicted modules. GO terms are a set of annotations generated from published literature which encapsulate information in a hierarchical structure with a biological process or molecular function of a gene product in the cell and the location where the function is carried out (Khatri & Drăghici 2005). The predicted modular genes provided are compared against GO terms for consistency so as to derive samples and background frequency annotations of GO terms, and this information is subsequently used to calculate *p*-values using the hypergeometric distribution function, as shown in equation (4.4):

[10] http://geneontology.org/

$$P = 1 - \sum_{i=0}^{k-1} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}} \tag{4.4}$$

Here, N represents the total of number of genes (background distribution), M the total number of annotated genes, n the number of genes in the cluster module and k the number of genes belonging to a certain functional class. The p-value provides a probabilistic measure of whether the genes found within a module are enriched in a particular GO category. There may be, for example, 20 genes in the input list of a group of modular genes obtained by performing enrichment analysis of the biological processes in the *S. cerevisiae* genome, which has ~6500 genes in the background distribution. Of the 20 input genes, only 10 are annotated to a particular GO term (say for example, a biosynthetic process) and thus the sample frequency is 10/20. If the biosynthetic process has a total of 100 genes annotated for that GO term, then the background frequency is 100/6500. A p-value close to zero indicates that a biologically meaningful result is less likely to be obtained by chance.

The GO enrichment analysis was performed using the Bioconductor package clusterProfiler (Yu et al. 2012). The modular subnetwork produced from the RedeR, WGCNA and SIMoNe algorithms were enriched for highly significant biological processes using the compareCluster function. The program performs functional enrichment using the hypergeometric distribution whilst all the other parameters are kept as default. For each gene cluster, the p-value calculated during GO enrichment analysis for biological processes was extracted for the most significant functional categories by default. The p-values were adjusted for false discovery rate (FDR) control so as to prevent false positives.

In order to compare the results derived from the GO enrichment associated with different network algorithms for same number of modules, we counted the number of GO terms that are enriched below a fixed significance threshold (p<0.05). This means that the GO terms that have a significance values less than the adjusted p-value of 0.05 are assumed likely to be associated with the modules biological process. In addition, we calculated the percentage of modules that possess at least one GO term lower than the significance threshold. This latter calculation was motivated by the possibility that some individual modules may have a sufficiently larger number of GO terms that are significantly enriched, and thus their contribution may dominate that of the other modules.

4.2.2.4 Module and Model score

In this section, we explain how we quantify modules that are biologically relevant and that relate to a particular functional process, following GO enrichment analysis. In order to quantify the modules, we proposed modular and model scores which incorporate the statistical significance values of gene modules that are over-represented for biological processes during the GO enrichment analysis. These modular and model scores will be particularly useful when evaluating individual modules and network algorithms respectively, with the aim of extracting biologically meaningful information.

4.2.2.4.1 Assumptions of the score

The assumptions that have been made when developing the score are discussed below.

1. Before developing the score one of the fundamental questions asked was: if a list of genes within a module, say M1, is significantly (p<0.05) enriched for more than one biological function, which one of these is actually biologically relevant? For example, from Figure 4.2 below, we can see that out of 8 modules identified by SIMoNe from

size 100 expression data, only 4 modules (M1, M2, M7 and M8) are significantly enriched.

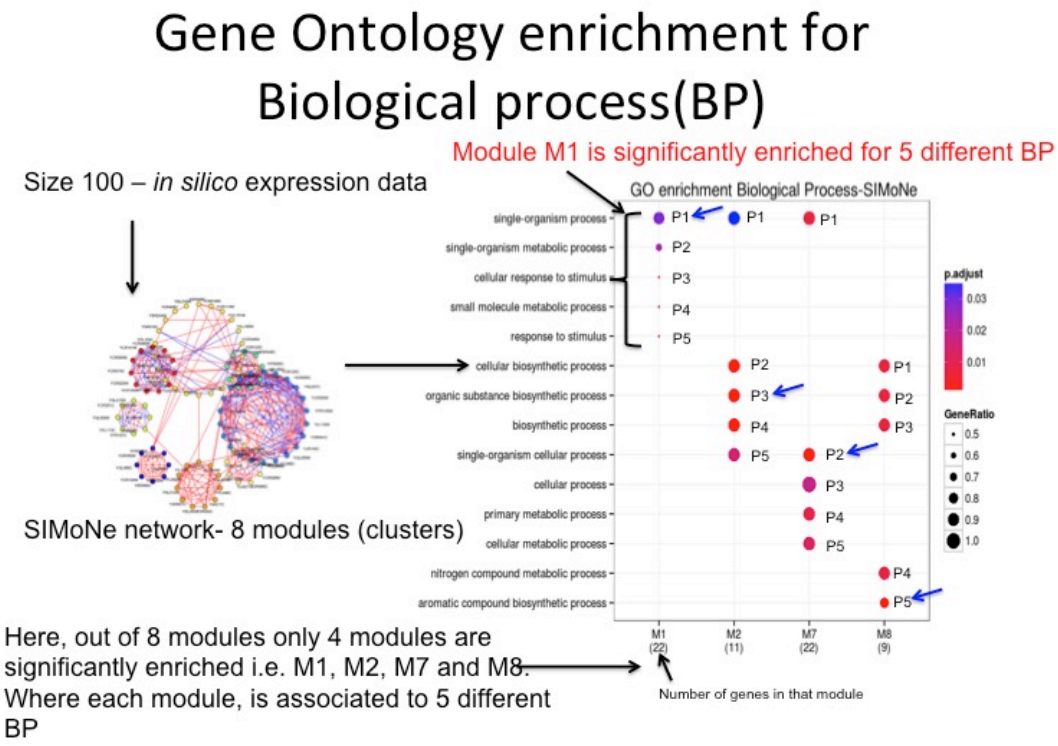


Figure 4.2: The flow process of GO enrichment analysis for each identified module**.** The color key reflects the significance scale. Low *p*-values (red) indicate high enrichment and high *p*-values (blue) indicate low enrichment. The dot sizes for each category relates to gene ratio (GR), where GR is expressed as decimal fraction in the key. GR is the ratio of the total number of genes within a module that are associated with a BP in GO enrichment analysis to the number of genes that are associated with a particular module (e.g. in module, M1 (27), 27 indicate total genes associated with M1).

For simplicity, we only show 5 significantly enriched biological processes (BP) for each of these modules. For instance, looking at module M1 which is associated with 5 different biological process with significance values (P1,P2,..., P5) - which BP is closely associated with M1? In order to address this concern and avoid any implicit

bias, we choose the lowest *p*-value, which is indicated by the solid blue arrows in the figure. The underlying assumption is that selecting the BP with the lowest *p*-values is more likely, by probability, to be associated with biological reality than any other BP

2. The magnitude of the module and model score is a reliable indicator of the closeness of a list of genes within a module to a biological function. Here the magnitude of the score ranges between 0 and infinity. The assumption is that a higher module score relates to a more significantly biologically associated functional module. Whereas, with lower module score, the probability that a particular module identified groups of genes that perform specific biological function *in vivo* is less likely. The model score is a scoring function which aggregates modular scores from each subnetwork to reflect the overall biological activity of all other modules within the sub-networks, and to examine the performance of the network algorithm.

4.2.2.4.2 Limitations of the score

1. Although the module and model score are empirical and quantify the biological activity of cluster modules using ground truth from GO databases, further experimental investigation is required to support these measures.
2. The scores do not reveal information on the transcriptional program within the modules. To overcome this limitation, these module scores can be coupled with targeted network analysis to determine their transcription factor (TF) activity.

4.2.2.4.3 Model score

The accuracy of the model in describing modular attributes was estimated using a quantitative module score $M(k)$ described in equation (4.5) for a particular module k that has been assessed to be enriched for N biological processes using an adjusted p-value threshold of 0.05:

$$M(k) = P^*_{\max} = \max\left(P^*_1,\ldots,P^*_N\right) \tag{4.5}$$

Here,

$$P^*_k = -\log_{10}\left(P_k\right),$$

where $\{P_1,P_2,\ldots,P_N\}$ are the significance p-values for the N enriched processes. Furthermore, the modular score $M(k)$ is extended to give the model score described by equation (4.6)

$$Model\ score = \frac{\sum_{j=1}^{K} M(k)}{K}, \tag{4.6}$$

where K represents the number of modular clusters produced by the network algorithm. The flow process for calculating biologically meaningful module and model scores is illustrated in Figure 4.3.

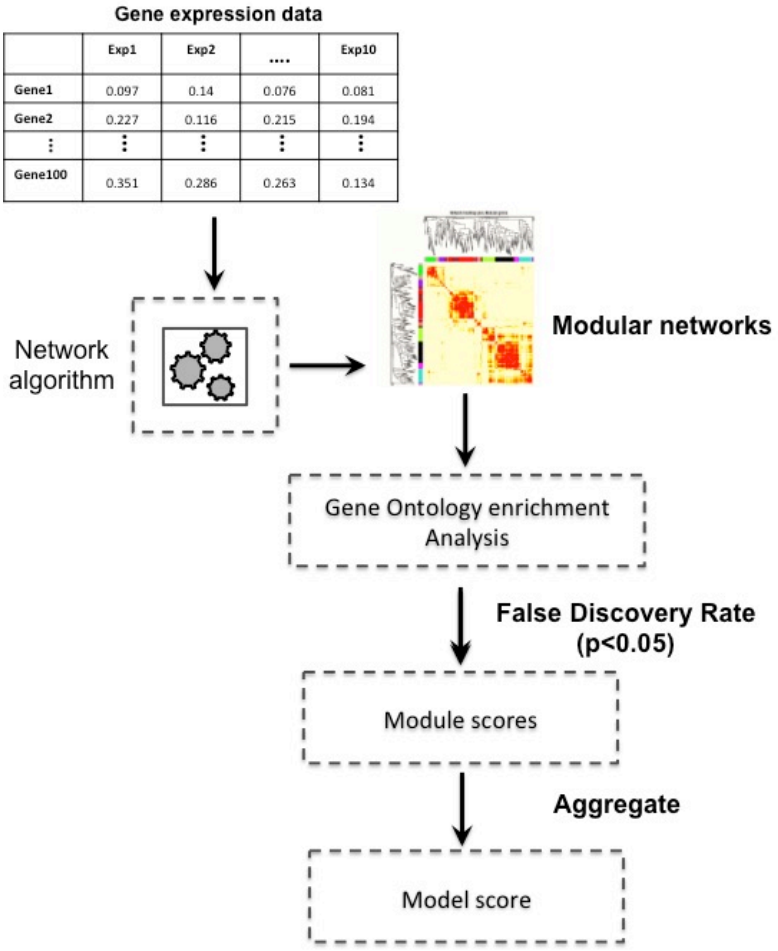


Figure 4.3: The workflow for deriving module and model scores.

4.2.3 Results & Discussion

In this section, we reconstructed modular networks from benchmark network algorithms using *in silico* gene expression data for networks of size 100 and size 500, generated from SynTreN (Refer Chapter 3 for more details). For unbiased predictions, we generated different number of cluster modules by varying the parameters (e.g. clusters.qmin/qmax in SIMoNe to adjust cluster size) for comparative analysis. Specifically, for a given dataset, we used each network algorithm to generate modular gene networks with 4, 8, 12 and 16 subnetworks by adjusting the clusters.qmin/qmax parameters in SIMoNe and cutting the hierarchical tree at different heights in RedeR and WGCNA in such a way that the desired sub-networks

generated fairer, comparative analysis. Refer to the Methods section in Chapter 2 for more details on how the subnetworks were generated.

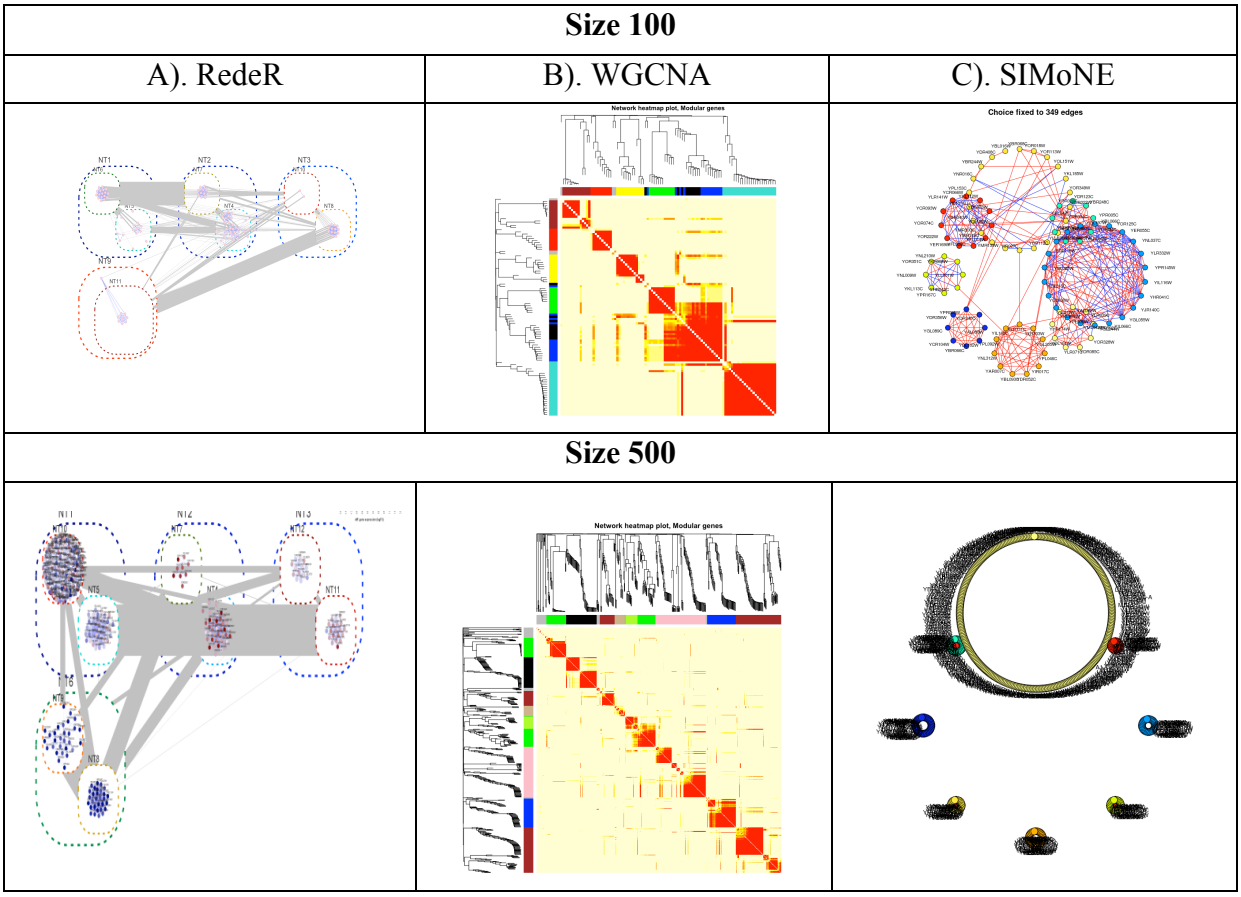


Figure 4.4: Hierarchical and modular network consisting of 8 modules with size 100 and size 500 gene expression data. A). RedeR. The modules are clustered in a hierarchical structure. B). WGCNA. The red box plots across the diagonal represents modules. C). SIMoNe. The clusters produced determine the modular network.

The networks with modular structures, reconstructed using different network algorithms, are depicted in Figure 4.4. For brevity and illustration purposes, we show only 8 modules (clusters) derived from the size 100 and size 500 *in silico* gene expression datasets. Likewise, we reconstructed networks containing various numbers of modules, ranging from 4 to 16 with a step size of 4, to address any existing bias for further analysis (figures not shown). Although these benchmark algorithms produce modular networks, it is imperative to evaluate the quality of these modules. Here, we refer to the quality of the cluster modules,

how well they are separated and how they concomitantly contain meaningful biological information encoded within the network. Before measuring the quality of modules for biological information, we evaluate the quality of the cluster modules using internal validation indices.

Internal validation measurements

The internal validation index quantifies the quality of cluster modules, taking into account intra-module homogeneity and inter-module separation distances. These measurements allow the network algorithm to be assessed for its ability to generate well-separated modules. It should be noted that these measures do not place emphasis on biological function, but focus mainly on the quality of modules. In order to evaluate the quality of cluster modules obtained using each network algorithm, we calculate the Average Silhouette width (ASW), the Dunn Index (DI) and Separation index (SI) using both medium sized (size 100) and large sized (size 500) gene expression data. For more details on these indices, refer to the Methods section.

The internal validation measurements were compared against a different number of modules, predicted using each of the networking methods as shown in Figure 4.5. It can be seen from the plots that by increasing the number of modules, the compactness of these modules increases as indicated by the ASW and DI values. However, the SI decreases when we increase the number of modules. This trend was quite evident with RedeR and WGCNA for both size networks, although the decrease in SI for RedeR with larger size data was not as steep. For the lowest number of modules tested (4), SIMoNe gave the best results overall; however, as the number of modules was increased, the performance of SIMoNe deteriorated, as indicated by DI for both size networks. By contrast, the performance of SIMoNe as assessed by ASW, increased slightly when the number of modules was set to 8 for a medium

size network (100), but with the same number of modules, its performance dropped for larger size networks (500). From these measures, it was concluded that the performance of RedeR was relatively better, as its measurements were consistent without many dramatic changes with module number.

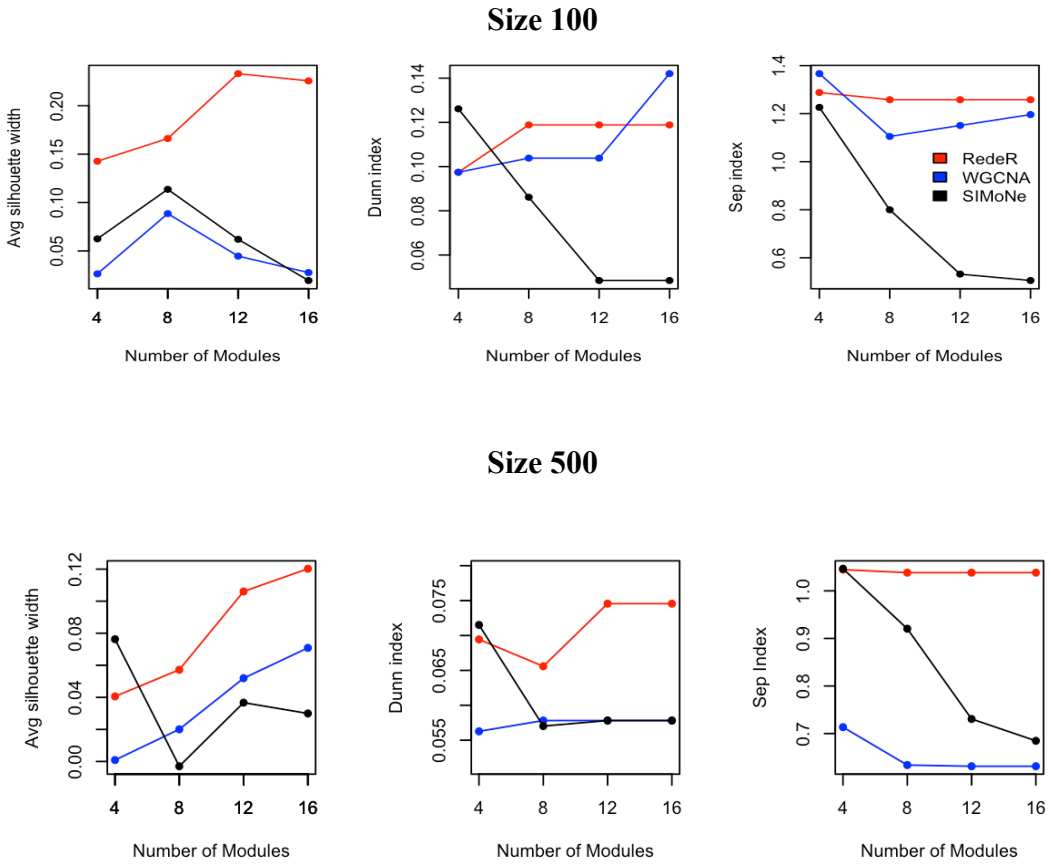


Figure 4.5: Average silhouette width, Dunn index and Separation index calculated for different numbers of cluster modules generated from each of the network algorithms using *in silico* gene expression data of size 100 (top plots) and size 500 (bottom plots).

Gene Ontology enrichment analysis for Biological process

We observed that in the GO enrichment analysis, there were several GO terms attached to each cluster module at various statistical thresholds. Here, the analysis of the first set of GO terms was examined by counting the number of terms that are over-represented in modules and calculating the percentage of modules that had at least one statistically enriched GO term

at several statistical thresholds. The results obtained from the preliminary analysis are presented in Figure 4.6. From these plots, it is apparent that the modules identified using WGCNA for the size 100 network contain more enriched GO terms at several significance thresholds, compared to those identified using the other two network algorithms. SIMoNe also performed well with 8 modules. However, with size 500 data, there was a strong overlap in the results for all module numbers, across all network algorithms. The percentage of modules that had at least one statistically significant enriched GO term at several significance thresholds is shown in Figure 4.7. From this figure, it appears that more enriched GO terms for size 500 networks are identified at several cutoffs with SIMoNe. What is interesting to see is that with 4 modules, the percentage of modules containing at least one enriched GO term reaches a maximum value of 100 at a threshold *p*-value of 0.05. For size 500, although there was consistency found with 12 and 16 modules, RedeR and WGCNA performed well with a lower number of modules.

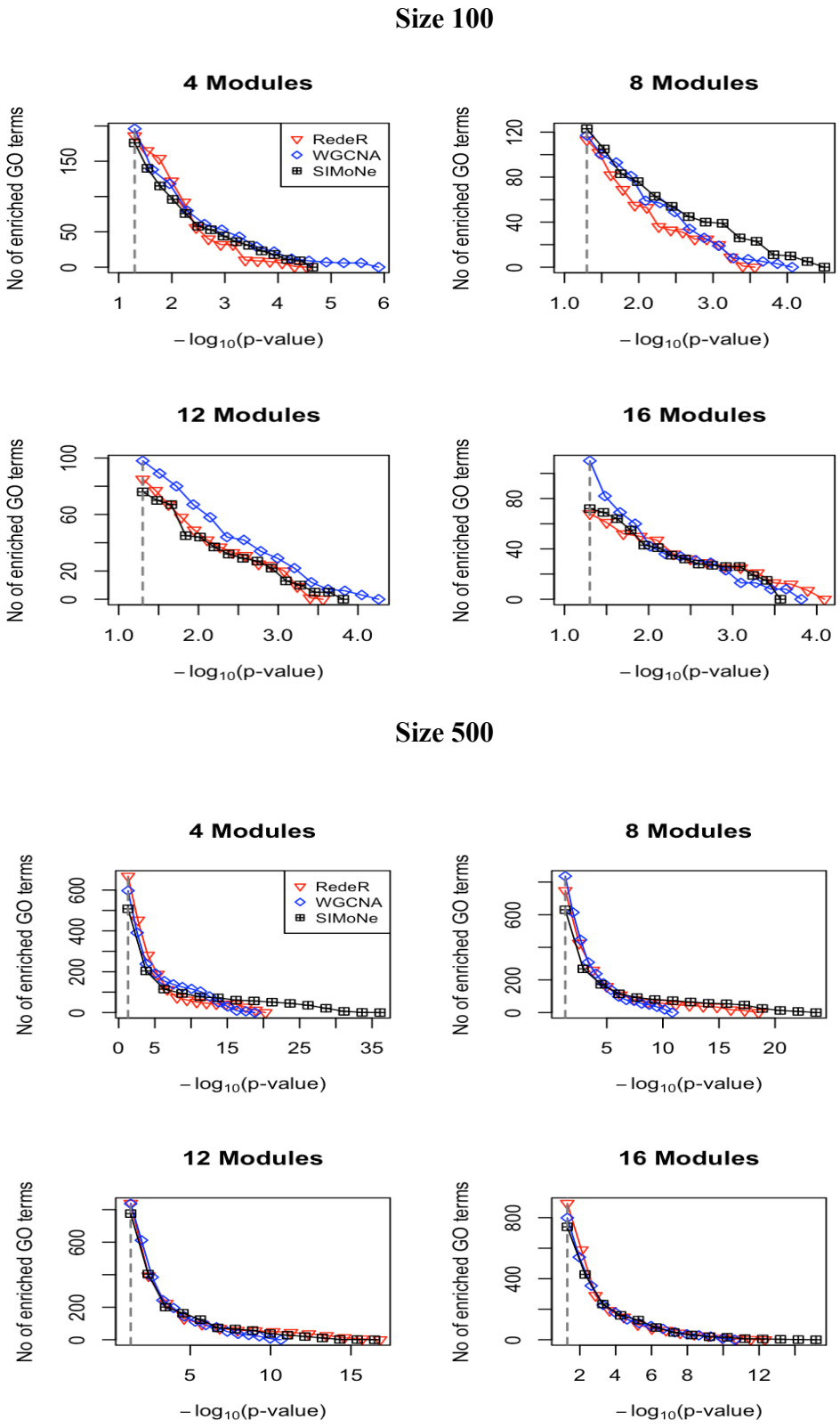


Figure 4.6: Number of enriched GO terms found for different number of modules generated from size 100 and size 500 *in silico* datasets at various *p*-values cutoffs. The dashed line in each plots indicates the critical significance threshold (p<0.05). *p*-values are plotted using the –log10 scale.

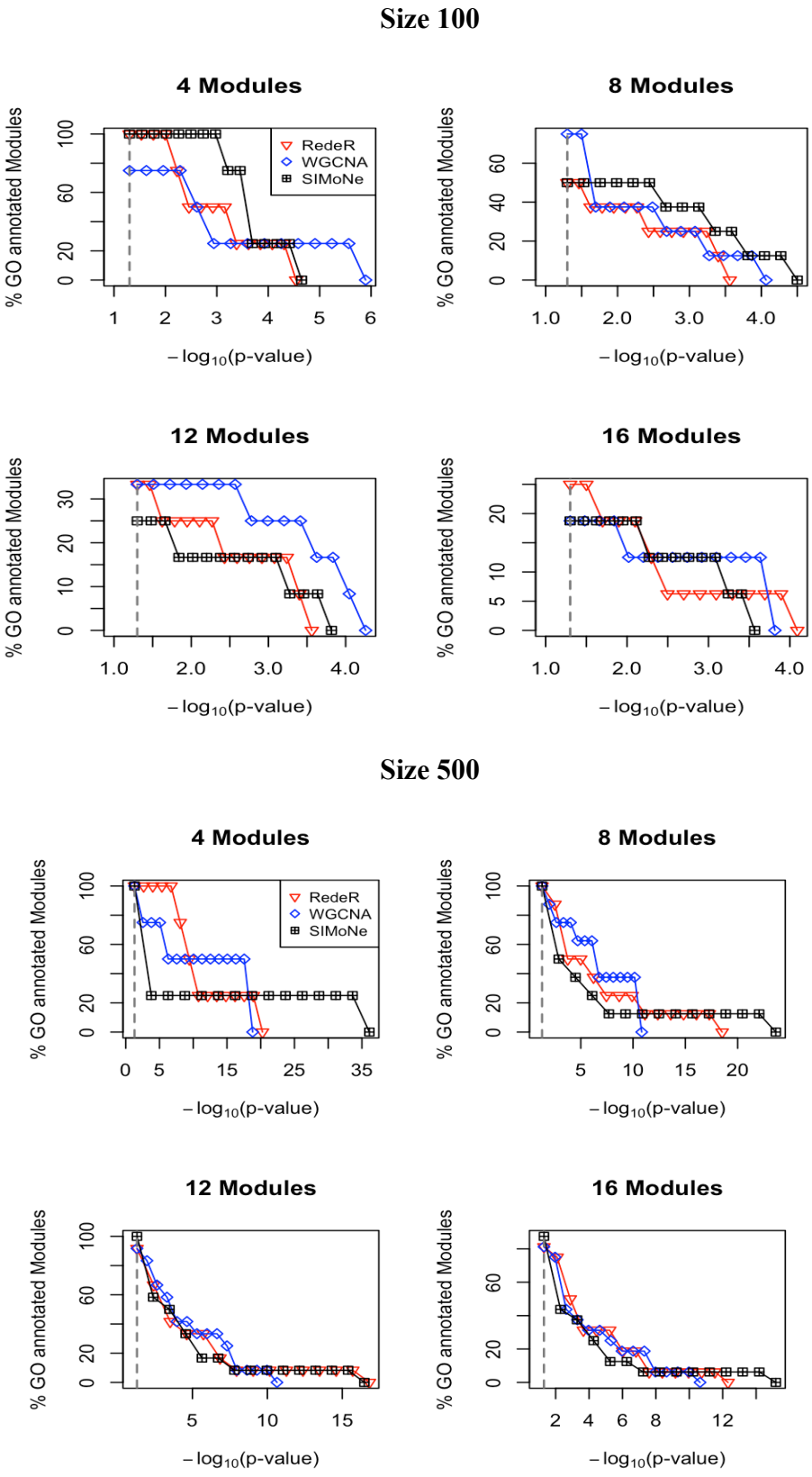


Figure 4.7: Percentage of annotated GO terms found for different numbers of modules generated from size 100 and size 500 *in silico* datasets at various *p*-value cutoffs. The dashed line in each plots indicates the critical significance threshold (p<0.05). *p*-values are plotted using the –log10 scale.

In Figure 4.8 we show the top 5 statistically significantly (adjusted p<0.05) enriched for biological processes (BPs) associated with each module from different network algorithms with size 100 data for illustrative purposes. The modules that are not statistically significantly enriched (adjusted p<0.05) are not shown in the plot. Here, we fixed the number of cluster modules to 4 for comparative analysis. In the x-axis of each plot we see the module number (for e.g. M1) with the number of genes under that module given within the brackets (e.g. M1 (31) denotes 31 genes predicted in module M1). The y-axis shows the corresponding biological processes associated with the module, represented by colored dots. The dot size and color denotes the gene ratio and its adjusted significance p-value respectively, with the corresponding color key given on the right hand side of each plot. One can see from Figure 4.8 that out of 4 cluster modules derived from each network algorithm, WGCNA shows only 3 modules (M1, M2, M3) that are significantly enriched (p<0.05) for BPs. However, all 4 modules generated by RedeR and SIMoNe show significant association with BPs. For RedeR, many modules are found to be associated with single organism cellular processes (i.e. BPs involving only one organism). However, WGCNA predicts modules M1 and M2 to be strongly associated with the cellular process. Here, cellular process refers to physiological cellular process that is associated with cell growth (or maintenance). The association of SIMoNe with BPs appears to be consistent with that of RedeR, except that many of its modules are related to primary metabolic process.

When the same dataset (size 100) was used to identify 8 cluster modules from different network algorithms, the results were consistent with the results for 4 modules as shown in Figure 4.9. From this figure, we observe that of the 8 modules identified by RedeR and SIMoNe, only 4 cluster modules are significantly enriched for BP. Notably, M1 from both algorithms is enriched for metabolic processes, whereas M2 and M8 are enriched for biosynthetic processes and M7 for single organism cellular processes. By contrast, for

WGCNA, it was interesting to note that module M1 was identified with the cellular response to stimulus, while modules M4 and M5 were enriched for small molecule metabolic processes and single organism processes respectively. Other modules were consistent with the results for RedeR and SIMoNe. It is also interesting to note that modules M4 and M5 from WGCNA are enriched for single biological processes, unlike other modules which are associated with more than one BP. Similarly, when the same data (size 100) was examined with 12 cluster modules (Figure 4.10), only 4 modules predicted from RedeR and WGCNA were found to be enriched for different BPs, appearing to show consistent results. Nevertheless, only 3 cluster modules (M2, M3, M11) were enriched for different BPs from SIMoNe. Finally, with 16 cluster modules (Figure 4.11) the results were similar to those for 12 cluster modules, suggesting that even when increasing module numbers, the modules showed consistent enrichment analysis results with those obtained with for lower module numbers.

With size 500 data, the enrichment analysis results demonstrated were very similar across different network algorithms. Furthermore, when the number of modules was fixed to 4, this notably improved the analysis, with many of the cluster modules showing strong overlap of BPs (Figure 4.12). In particular, the enrichment analysis identified most statistically significant association with single organism and organic substance metabolic processes in at least two of the network algorithms. However, SIMoNe identified biosynthetic processes in module M2. Correspondingly, with the number of modules set to 8 (Figure 4.13), the results observed were consistent with those for 4 modules, with the exception of the nitrogen compound metabolic process identified in module M5 by RedeR and SIMoNe and in M8 by WGCNA. It is worth mentioning that as the number of modules was increased to 12 and 16, as shown in Figure 4.14 and Figure 4.15 respectively, the gene lists were more accurately stratified to reveal a particular BP within a cellular process. Overall, the

enrichment analysis results were found to be consistent across different network algorithms.

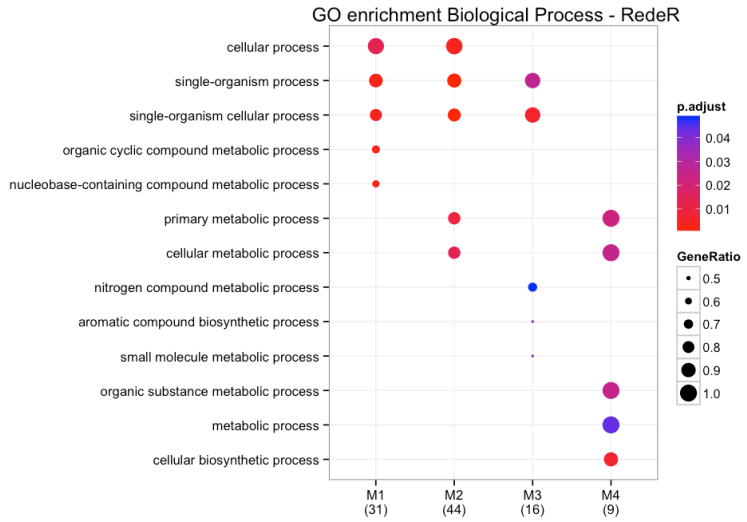
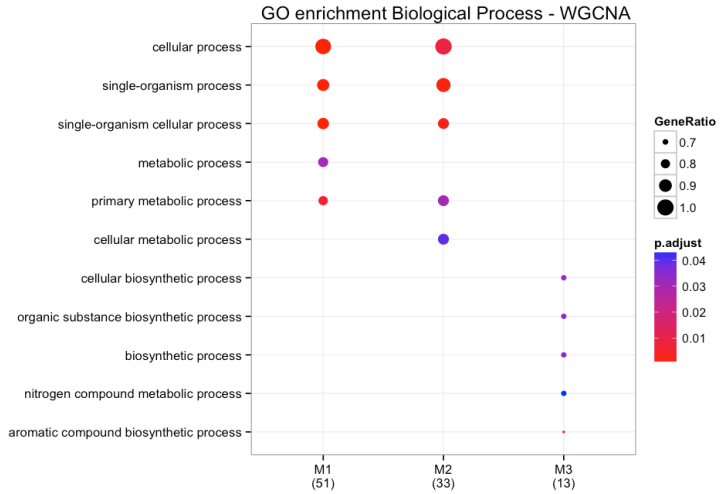
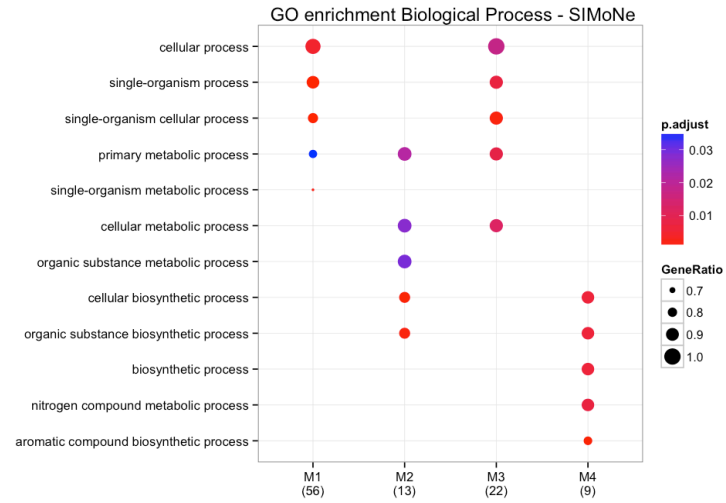
Size-100, 4 Modules	
RedeR	
WGCNA	
SIMoNe	

Figure 4.8: Gene ontology enrichment analysis for 4 cluster modules that are significantly enriched for biological processes with size 100 data. The dot size denotes gene ratio, and color indicates significance *p*-values.

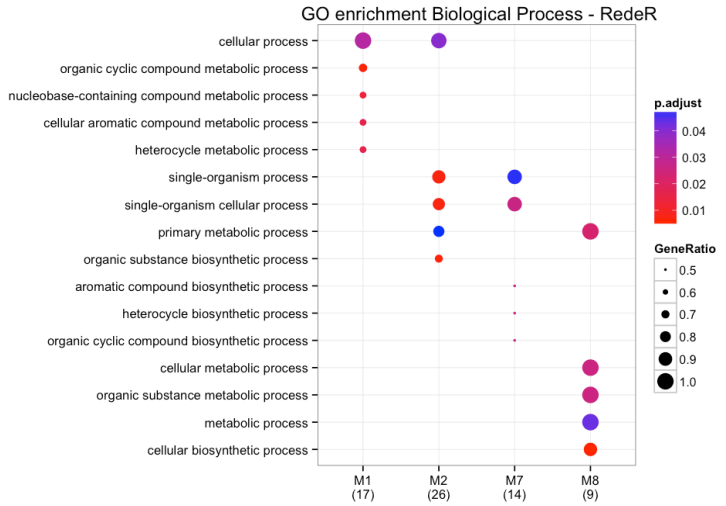
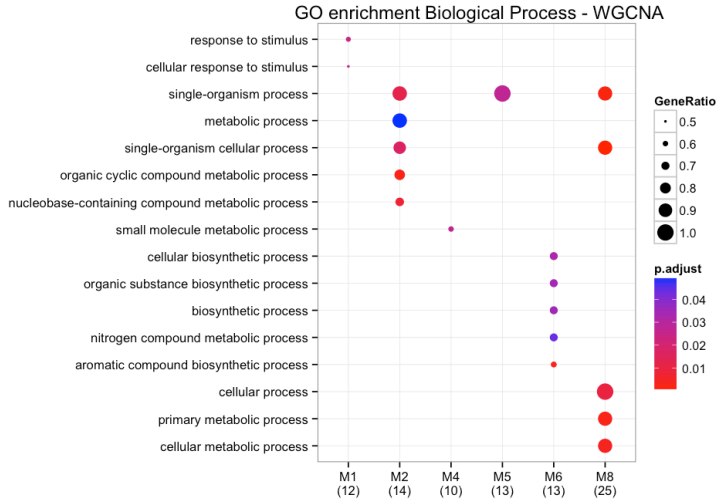
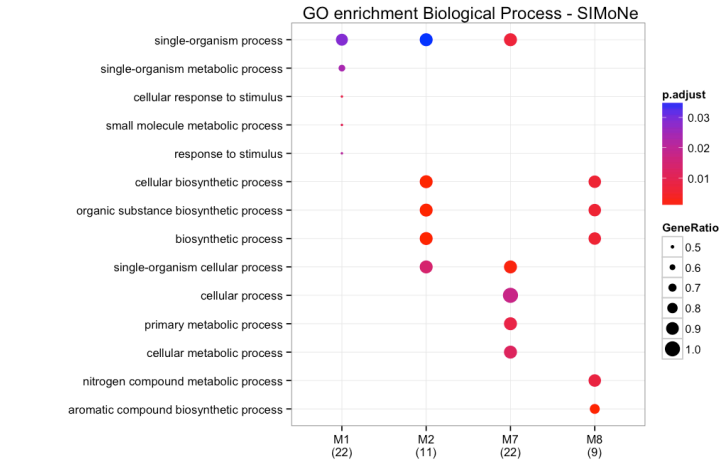
Size 100, 8 modules	
RedeR	
WGCNA	
SIMoNe	

Figure 4.9: Gene ontology enrichment analysis for 8 cluster modules that are significantly enriched for biological processes with size 100 data. The dot size denotes gene ratio and color indicates significance *p*-values.

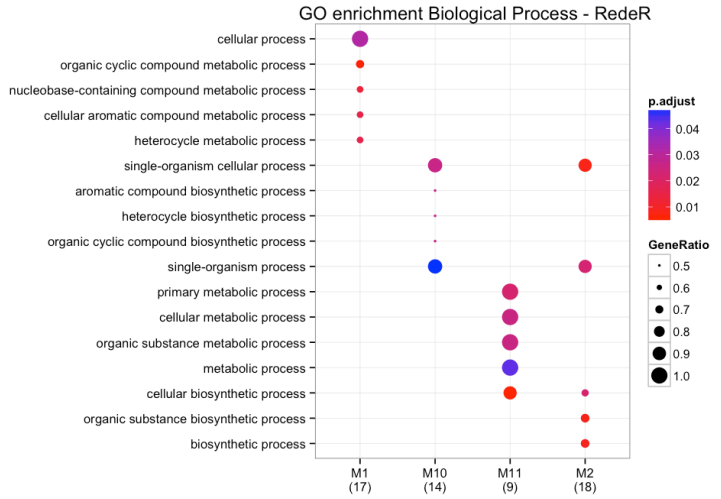
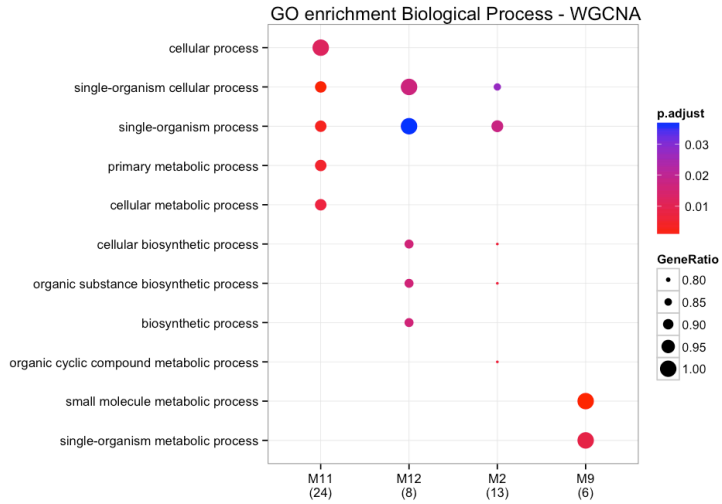
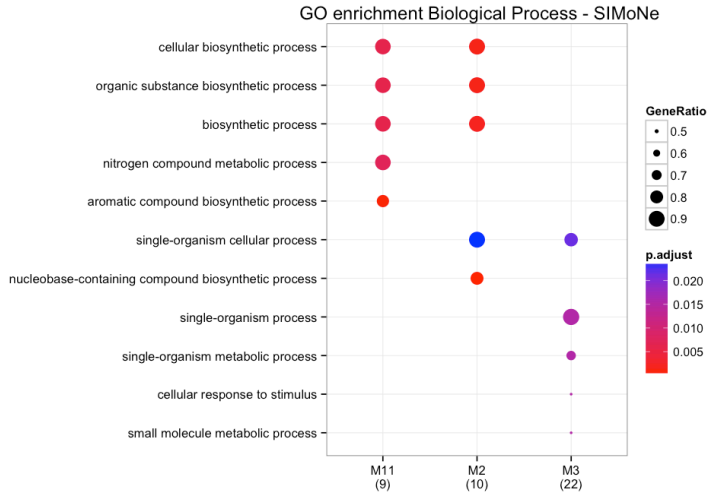
Size 100, 12 modules	
RedeR	
WGCNA	
SIMoNE	

Figure 4.10: Gene ontology enrichment analysis for 12 cluster modules that are significantly enriched for biological processes with size 100 data. The dot size denotes gene ratio and color indicates significance *p*-values

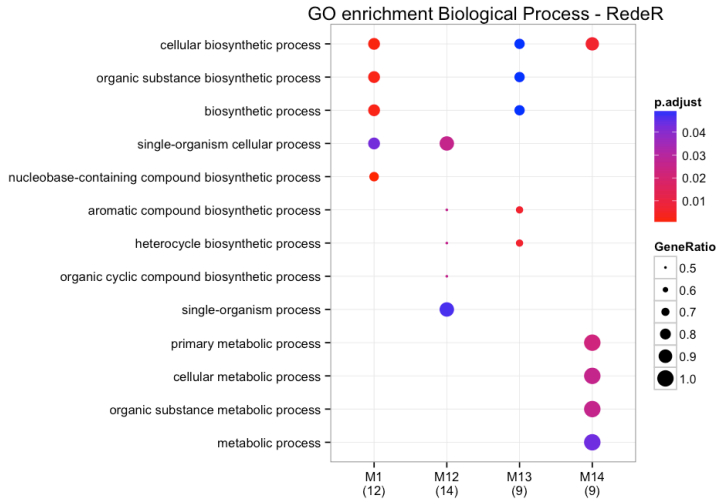
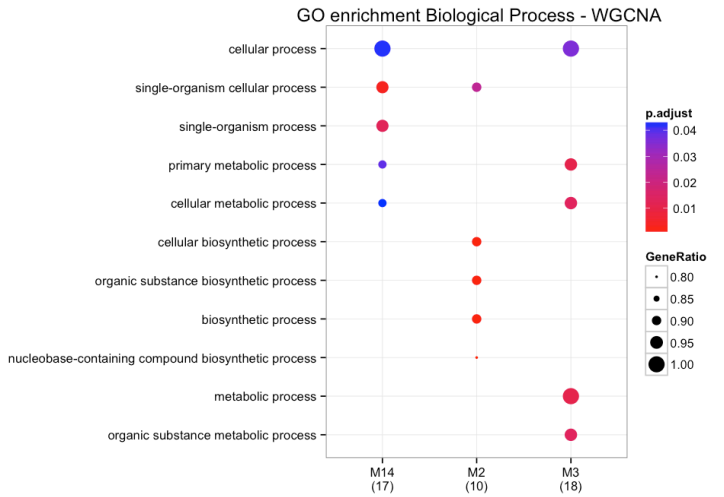
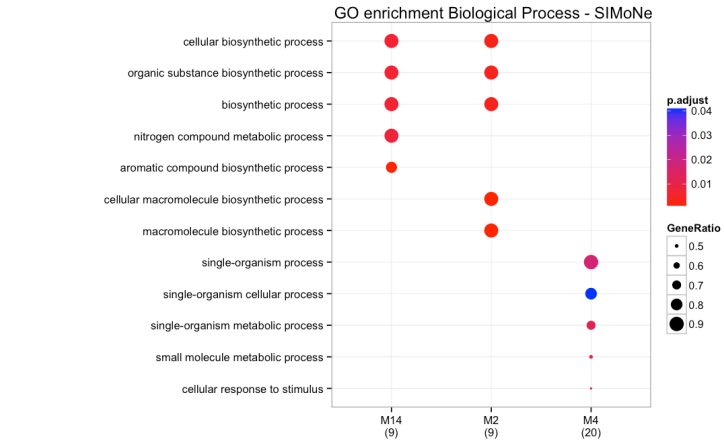
Size-100, 16 Modules	
RedeR	
WGCNA	
SIMoNe	

Figure 4.11: Gene ontology enrichment analysis for 16 cluster modules that are significantly enriched for biological processes with size 100 data. The dot size denotes gene ratio and color indicates significance *p*-values.

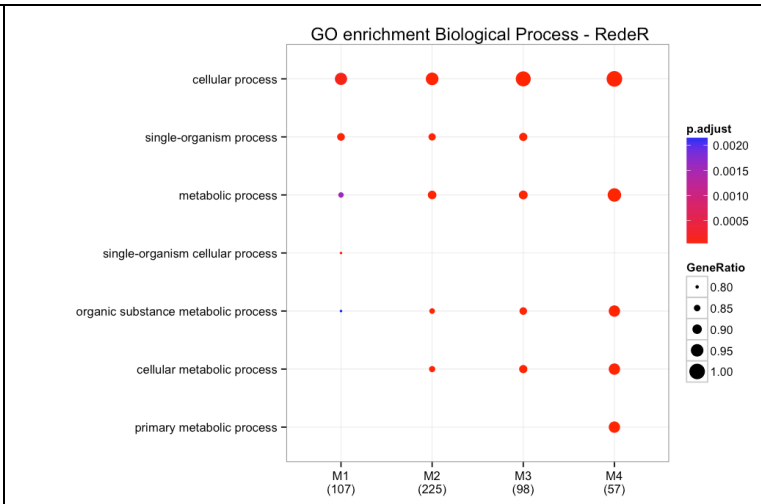
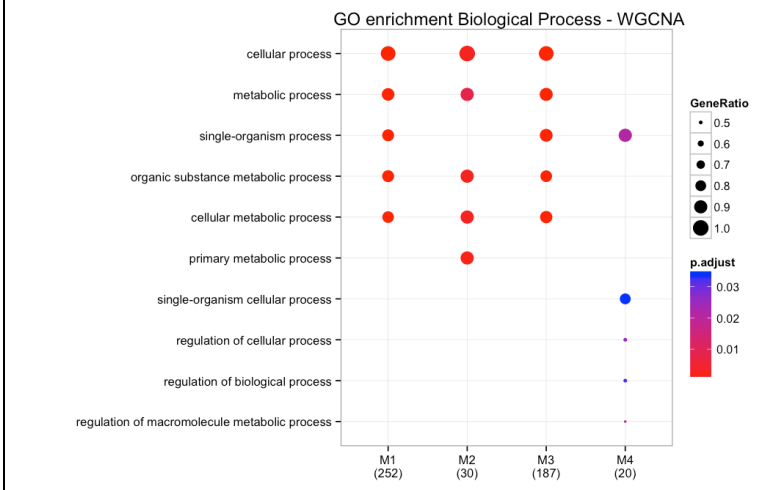
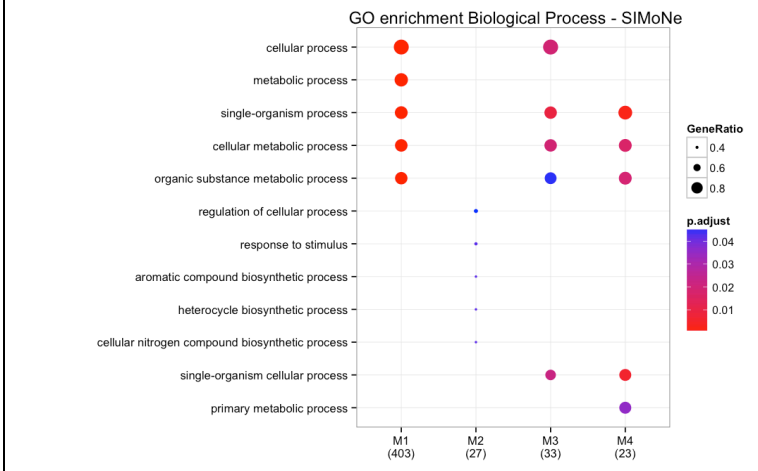
Size 500, 4 Modules	
RedeR	
WGCNA	
SIMoNe	

Figure 4.12: : Gene ontology enrichment analysis for 4 cluster modules that are significantly enriched for biological processes with size 500 data. The dot size denotes gene ratio and color indicates significance *p*-values.

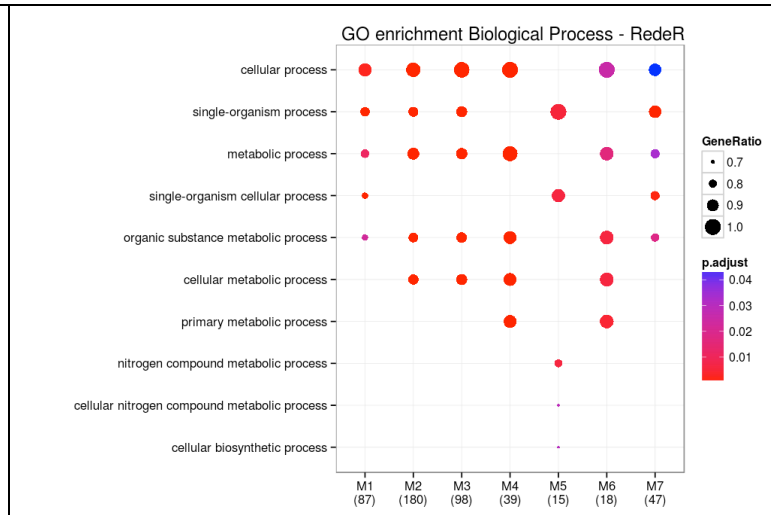
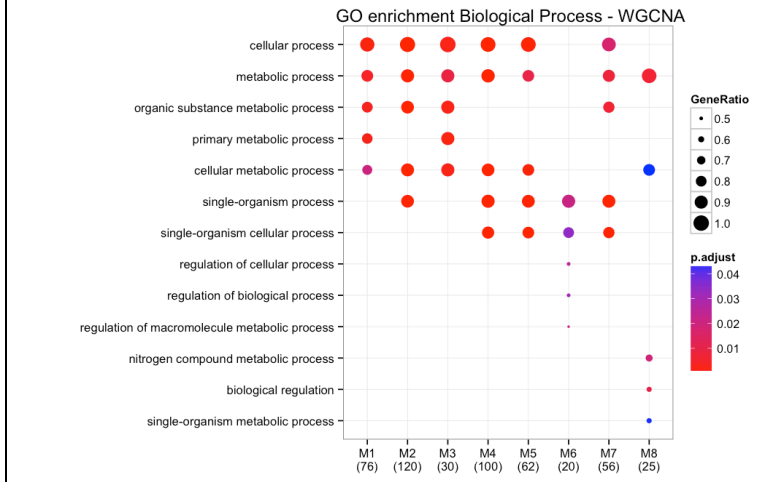
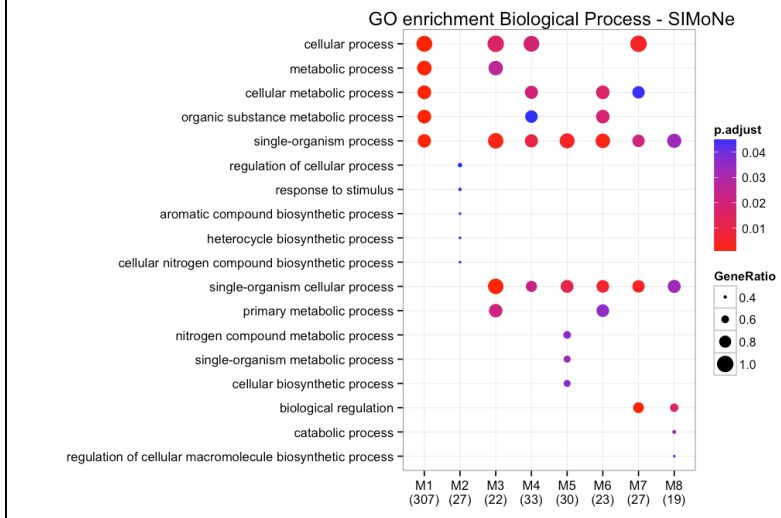
Size 500, 8 Modules	
RedeR	
WGCNA	
SIMoNe	

Figure 4.13: Gene ontology enrichment analysis for 8 cluster modules that are significantly enriched for biological processes with size 500 data. The dot size denotes gene ratio and color indicates significance *p*-values.

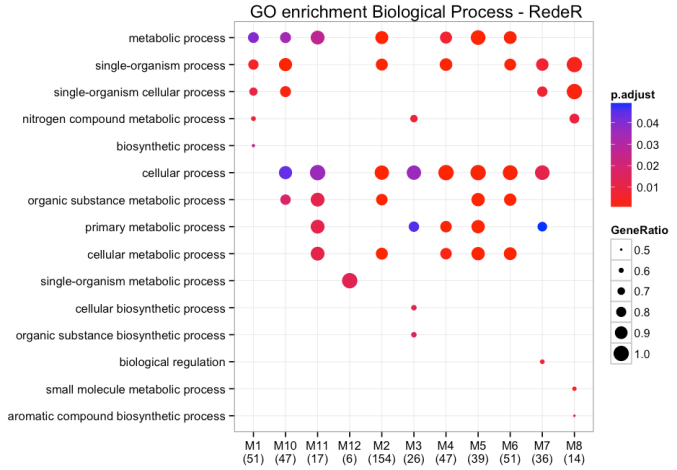
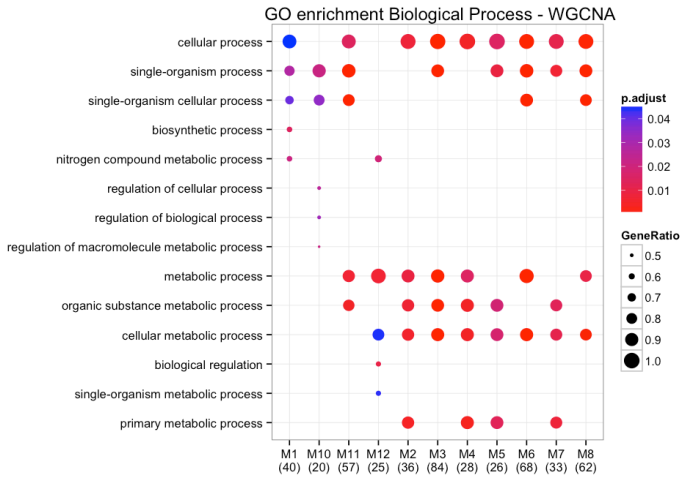
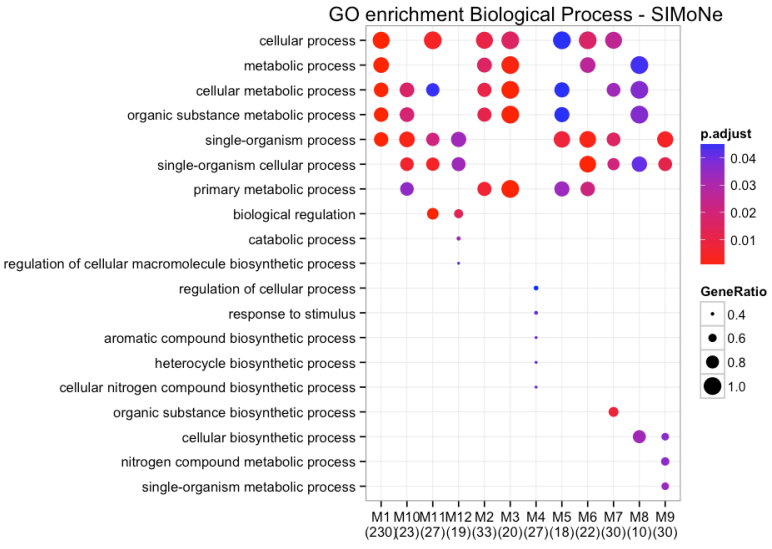
Size 500, 12 Modules	
RedeR	
WGCNA	
SIMoNe	

Figure 4.14: Gene ontology enrichment analysis for 12 cluster modules that are significantly enriched for biological processes with size 500 data. The dot size denotes gene ratio and color indicates significance *p*-values.

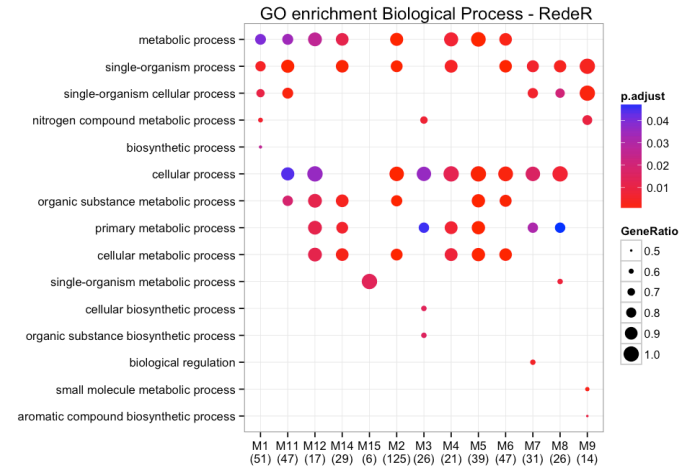
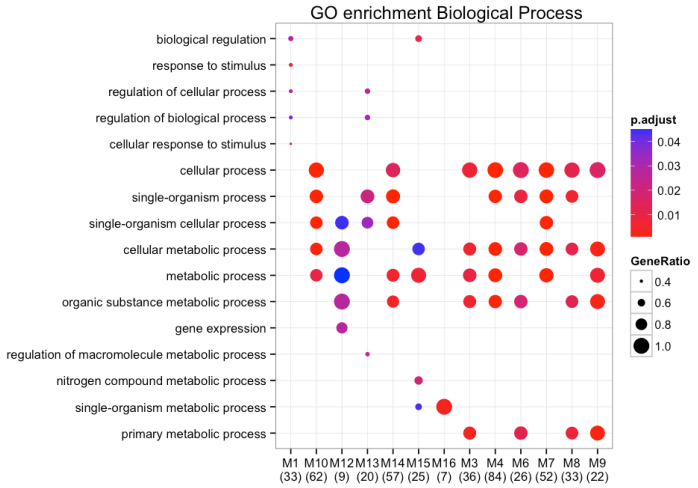
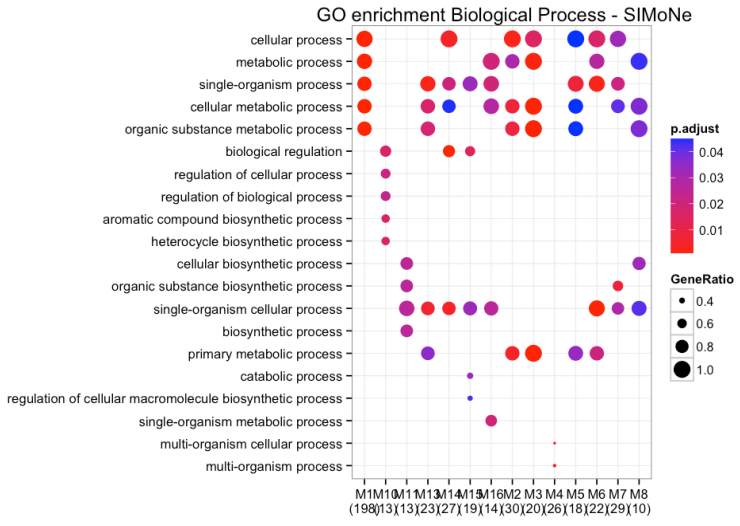
Size 500, 16 Modules	
RedeR	
WGCNA	
SIMoNe	

Figure 4.15: Gene ontology enrichment analysis for 16 cluster modules that are significantly enriched for biological processes with size 500 data. The dot size denotes gene ratio and color indicates significance *p*-values.

In summary, the results from the GO enrichment analysis revealed that a particular module from any network algorithm appears to have a statistically significant (p<0.05) to association with more than one BP. Therefore, these results require a decision to be made regarding which biological process is more likely to be associated with a particular cluster module. In order to facilitate this decision making process, we have therefore developed a simple scoring system known as module score that can quantify which module is more likely to be associated with a particular biological process, and a model score that quantifies the ability of network algorithms to reveal biologically meaningful results. See the Methods section for more details on how the module and model scores are calculated and the assumptions made whilst developing these scores. These measurements will help biologists to focus on a particular BP using module scores and subsequently model scores will help them to evaluate the quality of the network algorithm in order to provide biologically plausible results. It should be noted that these measurements act as a guide in order to focus direction on a particular biological process in contrast to other significantly enriched BPs.

External validation measures

In this section, we will focus our analysis on external validation measures by comparing module and model scores for different modules generated across different network algorithms. Figure 4.16 compares the module and model scores[11] obtained with different network algorithms for size 100 data. In the plot, the x-axis represents cluster module number and the y-axis represents module/model score plotted on a $-\log_{10}$ scale. The module and model scores are color coded in blue and red respectively. A higher module score indicates that the module is more significantly enriched for a biological process and thus provides biologically plausible results.

[11] See the Methods section for details on the calculation of module and model scores.

The results obtained from this analysis show consistency across the different numbers of cluster modules generated. However, looking closely, it is evident from this figure that for RedeR, reconstructing 4 modules provides biologically meaningful results, as indicated by the individual module scores which are relatively higher than those obtained for other module numbers, encapsulated by the high model score of 2.024. Likewise, a similar trend was observed with WGCNA, where reconstructing a network with 4 modules yields better individual module scores. This is because as we reduce the number of cluster modules, the number of genes within the modules grows relatively larger and yields a more statistically significant association with a biological process during the GO enrichment analysis. In contrast, SIMoNe displayed better results as the numbers of modules was increased. This could be possibly due to the number of genes being stratified to a lower number, as then the likelihood of association with a particular biological process increases. The modules that are consistently enriched for a biological process with a lower significance value yield a better module score. The better performing modules are those that yield better model scores. Here, RedeR performed best with the number of modules fixed to 4, WGCNA with 12 modules and SIMoNe with 16 modules. The annotations associated with the best performing modules are illustrated in Table 4.2. The table describes module numbers (for example, M1 denotes module 1), associated GO terms that show better significance values, corresponding module scores and counts. By counts is meant the number of genes within a module that are associated with its corresponding BP. It is apparent from this table that RedeR and WGCNA shows similar GO term associations, namely single organism cellular and metabolic processes, whereas SIMoNe show more association with a cellular biosynthetic process.

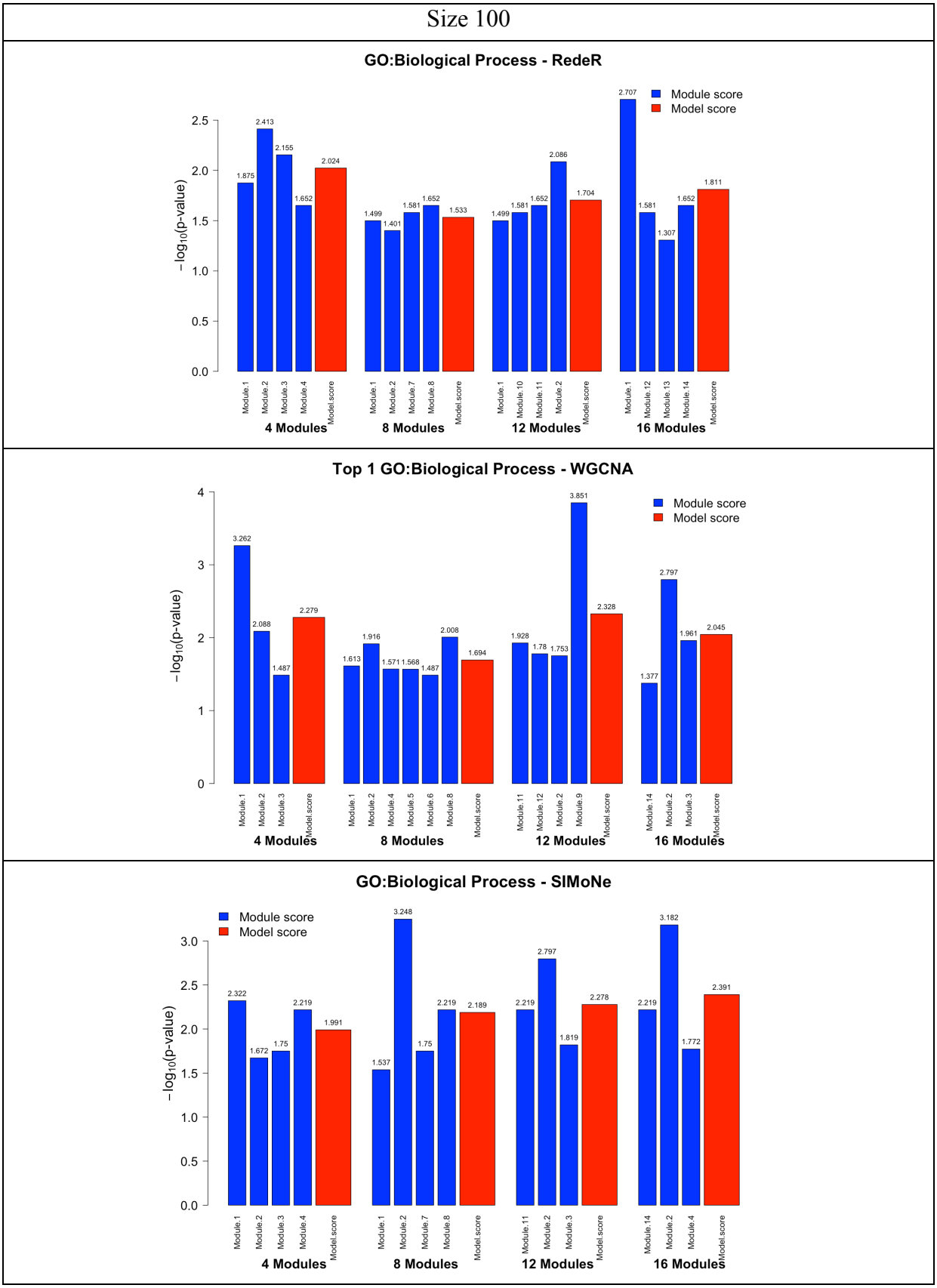


Figure 4.16: Module and model scores for the top enriched GO terms for different numbers of modules with size 100 data.

Table 4.2: Functional top ranked modules from different network algorithms that show statistically significant (p<0.05) association to biological process in GO enrichment analysis for size 100 data. Count signifies total number of genes within a module that are associated with a BP.

Module	GO.ID	Biological Process	Count	P-value	Module score
		RedeR			
M1	GO:0009987	cellular process	30	0.013328	1.875
M2	GO:0009987	cellular process	43	0.0038657	2.413
M3	GO:0044763	single-organism cellular process	15	0.0069951	2.155
M4	GO:0044238	primary metabolic process	9	0.022303	1.652
		WGCNA			
M11	GO:0009987	cellular process	24	0.011809	1.928
M12	GO:0044763	single-organism cellular process	8	0.016608	1.78
M2	GO:0044699	single-organism process	12	0.017671	1.753
M9	GO:0044281	small molecule metabolic process	6	0.00014107	3.851
		SIMoNe			
M14	GO:0044249	cellular biosynthetic process	8	0.0060416	2.219
M2	GO:0034645	cellular macromolecule biosynthetic process	8	0.00065753	3.182
M4	GO:0044699	single-organism process	18	0.016891	1.772

Similarly, when examining module and model scores with size 500 data, the results showed complementary trends to those for size 100 data, as presented in Figure 4.17. One can see that as the number of modules increases, the module scores and associated model scores tend to decrease. This trend was observed in at least two of the network algorithms. It should be noted that some of the modules showed very high module scores, in particular M1. For example, M1 showed a high module score when associated with the cellular process from GO analysis. This was consistent amongst all the network algorithms tested, suggesting that for large datasets, the algorithms show similar biologically meaningful results.

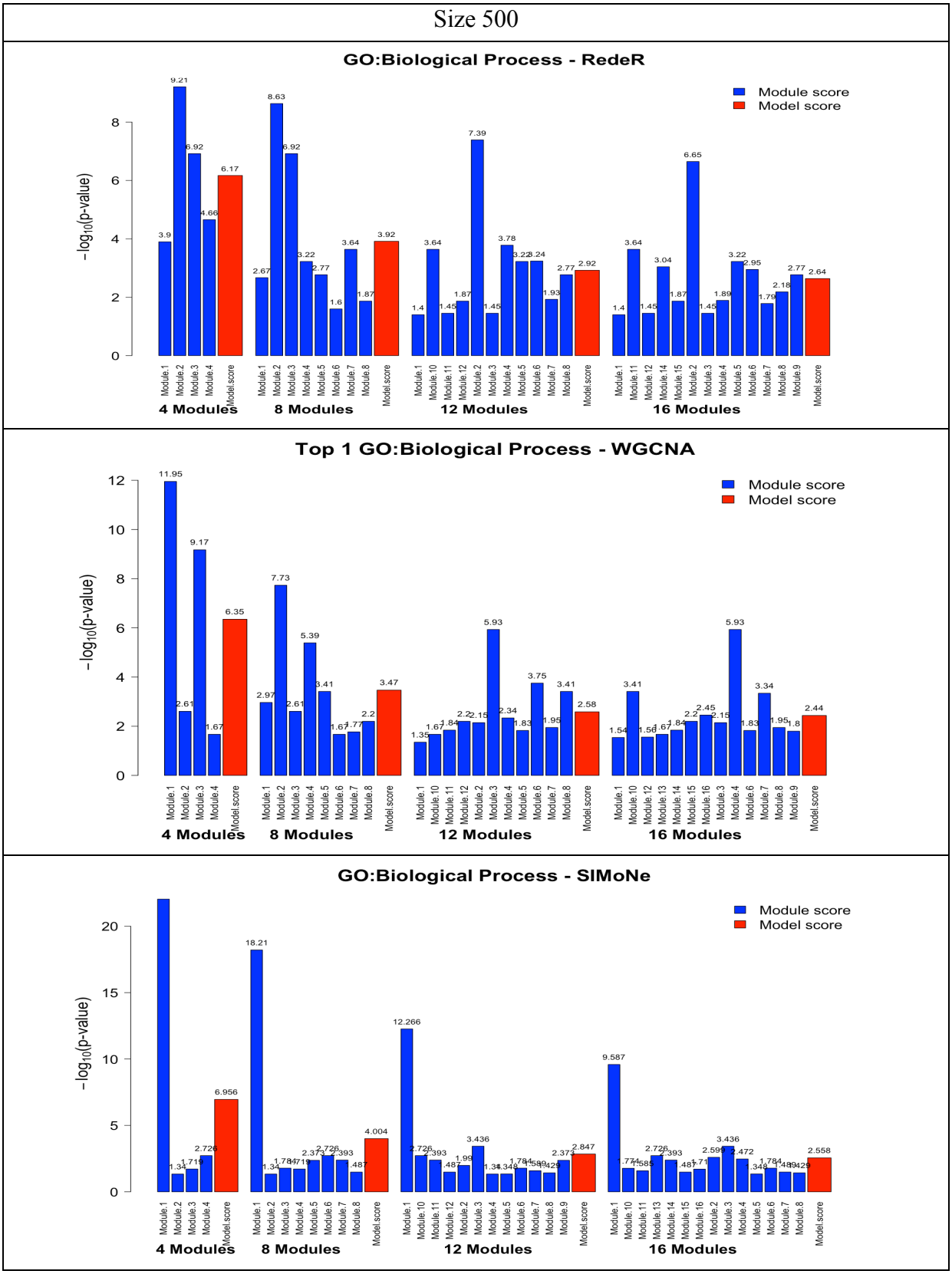


Figure 4.17: Module and model score for top enriched GO term found for different number of modules with size 500 data.

Table 4.3: Functional top ranked modules from different network algorithms that show statistically significant (p<0.05) association to biological process in GO enrichment analysis for size 500 data. Count signifies total number of genes within a module that are associated with a BP.

Module	GO.ID	Biological Process	Count	P-value	Module score
		RedeR			
M1	GO:0009987	cellular process	101	0.00012642	3.898
M2	GO:0009987	cellular process	214	6.18E-10	9.209
M3	GO:0009987	cellular process	97	1.21E-07	6.918
M4	GO:0009987	cellular process	57	2.20E-05	4.657
		WGCNA			
M1	GO:0009987	cellular process	242	1.12E-12	11.952
M2	GO:0009987	cellular process	30	0.0024823	2.605
M3	GO:0009987	cellular process	180	6.70E-10	9.174
M4	GO:0044699	single-organism process	18	0.021332	1.671
		SIMoNe			
M1	GO:0009987	cellular process	390	9.14E-23	22.039
M2	GO:0050794	regulation of cellular process	12	0.045754	1.34
M3	GO:0009987	cellular process	32	0.019077	1.719
M4	GO:0044699	single-organism process	21	0.0018785	2.726

The annotations associated with the best performing modules for size 500 data are illustrated in Table 4.3. As we can see from the table, many of the modules from different algorithms are associated with cellular processes, and for SIMoNe, one such module gives the highest module score (22.03).

The model scores comparing the performance of the network algorithms at several module numbers that determine biologically meaningful results are shown in Figure 4.18. From this figure, we can see that for medium sized networks (size 100), the performance of RedeR and WGCNA show similar trends, but the magnitude of the scores varies. In contrast, the performance of SIMoNe improved when the number of modules increased. It should be noted that the performance of WGCNA was found to be the best when the number of modules was fixed at 4 and 12, but ranked second for the other module numbers tested. In a

similar way, SIMoNe performed best when the number of modules was fixed at 8 and 16. Further comparative analysis with larger size networks (size 500) are displayed in the right column of Figure 4.18, showing a consistent trend of decreasing model score with increasing module number across all the network algorithms. In contrast to the results obtained with size 100 data, SIMoNe performed best with the number of modules fixed at 4 and 8. The scores from RedeR, however, were slightly better compared to the other algorithms when the number of modules was fixed at 12 and 16.

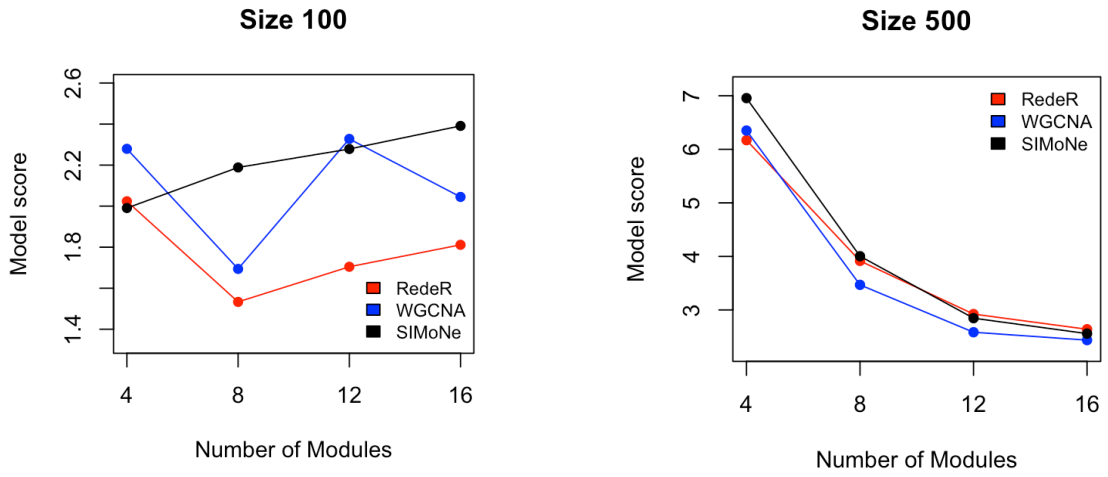


Figure 4.18: Model scores obtained using different network algorithms for various numbers of modules with size 100 (left) and size 500 (right) data.

Overall, these results indicate that depending on the dimension of the dataset and the number of modules fixed by thresholding, the performance of the network algorithms in determining biologically significant modules varied. This was particularly apparent with medium size (size 100) networks, but was also observed with large size network (size 500). This type of variation in the performance of network algorithms was observed in measuring the goodness of clustering for all the internal validation indices used, (i.e. Average Silhouette width, Dunn index and Separation index). Comparing the internal validation indices (Figure 4.5) against the model scores (Figure 4.18) show inconsistent trends across different number

of modules for both datasets (size 100 and size 500) on any network algorithm used. From this comparison two suggestions can be drawn. First, it suggest that the higher internal validation measure that relates to better quality cluster module may not necessarily show significant association to biological process. Second, the internal validation and the model scores are not biased towards a particular network algorithm. Furthermore, the performance and accuracy of algorithms can be improved to produce functional modular networks dependent on the complexity of the dataset and biological enrichment category. In this study, we used significance values (*p*-values) of highly enriched biological categories to evaluate the algorithms, rather than estimating percentage of gene coverage within a module that is associated with an enriched biological process (i.e. the number of genes per module involved in a BP to the total number of genes in that module) that were used in previous studies (Richards et al. 2008). We believe that using the percentage of GO enriched genes would show bias towards a particular network algorithm. Alternatively, the implementation of *K*-fold cross-validation coupled with external rand index (RI) assessment to evaluate the accuracy of network algorithm for generating functional modules has been useful from time series gene expression data (Costa et al. 2004). However, external validation RI requires prior knowledge of classification of the genes (module) given an expression dataset. RI evaluates the accuracy of the algorithm by identifying similarity between known and predicted modules. Drawback of this approach is that it focuses on prior biological knowledge which is limited in reality.

4.2.4 Conclusions

The present study compares three networking algorithms in the context of their ability to identify biologically meaningful hierarchical modular networks. A traditional approach to evaluating the quality of cluster modules derived from network algorithms is the use of

internal validation indices. In this chapter, we used the popular Average Silhouette Width, the Dunn Index and the Separation Index to identify the quality of modular networks. Although an internal validation index helps to evaluate how well the modules are separated, it does not indicate its relevance to biology. We therefore developed and proposed external validation measures (module and model score) that quantify biologically meaningful cluster modules predicted by any network algorithm. Although the module and model scores are simple, they provide robust measures for identifying biologically meaningful results. These scores depend on Gene Ontology (GO) enrichment analysis, being based on the assumption that groups of genes within cluster modules that are over-represented for a particular biological process are more likely to show biologically meaningful results. Depending upon the number of modules fixed by thresholding, the dimension of the datasets used, and the internal complexity of each individual network algorithm, there was a varied performance when determining true modular networks. For instance, RedeR outperformed the other algorithms (when the. number of modules was fixed at 12 and 16) in identifying true modular networks for biological processes with large size networks (500 genes) - but ranked last with medium size networks (size 100 genes). Similarly, with larger size networks (500 genes), SIMoNe outperformed other algorithms in identifying true modular networks for biological processes (when the number of modules was fixed at 4), but ranked last with medium size networks (100 genes). Nonetheless, these measurements will facilitate future research in the downstream analysis of true modular networks of choice, in order to investigate their transcriptional program further. Alternatively, this leads us to investigate the relationship between modular networks. Although these network algorithms are able to address modular attributes, they also have the potential to predict biologically relevant modules from real gene expression data, which we will investigate in the next chapter.

Chapter 5

Application of consensus approach to study yeast network

Abstract

In this chapter, the proposed consensus approach (by FCPT), module score and model score is applied to the yeast network study, in particular the highly developed eukaryotic model organism S.cerevisiae using real microarray data. Five frequency-based network inference algorithms were combined using statistically significant values attached to network edges to produce a quantitative consensus network. The constructed consensus network was evaluated for performance by using sensitivity and specificity to test for edge coverage in yeast interactions from a curated database, which was experimentally verified. The quantitative consensus network was compared against qualitative consensus networks, dynamic Bayesian networks, random networks and other individual networking methods. The results demonstrate that quantitative consensus networks predict many real biological interactions with high accuracy, and outperform other methods. In addition to the consensus network analysis, we further examined modularity of networks using module score and model score to obtain biologically meaningful results.

5.1 Yeast network

S. cerevisiae is a highly developed eukaryotic model organism for genetic, pharmacological and biochemical studies. The yeast network has been the subject of intensive study and most of the network components have been well characterized (Gutteridge et al. 2010; Gasch & Werner-Washburne 2002; Harbison et al. 2004). Taking into account its well-defined simple eukaryotic characteristics, yeast networks are an attractive model for systems biology research, yielding results that can be later applied to higher eukaryotes (Petranovic et al. 2010). Genome-wide identification of regulatory interactions measuring molecular activity still remains a challenge, though. Many, previous studies have focused on finding gene interactions from subnetworks of model organisms (Steele & Tucker 2008; De Smet & Marchal 2010). Here, we present an application of our consensus approach, using it to identify targeted gene interactions in yeast (*S. cerevisiae*) from real time course (dynamic) gene expression data.

5.2 Biological Data

The biological data used for building the consensus network was the time series microarray gene expression dataset (GSE22832) downloaded from the publicly available Gene Expression Omnibus (GEO) repository[12]. The microarray data was already normalized using the Robust Multichip Average (RMA) method. The gene expression data was obtained from *S. cerevisiae* cultures grown on a glucose-limited media, after a shift in oxygen (O_2) concentration from 20.9% to fully anaerobic conditions. Seven time points sampled at (0, 0.2, 1, 3, 8, 24, 79 hr) for 20.9% O_2 were included in our study. Sampling data comprised two biological replicates, i.e. two expression values for each time point, each corresponding to

[12] http://www.ncbi.nlm.nih.gov/geo/.

one biological replicate. Further experimental specifications can be found in Rintala *et al* (Rintala et al. 2011).

5.2.1.1 Significance Analysis

The differential expression (DE) of genes was analyzed for consecutive time differences, in contrast to the standard approach of comparing each time point to the control (t0) (e.g., t0.2-t0,t1-t0.2,t3-t1,t8-t3,t24-t72 rather than t0.2-t0,t1-t0,t3-t0,t8-t0,t24-t0,t72-t0) (Rintala et al. 2011; Morandi et al. 2008) as illustrated in Figure 5.1. The idea of using consecutive time points in differential expression analysis was to encapsulate the dynamics of gene expression changes across the sampling time. The union of DE genes across the individual sampling points $\{DE_1 \cup DE_2 \cup \ldots \cup DE_n\}$ was considered in the analysis, where $DE=t_{i+1}-t_i$ and t and n signify the individual time points and number of time points respectively.

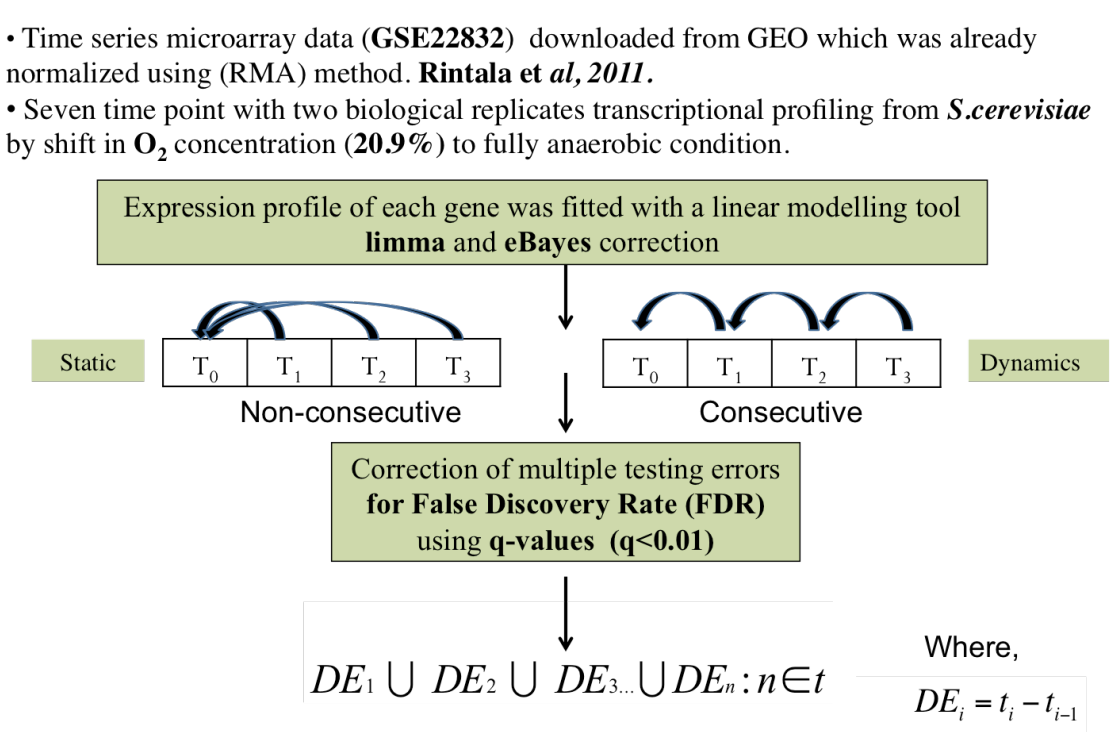


Figure 5.1: Schematic depicting the gene selection process by differential gene expression analysis, contrasting consecutive comparison against non-consecutive comparison.

Statistically significant differential expressed genes (DEGs) were identified using the limma package, a linear modeling tool (Smyth 2005). A linear model was fitted to each gene, using the least squares method with the empirical Bayesian (eBayes) applied to calculate differential expressions within experimental conditions across pairs of consecutive time points (Smyth 2004). The eBayes step approximates the average variability across all genes and also adjusts low and high variable genes up and down respectively. To enable the correction of multiple testing errors, the *q*-values were computed using bootstrapping from *p*-values generated from eBayes for corresponding genes, to control false discovery rate (FDR) estimation (Dabney A, Storey JD 2013). The *q*-value is defined as an FDR analogue of a *p*-value, providing a measure of the proportion of false positives incurred in a particular test, which are said to be significant. A cutoff *q*-value of 0.01 was applied for differential gene expression analysis. The analysis was performed in R/Bioconductor version 2.14.2 (R Development Core Team (2011) 2011).

5.2.1.2 Gene Selection

Significance of microarray data analysis for the expression dataset (20.9% O_2) designed for consecutive time points is represented in Figure 5.2A. There are 329 DEGs found in the dataset after taking the union of DEGs across individual time points (0.2 hr to 79 hr). In addition, we investigated differential genes for non-consecutive time points, where each of the time points were compared against the control time (t0) (i.e. t0.2-t0, t1-t0, t3-t0, t8-t0, t24-t0, t72-t0): this yielded 1828 DEGs. The breakup of DEGs from both the approaches (i.e. consecutive and non-consecutive) is shown in Fig 5.2A and 5.2B respectively. The Venn-diagram shown in Figure 5.2C illustrates the number of differential genes that are common and uniquely expressed, by applying consecutive and non-consecutive approaches.

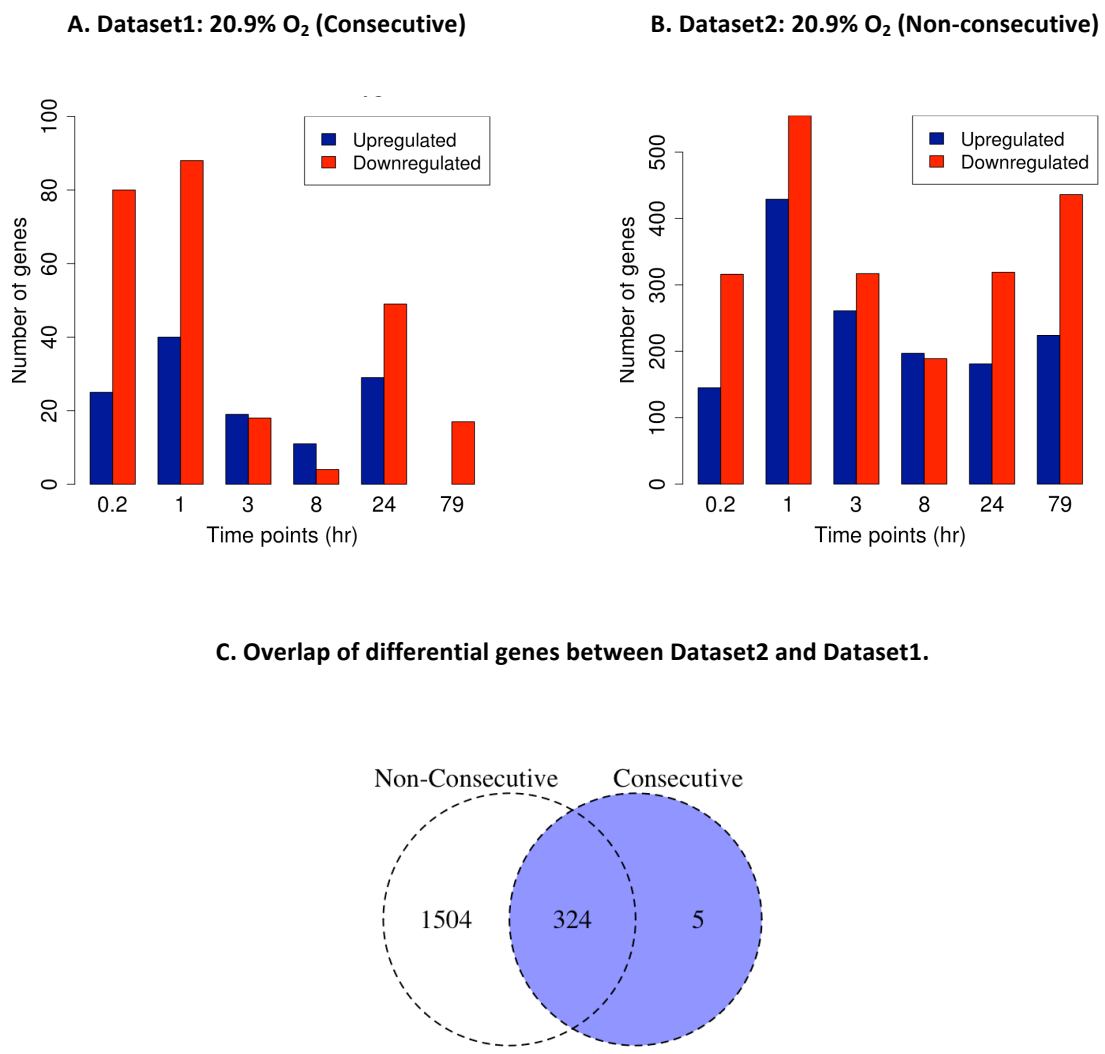


Figure 5.2: Statistically significant DEGs (q<0.01) changing across consecutive and non-consecutive time points. A) Consecutive differential genes that are up- and down-regulated for dataset1 (each time point was compared to the previous time point). Blue bars signify up-regulated genes and red bars signify down-regulated genes. B) Non-consecutive differential genes for dataset2 (each time point was compared to the control time point t0). C) Venn diagram showing the number of differential genes that are common from both consecutive and non-consecutive approaches. The genes highlighted in blue are used in this study.

From both these approaches, 324 genes were found to be consistent, while 5 genes and 1504 genes were uniquely identified using consecutive and non-consecutive approaches respectively. However, the analysis of Rintala *et al.*(Rintala et al. 2011) revealed 3811 DEGs

at significance (p<0.01). The acute drop in the number of DEGs in our analysis to 329 for dataset1 can be attributed to the inclusion of q-values to control a false discovery rate at significance q<0.01 and the method of comparing each time point to the previous one (consecutive). This seems to suggest that there is a less significant change in gene expression profiles across successive time points compared with the change relative to the initial time point (non-consecutive). In this study, we use the former (consecutive) approach to select genes to study consensus networks, so as to encapsulate the dynamics changes in genes at each of the time points.

5.3 Network Validation

In order to validate the consensus network model, we compared our results with known yeast interactions extracted from the curated Saccharomyces Genome Database (SGD) (Cherry et al. 2012). The database consists of 339,346 interactions, of which 205,865 are genetic interactions (GIs) and 133,481 are physical interactions (protein interactions) (PIs). If we take each interaction from the SGD database to be an experimentally verified edge, the performance of each networking method was evaluated for edge coverage by examining the Receiver Operating Characteristic (ROC) curves that summarise the relationship between *Sensitivity* - or True Positive Rate - and False Positive Rate (1-*Specificity*) at several critical significance thresholds. Furthermore, the performance of individual and consensus methods, were quantified by calculating AUROC[13] measures as discussed in section 3.2.5 of Chapter 3.

In each predicted network, an edge is a true positive (TP) if it corresponds to an experimentally verified genetic interaction (GI) or physical interaction (PI):

[13]The scoring methodology to evaluate the performance of consensus networks and individual algorithms is consistent as indicated in Chapter 3. The true (target) network interactions were extracted from the curated Saccharomyces Genome Database (SGD).

$$\mathrm{TP} = E(G)_{GI} \bigcup E(G)_{PI} \tag{5.1}$$

In equation (5.1) above $E(G)_{GI}$ and $E(G)_{PI}$ represent the set of predicted edges corresponding to real GIs and real PIs respectively. A false negative (FN) is an edge that does not correspond to an experimentally verified interaction. A false positive (FP) is an edge that is present in a predicted network and absent in experimentally verified interactions, whilst a true negative (TN) is an edge that is absent in both. Furthermore, self-edges were included in the edge selection strategy while calculating performance measures. The network validation schematic is illustrated in Figure 5.3.

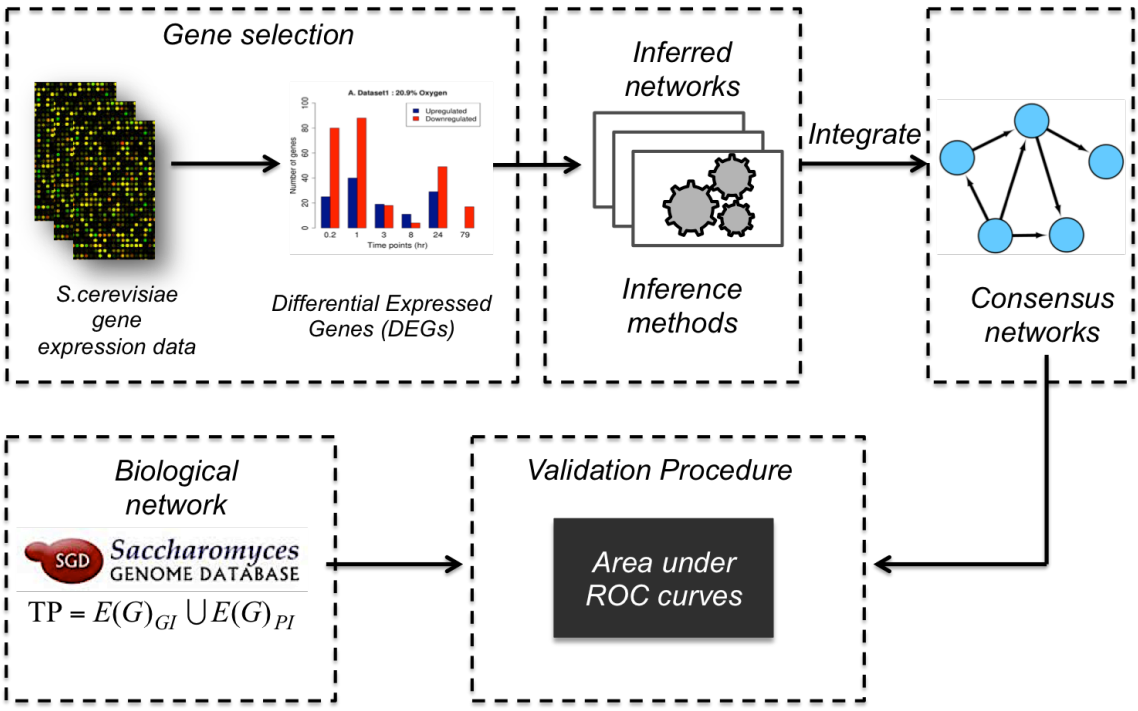


Figure 5.3: Network validation workflow for real gene expression data.

5.4 Results and Discussion

In this section, the results obtained while generating the consensus network using real gene expression data are discussed. We employed five popular benchmark network algorithms[14] to derive a consensus network – this was consistent with our previous analysis in Chapter 3 with *in silico* data. By applying our new permutation algorithm, the frequency statistics generated from the mutual information based algorithms CLR, ARACNE and MRNETB were transformed to *p*-values using a random sampling approach via permutation. (See the Methods section in Chapter 3 for further details on the permutation approach). For the correlation-based algorithms RedeR and WGCNA, *p*-values were calculated using the R function *corr.prob*.

The scatter plots in Figure 5.4 illustrates the relationship between the *p*-values and correlation coefficients/mutual information. The correlation coefficients calculated for each gene pair using RedeR and WGCNA from real data appear to have an inverse relationship with the corresponding *p*-values, as expected (Figures 5.4A and 5.4B). Similarly, Figures 5.4C, 5.4D and 5.4E indicate that there is a negative correlation between the mutual information estimates for ARACNE, CLR and MRNETB and the *p*-values calculated using our new permutation algorithm: if the *p*-value is smaller than a critical *p*-value, the null hypothesis is rejected and the interaction between two genes is considered statistically significant.

[14] Refer to Chapter 2 for more details on the network inference methods used.

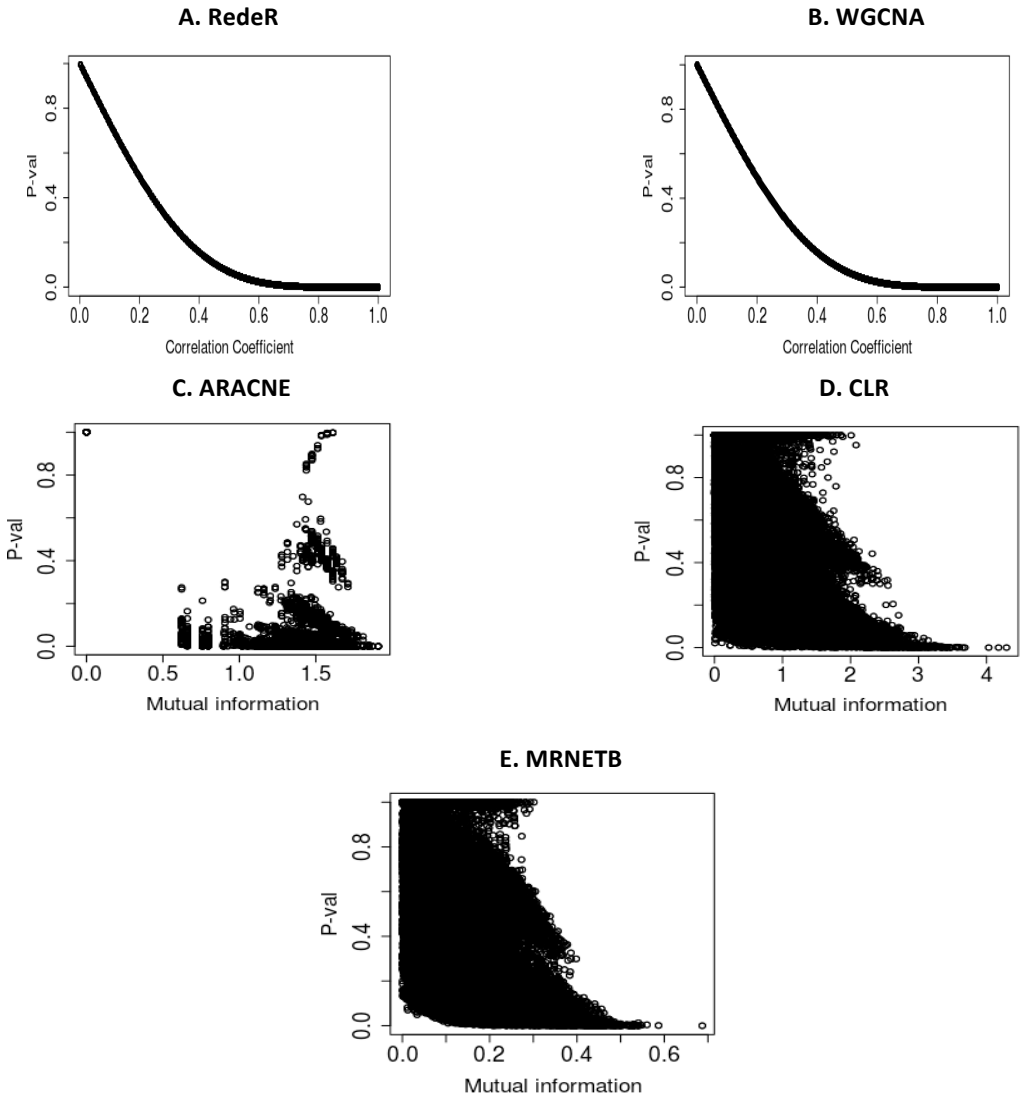


Figure 5.4: Correlation coefficients (A and B) and mutual information values (C, D and E) generated from all gene pair interactions (edges) plotted against corresponding *p*-values for different network algorithms.

Figure 5.5 compares the distributions of *p*-values generated by the different inference algorithms. For the correlation-based methods (RedeR and WGCNA - Figures 5.5A and 5.5B), a uniform distribution of *p*-values between 0 and 1 appears to be obtained. Similarly, the *p*-values calculated for the mutual information-based methods by our permutation-based algorithm (ARACNE, CLR and MRNETB - Figures 5.5C-E) also approximate a uniform

distribution. The distribution of Fisher's combined probability test (FCPT) statistic calculated for each edge is shown in Figure 5.5F.

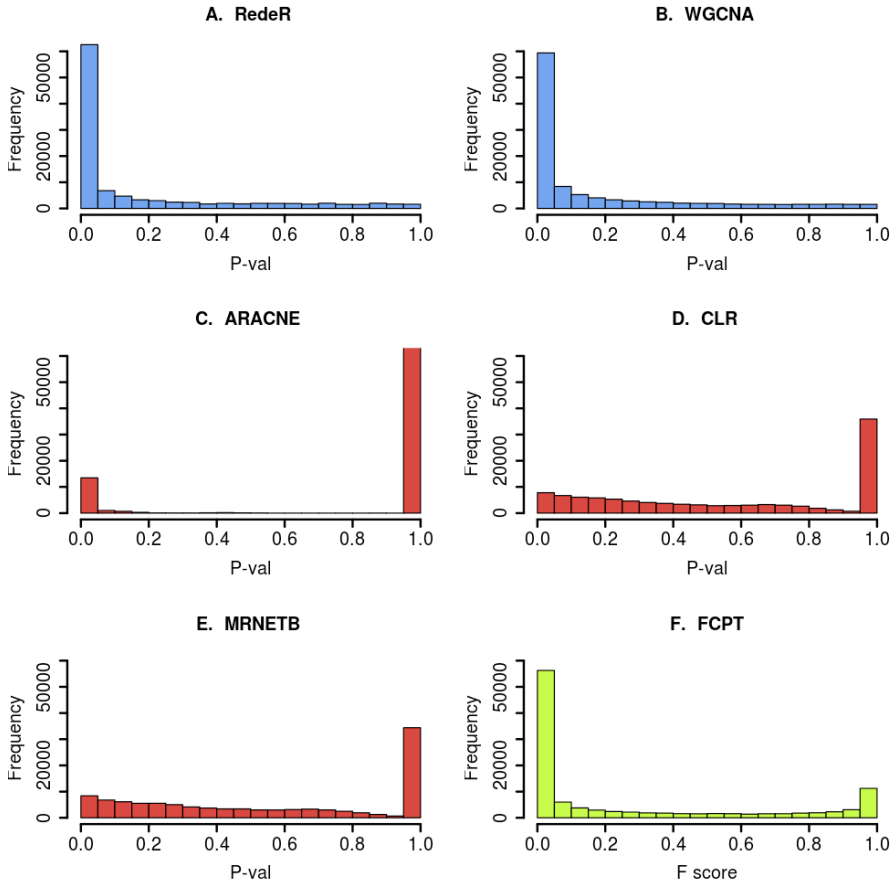


Figure 5.5: A-E: significance *p*-value distribution plots obtained from the individual network algorithms. F: the corresponding distribution of Fisher's combined test statistic.

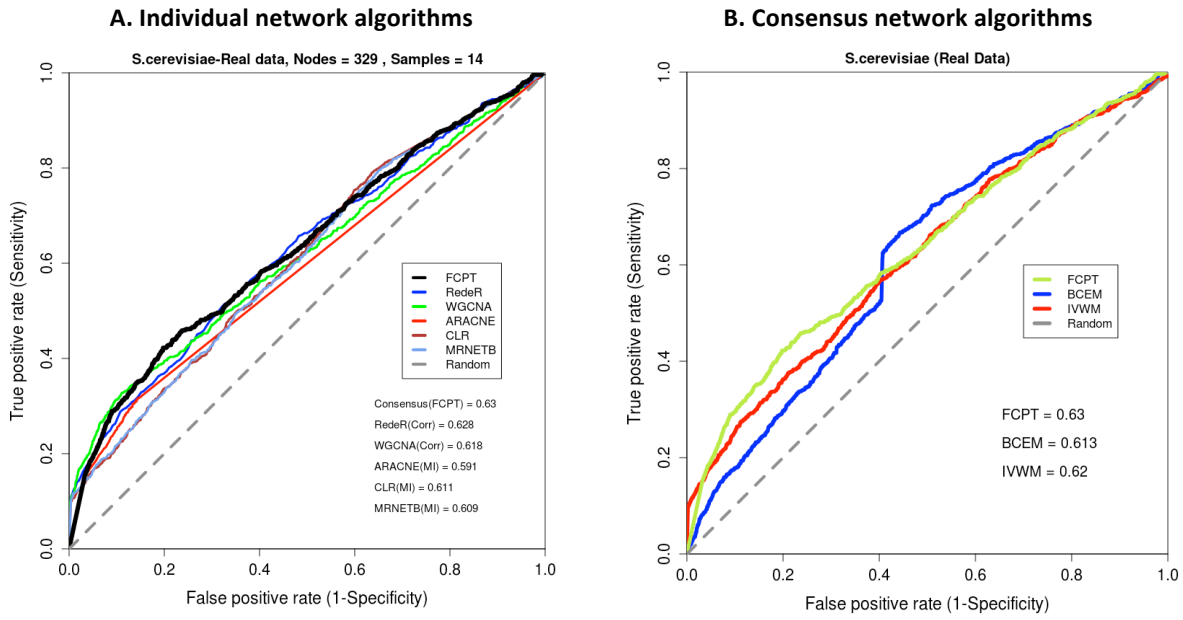


Figure 5.6: ROC curves showing the relationship between sensitivity and specificity for individual network inference methods (A) and consensus approaches (B) using a real gene expression dataset. Abbreviations: Corr-Correlation; MI-Mutual information; FCPT-Fishers combined probability test; BCEM-Borda count election method; IVWM- Inverse variance weighted method.

In order to evaluate the performance of individual network algorithms against that of consensus networks, we examined the relationship between sensitivity and specificity by calculating ROC curves at several significance thresholds. Figure 5.6A illustrates that consistent with our previous analysis in Chapter 3 using *in silico* data, the performance of the consensus network by FCPT - indicated by the solid black line - is closest to the best performing individual network (i.e. RedeR). Interestingly, when the performance of FCPT was compared against well-established consensus methods (BCEM and IVWM), it displayed promising results (see Figure 5.6B). The results obtained here with real gene expression data showed that FCPT gave an improved performance, measured against the individual network methods and other consensus algorithms in this case study.

Furthermore, the performance of each of the methods was quantified by calculating the area under the ROC curves (AUROCs) as illustrated in Figure 5.7. The comparison from this plot reveals that FCPT showed improved performance over individual networks including the dynamic Bayesian network (DBN) and popular consensus methods (BCEM and IVWM), with the highest AUROC measure of 0.63. The lower performance of DBN, can be attributed to the small number of experimental samples (14 in this study).

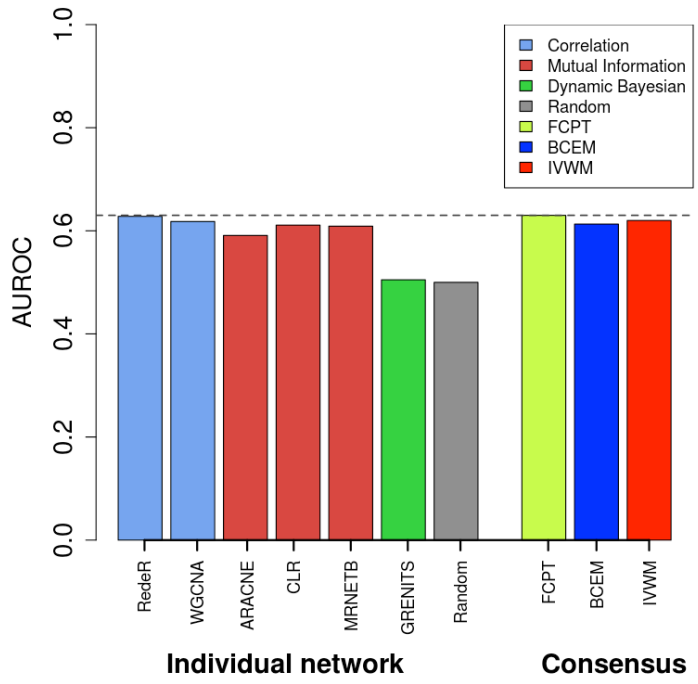


Figure 5.7: Using AUROC measures to compare the performance of the individual and consensus network inference algorithms in identifying experimentally verified regulatory interactions (genetic or physical) from the SGD database (Cherry et al. 2012).

To further investigate the potential of FCPT on real data, we evaluated the robustness of FCPT against the existing consensus methods using AUROCs, as previously implemented in Chapter 3[15]. The consensus network was built by forming an ensemble combining the best performing network inference algorithms and cumulatively adding the weaker performing

[15] Refer to section 3.3.6.4 in Chapter 3 for more details on how the weaker performing networks were combined cumulatively to form an ensemble.

algorithms. The data shown in Figure 5.8 reveals that the performance of FCPT is robust and outperforms the Inverse variance weighted method (IVWM) and Borda count election method (BCEM), even when weaker algorithms are combined. While the IVWM method performance is nearly the same as that of FCPT, when the number of weaker performing algorithms was increased, the BCEM performance decreased. A similar trend was exhibited by IVWM. However, when all five algorithms were integrated, FCPT outperforms all other consensus methods, with the IVWM performance also improving. This could possibly be due to the increased numbers of algorithms with the influence on the effect size that is weighted by the sample size. Notably, these results suggest overall that consensus approaches are robust in handling weaker performing algorithms applied to real expression data, with the FCPT yielding the most robust performance.

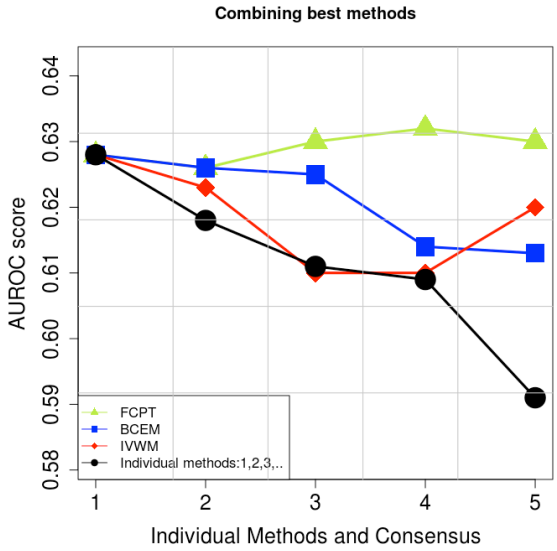


Figure 5.8: AUROC scores obtained using real gene expression data from *S.cerevisiae* with the top performing inference algorithm {1} and also by combining the top two algorithms {1,2}, the top three algorithms{1,2,3} the top four algorithms {1,2,3,4} and all five algorithms together {1,2,3,4,5}. Network predictions were validated against experimentally verified regulatory interactions (genetic and/or physical) from the SGD database (Cherry et al. 2012).

In summary, the consensus approaches are indeed, a better network inference strategy. The finding of the present study was consistent with those of Marbach *et al* (Marbach et al. 2012) and Steele *et al* (Steele & Tucker 2008) who found that consensus networks improve the accuracy of predictions. Furthermore, these results were found to be consistent with the *in silico* data described in Chapter 3.

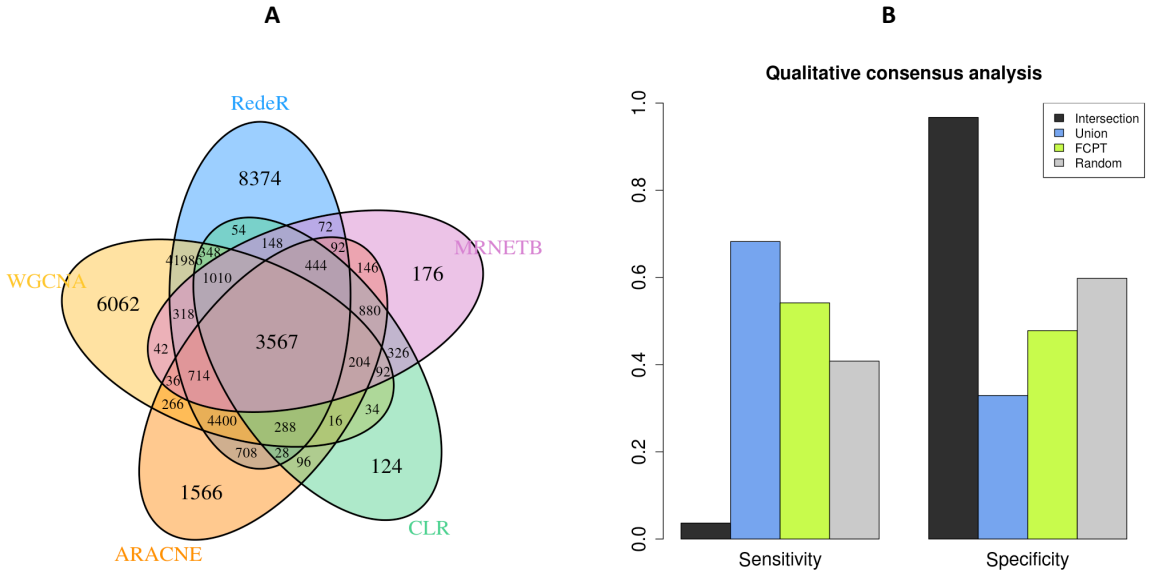


Figure 5.9: A) Venn diagram comparing the statistically significant interactions (p<0.05) obtained using five different network inference algorithms. The intersection set of common predicted interactions corresponds to the naïve qualitative consensus network. B) Sensitivity and specificity measures for qualitive consensus networks (intersection and union) against the consensus network by FCPT (q<0.05) and random networks.

Next, we compared the consensus network via FCPT against a qualitative consensus networks (intersection and union) at statistical significance (p<0.05) as we previous analysed in Chapter 3. We used two measures to evaluate performance: sensitivity and specificity. The results obtained for this analysis are presented in Figure 5.9. The Venn diagram in Figure 5.9A shows the consistency of regulatory interactions across the individual inference algorithms, of which 3,567 interactions were observed to be common. One can see from the

plots shown in Figure 5.9B that the union method shows the highest sensitivity and lowest specificity, while the intersection method showed the lowest sensitivity and highest specificity. A possible explanation for this might be attributed to the nature of the union method, which has a larger set of edge predictions (compared to intersection and FCPT) - includes many true positives - but with many more false positives than that of Intersection and FCPT. Notably, FCPT showed better sensitivity than the intersection and Erdős–Rényi (ER) random networks and with relatively better specificity. These results were found to be consistent with those obtained with *in silico* data.

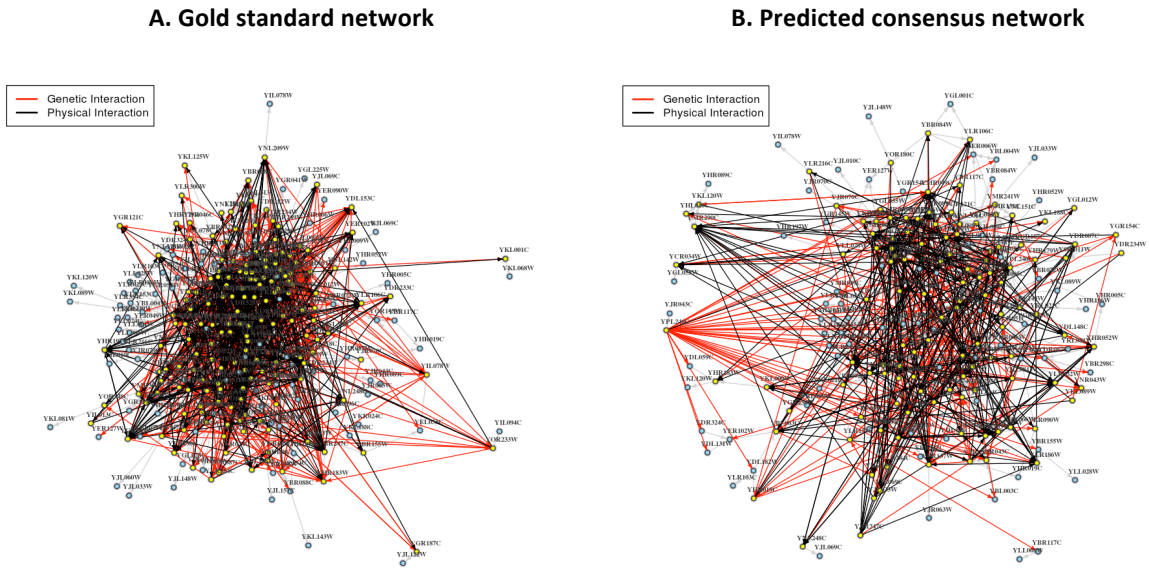


Figure 5.10: Gold standard and predicted consensus networks. A). Gold standard network showing experimentally verified edges corresponding to genetic (GI) and physical interactions (PI), indicated by red and black edges respectively. Yellow and blue nodes denote genes/proteins, where the former are either associated with GI or PI and the latter are not. B). Predicted consensus network by FCPT showing true interactions that are statistically significant (q<0.05). The color of nodes and edges is the same as in A.

Figure 5.10 compares a schematic of the gold standard network with a schematic of the consensus network obtained by identifying statistically significant edges (q<0.05) using

FCPT. The gold standard network (Figure 5.10A) was constructed by extracting genetic interactions (GIs), and physical interactions (PIs) associated with DEGs (see the Methods section for details). The red and black edges represent true genetic and physical interactions that are experimentally verified and comprise 875 edges. The yellow nodes denote genes that are associated with true interactions. While the blue nodes indicate genes that are not associated with any true interactions (GIs or PIs). Figure 5.10B shows true edges predicted by the consensus network (FCPT), composed of 475 statistically significant edges (q<0.05), of which 191 and 303 edges are associated with GI and PI respectively and are verified as real biological interactions using the curated Saccharomyces Genome Database (SGD). For simplicity, self-edges are not included in the graphs. Both the graphs are drawn using igraph (Csardi 2010). The performances and other statistical measures of individual inference algorithms (RedeR, WGCNA, ARACNE, CLR, MRNETB) and consensus by FCPT are summarized in Table 5.1. It can be seen from the aforementioned Table that FCPT has improved performance measures and identifies many experimentally verified GIs and PIs. Here, GI.nodes and PI.nodes are those nodes that are associated with GIs and PIs respectively.

Table 5.1: Performance statistics for individual network inference methods (RedeR, WGCNA, ARACNE, CLR and MRNETB) and Consensus by FCPT at significance level $p<0.05$. Abbreviations: GI–genetic interaction; PI-physical interaction.

	RedeR	WGCNA	ARACNE	CLR	MRNETB	Consensus (FCPT)
Nodes	216	215	133	87	92	195
Edges	615	555	187	82	92	475
GI.nodes	167	161	80	43	45	147
GI	253	233	79	32	34	191
PI.nodes	139	140	79	55	60	122
PI	383	342	116	52	61	303
Sensitivity	0.718	0.657	0.286	0.18	0.191	0.577
Specificity	0.423	0.452	0.877	0.93	0.925	0.599

5.4.1 Hierarchical Modularity

In this section, we extended the investigation on real gene expression data from *S.cerevisiae* to explore hierarchical modularity within GRNs in order to delineate the biological processes associated with modular networks. Furthermore, we quantified biologically meaningful sub-networks using our proposed module and model scores[16] by using the Gene Ontology (GO) enrichment analysis. The network algorithm employed here are RedeR, WGCNA and SIMoNe (refer to the Methods section in Chapter 2) that investigated modules. The flow process employed to address modularity in this section is similar to that of Chapter 4.

Network graphs showing a hierarchical modular structure constructed from real gene expression data are shown in Figure 5.11. The plot attributes are consistent with those of corresponding plots in Chapter 4. Here, we have shown graphs constructed by optimizing the

[16] See Methods section in Chapter 4 for more details on the calculation of module and model scores

number of the cluster module thresholds to 8. In addition, we have investigated several other cluster modules (4,12 and 16) by fixing the threshold for unbiased comparative analysis.

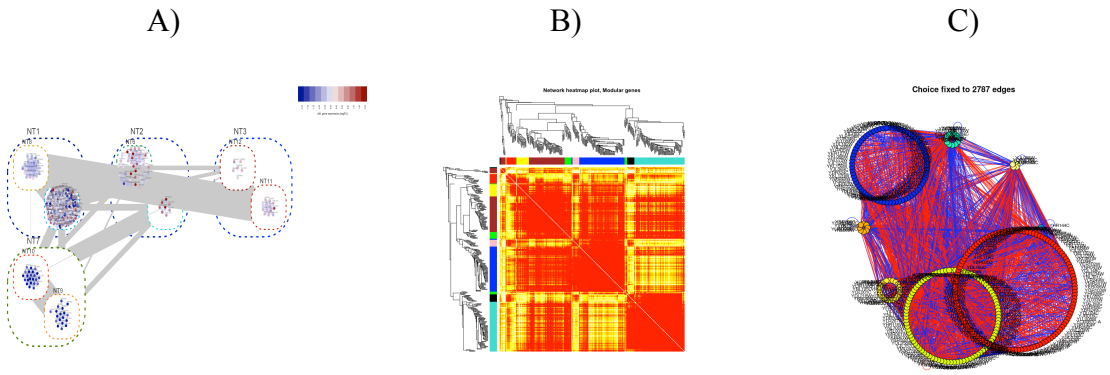


Figure 5.11: Hierarchical and modular networks consisting of 8 modules obtained with real gene expression data. A). RedeR. The modules are clustered in a hierarchical structure. B). WGCNA. The red box plots across the diagonal represents modules. C). SIMoNe. The clusters produced determine the modular network.

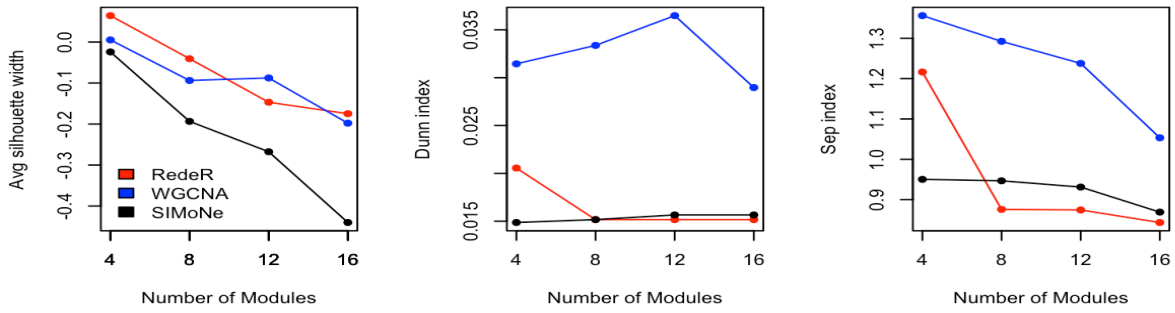


Figure 5.12: Internal validation indices. Average silhouette width, Dunn index and Separation index calculated for different numbers of cluster modules generated from each of the network algorithms using real gene expression data.

Consistent with our previous analysis in Chapter 4, we validate modular structure by measuring internal and external validation indices to investigate the quality and relevance respectively of cluster modules for biological processes. Figure 5.12 shows the following internal validation measures computed for several cluster modules using different network

algorithms: Average Silhouette Width (ASW), Dunn Index (DI) and Separation Index (SI) (see the Methods section in Chapter 4 for more details). In all, for each of the internal validation indices used, higher values are associated with a better quality of partitioning cluster modules. Furthermore, a maximal index suggests that it predicts an optimal number of modules. These results were obtained by measuring Euclidean distances between samples. Notably, with a lower number of cluster modules, the quality of separation was found to be better. As the number of clusters increases, the quality of separation appears to deteriorate for all the network algorithms tried. This observation was noted in the ASW and SI measures. Furthermore, this trend was examined in the DI measure for RedeR and WGCNA, for which the quality of partitioning improved when the number of modules was set to 12. Overall, these measures showed consistent results, with WGCNA giving the best internal validation indices measures, followed by SIMoNe and RedeR, over several module numbers. In addition, these measures help to determine the optimal number of cluster modules to use for any given network algorithm and expression dataset.

To further investigate the performance of network algorithms with respect to biological relevance, we used an external validation measure. In particular, we applied gene ontology (GO) enrichment analysis to identify statistically significant (adjusted $p<0.05$) cluster modules that show are over represented for biological processes (BPs), which were later quantified the ability of network algorithms to reveal biologically meaningful results using module and model score, as described in the Methods section of Chapter 4. The results of the GO enrichment analysis for each cluster module obtained from different network algorithms at various statistical thresholds are presented in Figure 5.13.

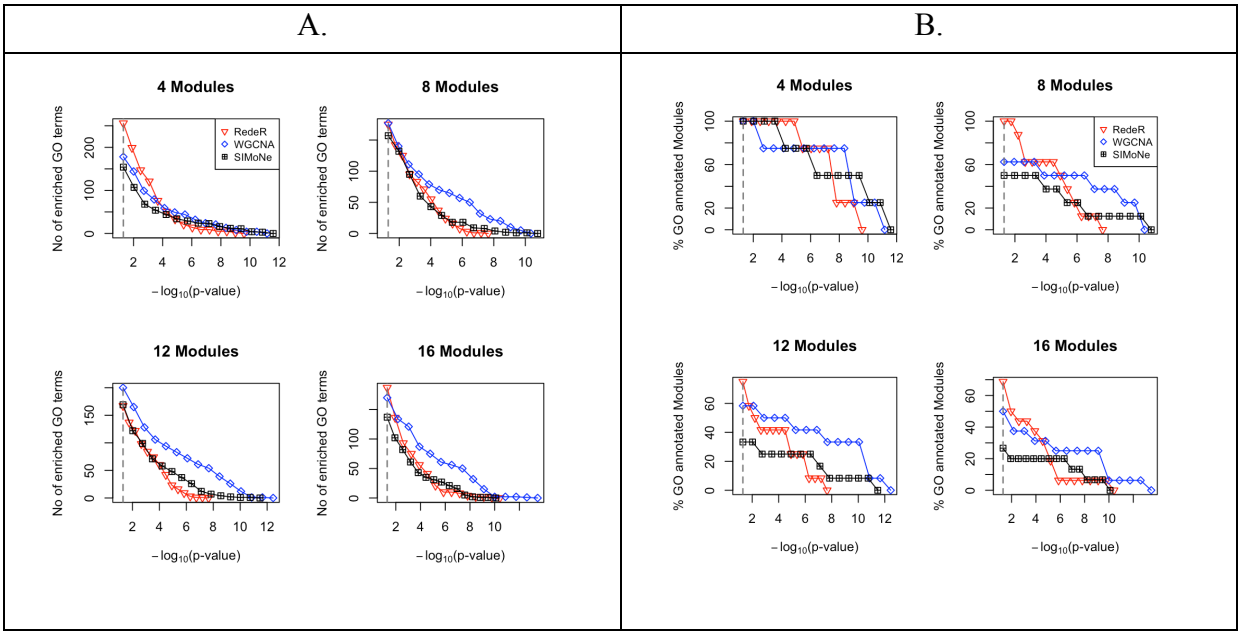


Figure 5.13: A) Number of enriched GO terms found for different numbers of modules generated from real gene expression data at various *p*-value cutoffs. *p*-values are plotted on the –log10 scale. B). Percentage of annotated GO terms found for different numbers of modules at various *p*-value cutoffs as in A). The dashed line in each plot indicates the critical significance threshold ($p<0.05$).

The preliminary GO analysis shown in Figure 5.13A was performed by counting the number of GO terms that are enriched at various statistical thresholds on the transformed –log10 scale, for different numbers of modules. It should be noted that lower significance values correspond to higher measures on the –log10 scale, as presented in the x-axis of Figure 5.13. One can see from these plots that WGCNA contains more enriched GO terms at several numbers of modules numbers and significance thresholds, compared to SIMoNe and RedeR. Similarly, in Figure 5.13B, at a higher significance level, WGCNA yielded a higher percentage of modules that had at least one statistically enriched GO term at higher statistical thresholds. The overall trend of the preliminary GO enrichment analysis with real data was found to be consistent with the *in silico* data in Chapter 4.

The plots presented in Figure 5.14-5.16, show the top 5 biological processes (BPs) which are statistically significant (adjusted $p<0.05$) after GO enrichment analysis, for varying

numbers of cluster modules using different network algorithms with real expression data. As an illustration, we also show those cluster modules that are associated with the top 5 BP that are statistically significant.

One can see from Figure 5.14, that with 4 cluster modules derived from each network algorithm, many modules are found to be common across those algorithms. For example, M1 from most algorithms was associated with nitrogen compound metabolic processes, whilst M2 was associated with primary metabolic processes. M4 from RedeR and M3 from WGCNA and SIMoNe were associated with a single organism cellular process. It is interesting to note that additionally, WGCNA and SIMoNe identify cellular response to pheromone and aromatic compound metabolic processes. Similarly, when the number of modules was increased to 8 (Fig 5.15), more stratified BPs were revealed by RedeR (for example, RedeR identified ribosome biogenesis regulation), whereas WGCNA and SIMoNe showed similar results to those obtained with 4 cluster modules. With 12 cluster modules (Fig 5.16), WGCNA showed association with cellular response to pheromone and to a carbohydrate catabolic process. Finally, with 16 cluster modules (Fig 5.17), many BPs appear to show commonality across algorithms, although WGCNA shows association with coenzyme and cofactor metabolic process with conjugation in addition to the previous BPs identified.

In the experimental culture set up, the transcriptomic data from *S.cerevisiae* was extracted following a shift from fully aerobic conditions (20.9% O_2) to anaerobic conditions, which stimulated many metabolic pathways associated with catabolic processes that included the following: organic cyclic compound metabolic process, phosphate-containing compound metabolic process, cellular aromatic compound metabolic process, ribosome biogenesis, response to pheromone and oxidation-reduction process. This was reflected in our GO analysis, which was found to be consistent with the data analysis in (Rintala et al. 2011).

As the above GO enrichment results may be biased by the number of modules used, or the module size distribution, we used our proposed module and model scores to compare the performance of the difference network algorithms. The results of this analysis are shown in Figure 5.18, which show the relevance of module scores to the most functional modules that are biologically meaningful across different modules obtained from different network algorithms. In order to score the respective modules, we select the BP that has the smallest corresponding *p*-value[17].

[17] Refer to the Methods section in Chapter 4 for more details on the selection of *p*-values from GO enrichment analysis.

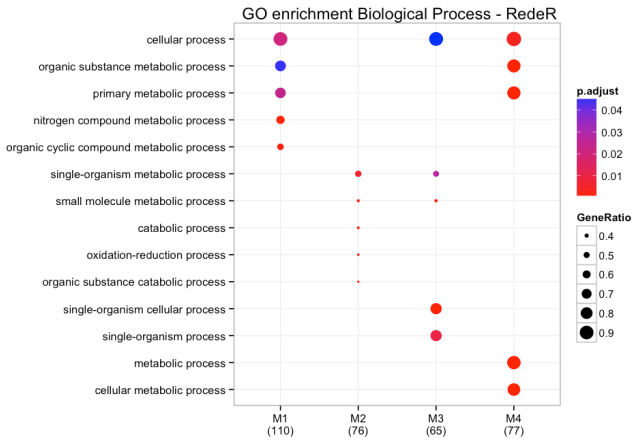
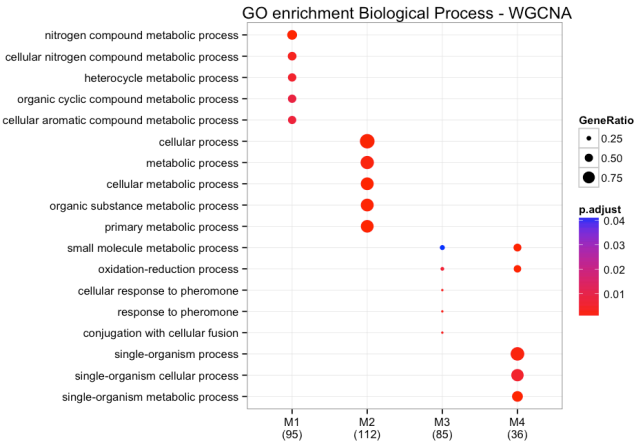
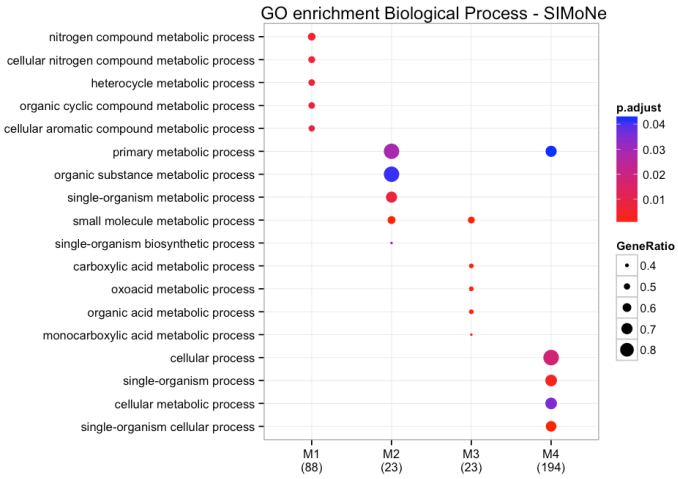
RedeR	
WGCNA	
SIMoNe	

Figure 5.14: GO enrichment analysis for 4 cluster modules that are significantly enriched for BPs, using real gene expression data. The dot size denotes gene ratio (GR) and color indicates significance *p*-values. GR is the ratio of the total number of genes that are associated to a BP in GO enrichment to the number of genes that are associated with a particular module.

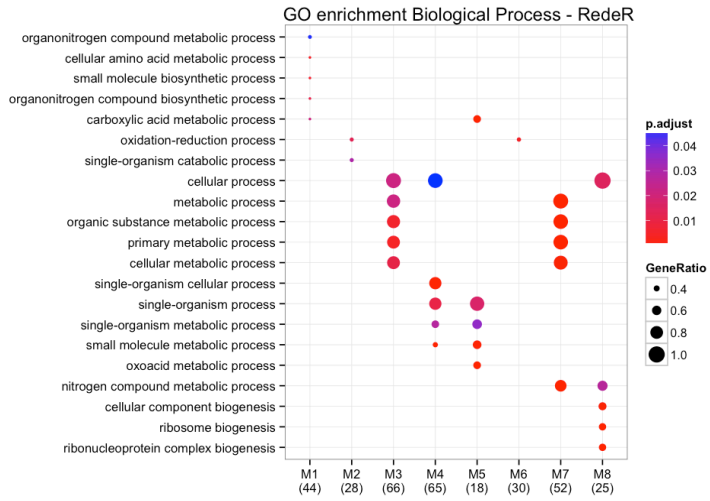
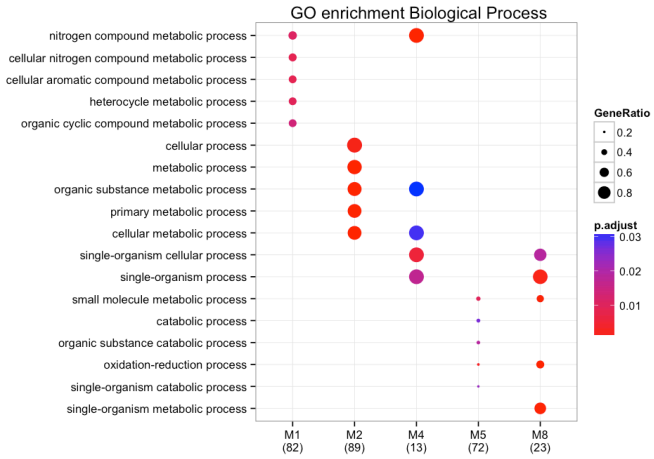
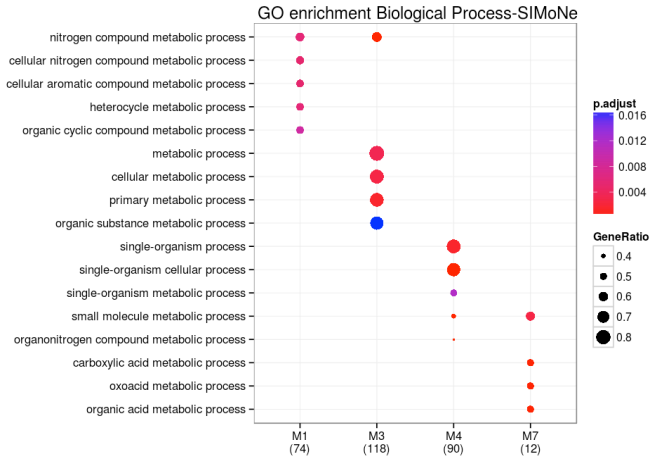
RedeR	
WGCNA	
SIMoNe	

Figure 5.15: GO enrichment analysis for 8 cluster modules that are significantly enriched for BPs, using real gene expression data. The dot size denotes gene ratio (GR) and color signifies significance *p*-values. GR is same as in Figure 5.14.

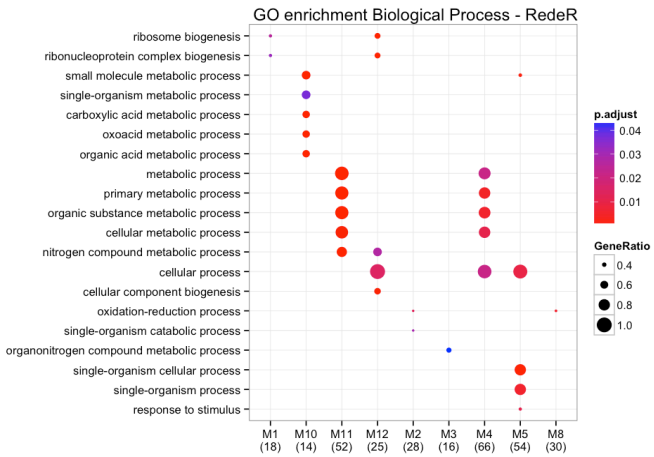
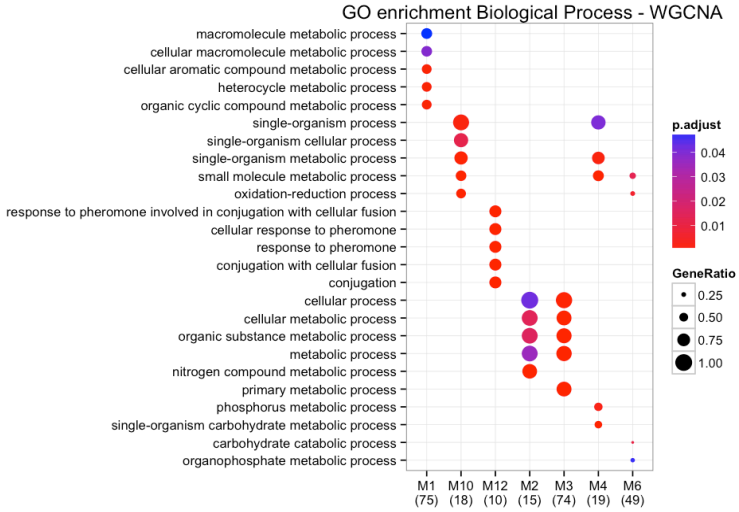
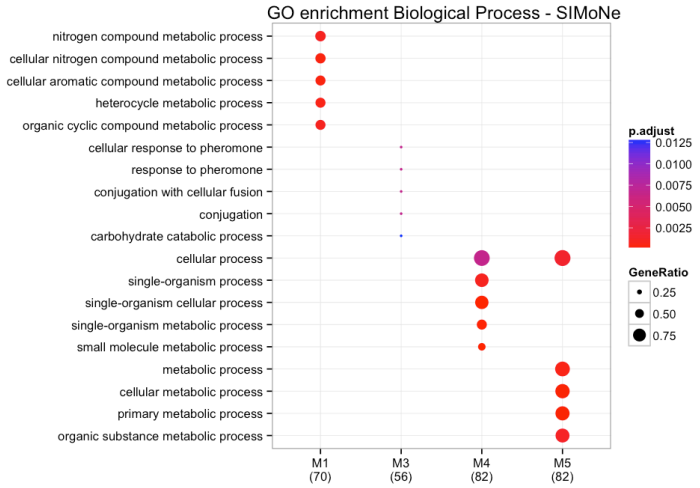
RedeR	
WGCNA	
SIMoNe	

Figure 5.16: GO enrichment analysis for 12 cluster modules that are significantly enriched for BPs, using real gene expression data. The dot size denotes gene ratio (GR) and color signifies significance *p*-values. GR is same as in Figure 5.14

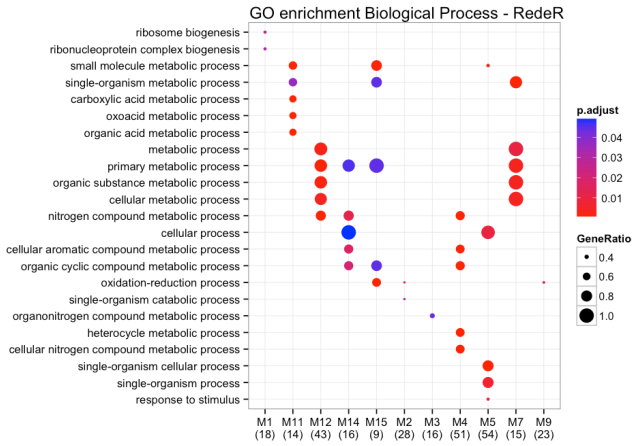
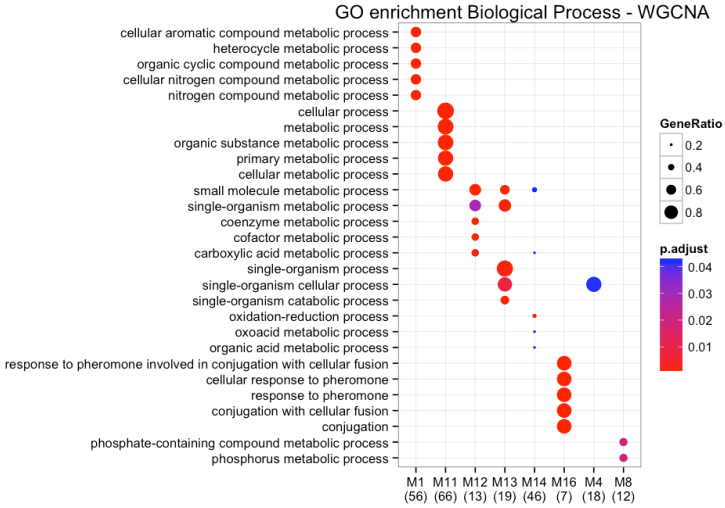
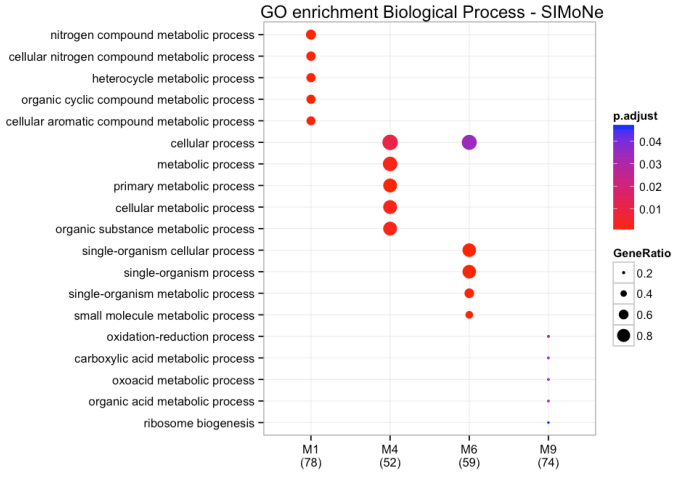
RedeR	
WGCNA	
SIMoNe	

Figure 5.17: GO enrichment analysis for 16 cluster modules that are significantly enriched for BPs, using real gene expression data.The dot size denotes gene ratio (GR) and color signifies significance *p*-values. GR is same as in Figure 5.14

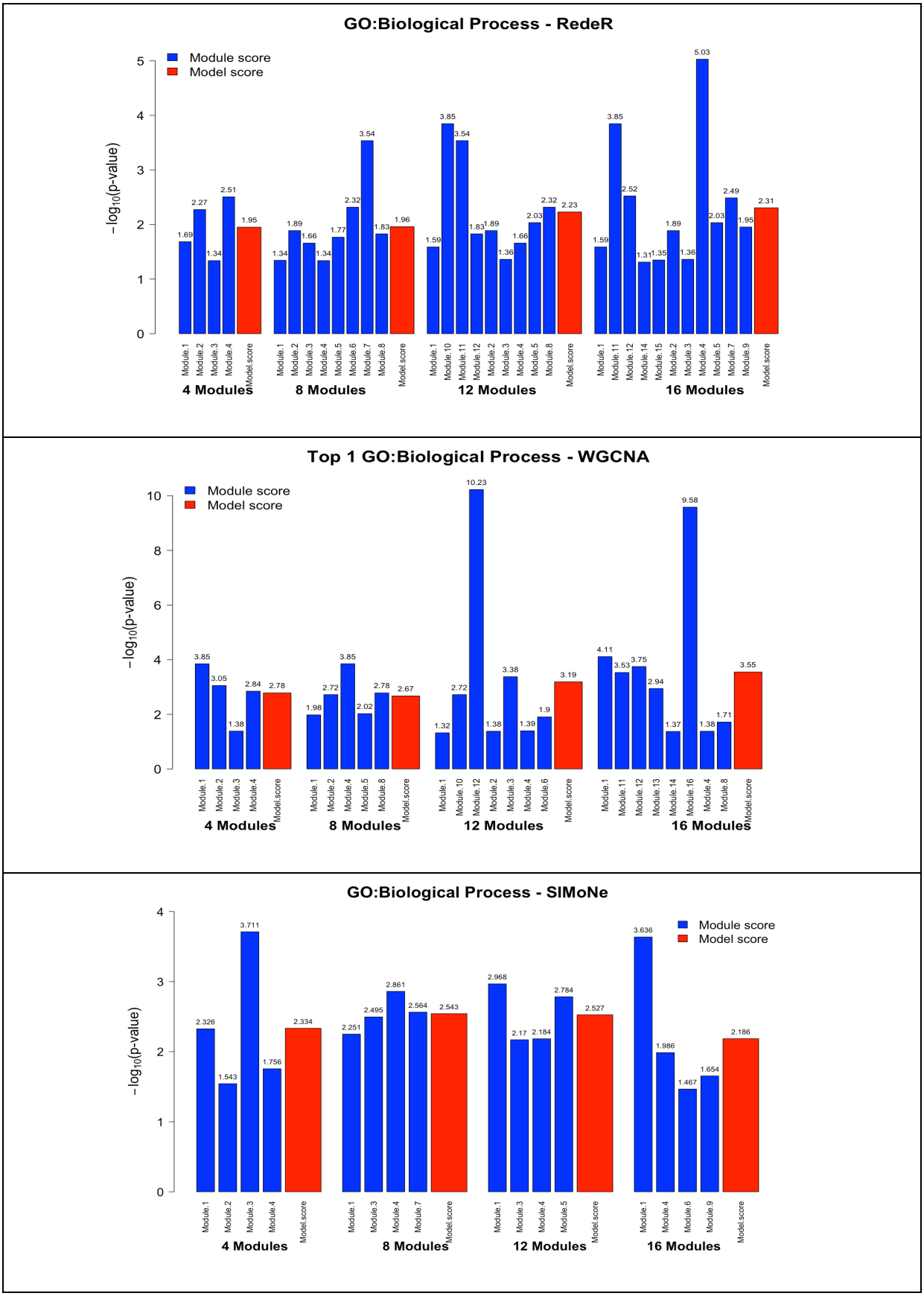


Figure 5.18: Modular and model scores for the top enriched GO terms for different numbers of modules with real gene expresion data.

Figure 5.18 compares module and model scores, evaluating the performance of network algorithms and intra-modules for their ability to identify biological meaningful modular networks. From this data, we can see that the performance of RedeR and WGCNA improves as the number of modules is increased. This may be because as the module number increases, the number of genes associated with BPs are further stratified, with a relatively smaller number of genes being over–represented for a particular BP during the GO analysis. For example, with 4 modules, M4 in RedeR was associated with the cellular process, which is the generic top level BP; however, when the number of modules increased to 12, M4 showed association with an organic, cellular metabolic process. In contrast, the performance of SIMoNe improved with the number of modules, but then subsequently decreased. These results suggest that based on model scores, WGCNA consistently demonstrated the best performance when identifying biologically meaningful results with real gene expression data, followed by SIMoNe and RedeR (see Figure 5.19). Furthermore, the best performing modular networks that maximize model score - and its corresponding annotations to GO analysis - are summarized in Table 5.2.

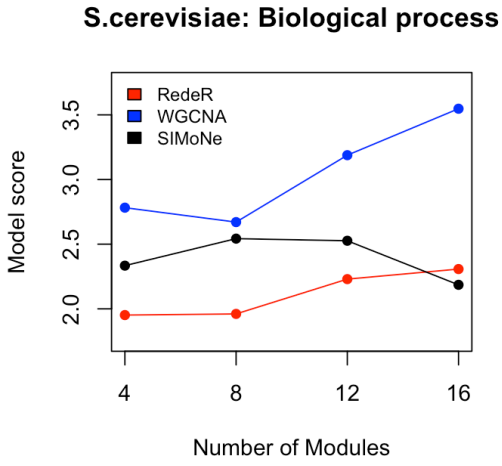


Figure 5.19: Model scores obtained using different network algorithms for a various numbers of modules with real gene expression data.

Table 5.2: Functional top ranked modules from different network algorithms that show statistically significant (p<0.05) association to biological process in GO enrichment analysis from real gene expression data.

Module	GO.ID	Biological Process	Count	P.value	Module.score
		RedeR			
M1	GO:0042254	ribosome biogenesis	6	0.02575	1.589
M11	GO:0044281	small molecule metabolic process	9	0.00014201	3.848
M12	GO:0008152	metabolic process	39	0.0029889	2.524
M14	GO:0009987	cellular process	16	0.048776	1.312
M15	GO:0044238	primary metabolic process	9	0.044606	1.351
M2	GO:0055114	oxidation-reduction process	8	0.01292	1.889
M3	GO:1901564	organonitrogen compound metabolic process	7	0.043432	1.362
M4	GO:1901360	organic cyclic compound metabolic process	35	9.34E-06	5.03
M5	GO:0009987	cellular process	51	0.0092284	2.035
M7	GO:0044238	primary metabolic process	15	0.0032435	2.489
M9	GO:0055114	oxidation-reduction process	7	0.011097	1.955
		WGCNA			
M1	GO:0006725	cellular aromatic compound metabolic process	35	7.74E-05	4.112
M11	GO:0009987	cellular process	64	0.00029501	3.53
M12	GO:0044281	small molecule metabolic process	9	0.00017853	3.748
M13	GO:0044699	single-organism process	18	0.0011432	2.942
M14	GO:0044281	small molecule metabolic process	15	0.042452	1.372
M16	GO:0000749	response to pheromone involved in conjugation with cellular fusion	6	2.61E-10	9.583
M4	GO:0044763	single-organism cellular process	16	0.041899	1.378
M8	GO:0006796	phosphate-containing compound metabolic process	6	0.01928	1.715
		SIMoNe			
M1	GO:0006807	nitrogen compound metabolic process	42	0.0056137	2.251
M3	GO:0008152	metabolic process	97	0.0031956	2.495
M4	GO:0044699	single-organism process	70	0.0013771	2.861
M7	GO:0044281	small molecule metabolic process	7	0.0027278	2.564

5.5 Conclusions

The objective of this chapter has been to investigate whether the consensus network obtained by combining the statistical significance scores generated by different network algorithms for each gene pair (edge) provides the best results compared to the individual network algorithms using publicly available real time series microarray gene expression data from *S.cerevisiae* (Rintala et al. 2011). Our proposed quantitative consensus method (FCPT) is an unsupervised approach which requires no prior biological knowledge. It aims to identify genome-wide regulatory interactions by integrating predictions from frequency-based individual network inference methods which are less expensive and therefore yield a higher confidence consensus network. In addition, we find the integrative approach to predicting weighted biological interactions provides a robust and powerful tool with which to investigate regulatory networks.

The results from the consensus study were compared against two popular existing quantitative consensus approaches: BCEM and IVWM. Although BCEM and IVWM showed promising results, FCPT demonstrated the best results overall, assessed using AUROC measures. These results suggest that the consensus approaches do indeed provide a better network inference strategy. Furthermore, when FCPT was compared against qualitative consensus networks (i.e. intersection and union), better results were also demonstrated at significance level $q<0.05$ in terms of sensitivity and specificity, where the naïve consensus (intersection) approach underperforms in determining true biological interactions. It should be noted that many of the unique consensus predicted interactions (267) have not been documented in the curated SGD and thus require further experimental investigation.

The investigation on real data was further extended to explore modularity for hierarchical networks within the GRNs, in order to identify the statistically significant

biological processes (BPs) associated with modular networks. In the latter part of this chapter, we quantified the quality in the separation of modular networks using internal validation indices. Furthermore, module and model score were implemented as an external validation measure to investigate biologically meaningful sub-networks by examining Gene Ontology (GO) enrichment analysis associated with BP. The results from our scoring reveal that WGCNA outperformed other algorithms in identifying biologically meaningful modular networks at several module number thresholds. Therefore, these measurements will facilitate future research in downstream analysis of biologically meaningful modular networks of choice, in order to investigate the gene regulatory interactions within modules to understand the transcriptional program further.

In summary, although our method has so far only used gene expression data from a simple eukaryote, it has the potential to predict regulatory interactions from higher eukaryotes. In addition, we find there is scope to improve the performance of the algorithms by combining multiple expression datasets as input, so that the network generated is not biased to a particular input dataset. Further investigation is required, we feel, to incorporate the ability to differentiate between activation and inhibition in gene-protein interactions, as this will open a new avenue for identifying large scale molecular targets for drug discovery.

Chapter 6

Conclusion and future work

6.1 Summary

The ability of a cell within a living organism to continuously sense and respond to changes in the environment reflects its proper functioning. This involves the coordination of different layers of regulatory networks and in particular GRNs. These control various parts of the system and are one of the central components in computational and systems biology when unraveling cellular functions in biological networks between genes and gene products. Furthermore, the phenotypic changes observed in the system are not associated with just one single gene interaction, but to a cascade of them. Therefore, in order to understand the behavior of the cellular system, it is important to discover the modular components within the system. It is not sufficient to understand only the behavior of the organism, but also the essential part GRNs plays in unraveling the connectivity between modular components.

Despite recent advances in microarray technology that has enabled us to produce more accurate GRNs that represents biological phenomena - but with associated noise - we only have partial knowledge of the mechanisms of GRNs. This coupled with the incompleteness of data generation, impairs our ability to characterize functional organization at the system level. In this thesis, we addressed a major challenge in the field of systems biology - optimizing network inference methods to provide robust GRNs and quantify biologically meaningful modular networks. In particular, we focused our attention on heterogeneous network prediction via several reverse engineering methods when the same experimental data was

used. In other words, an edge interaction predicted by one algorithm might not be predicted by the others. This raises a lot of concerns regarding the validity of the inference algorithm and its predictions. Analogous to the traditional approach to developing a new algorithm, we formulated a new approach to forming an ensemble of predictions from diverse inference algorithms in order to yield a consensus network. Consensus approaches are well known for their robustness in decision making; however the Naïve consensus method - the most conservative approach - may not be a good alternative. The Fisher combined probability test (FCPT), in theory, is a sound approach for consensus decision-making, as it has been successfully used in many other disciplines.

Motivated by the "*Wisdom of Crowds*", the main deliverable of this thesis is to leverage the power of FCPT and the diversity of reverse engineering methods, so as to provide more accurate and robust GRNs which resonate closer to the true biological networks. More specifically, the current study investigated a variety of existing network inference methods and built up a new network inference approach referred to as a consensus network, using an ensemble of predicted edge interactions by means of FCPT which is presented in Chapter 3. The underlying hypothesis is that FCPT provides a robust probabilistic measure to detect if a gene interaction is significant. Not all of the network inference algorithms deliver *p*-values for edge significance. Therefore, in order to apply FCPT, a non-parametric algorithm was developed for converting frequency statistics to *p*-values using the random sampling approach through permutation analysis. The consensus network was generated using a single hypothesis test derived from FCPT, which was further enhanced by performing multiple hypothesis testing by considering the false discovery rate (FDR) as a means of controlling the occurrence of false positives. The performance of the consensus network was validated with a variety of *in silico* benchmarks datasets, that including the DREAM4 challenge, and was compared against individual methods and

consensus learning methods, specifically: 1) static and dynamic Bayesian networks: 2) quantitative consensus approaches (BCEM and IVWM): and 3) qualitative consensus (intersection and union). The results of this investigation showed that consensus networks by FCPT are robust and predict many biological interactions with higher performance measures than individual and existing consensus methods.

The most interesting finding of this study was that for larger networks (500 genes), the consensus network outperforms all single inference methods. This included the Bayesian networks and other consensus methods when performance was assessed using AUROC measures, suggesting that consensus by FCPT provides better value for large scale network inference. In addition, it delivers robust and efficient predictions, compared to other existing consensus methods. Another important discovery was that at high levels of experimental noise (30%), the FCPT consensus network demonstrated the best performance for medium size networks (100 genes) and the third best for large sized networks (500 genes), in terms of AUROC measures. Furthermore, when compared to existing consensus approaches, FCPT performed the best for large sized networks and second best with medium sized ones. A possible implication of this is that the consensus network by FCPT, overall, is a good alternative method for robust network inference. It is also interesting to note that FCPT is equally efficient when compared against popular existing consensus methods, for combining predictions for medium sized networks (Table 3.4), outperforming BCEM and IVWM for both medium and large sized networks with higher efficiency.

As an extension of the *in silico* work, the consensus model was applied to real gene expression data from the *S.cerevisiae* network in Chapter 5. These results demonstrate that the consensus network predicts many genome-wide biological interactions with high accuracy, in terms of AUROC, while outperforming other qualitative approaches. Hence, these findings relate back to the claims made in Chapter 1, thus confirming our hypothesis

that consensus by FCPT provides a robust and efficient consensus learning strategy, with great potential for decision making based on a variety of *in silico* benchmarks (Chapter 3) and also real datasets (Chapter 5). The summary of performance gains by consensus networks in terms of fold changes are reported in Table 6.1.

Table 6.1: The gain in average performance by consensus network in terms of fold changes from different sized datasets. The standard deviation is indicated in brackets and is followed by the maximum value.

Diversity in data	Size	**Transcriptional GRNs**
		Fold changes
***In silico* data**	10	1.089 (0.14) / 1.34
	100	1.081 (0.08) / 1.26
	500	1.075 (0.069) / 1.217
***In vivo* data**	329	1.067 (0.091) / 1.247

In Chapter 4, we make use of the internal validation procedure, essentially focusing on the evaluation of the quality, or goodness, of cluster modules. These scores do not reveal information on biological context, but place emphasis on topological characteristics as to how well cluster modules are separated from each other. Although there are numerous quantitative internal validation methods which exist, here we evaluate the quality of cluster modules generated from each network algorithm by employing popular non-linear internal validation indices, namely: 1) Average Silhouette Width (ASW); 2) Dunn Index (DI); and 3) Separation Index (SI). Furthermore, in this thesis, we proposed module and model scores as a means of external validation measure to quantify functional modular networks and performance of network algorithms. Module scores incorporate the statistical significance values of gene modules that are over-represented for biological processes (BPs) by examining Gene Ontology (GO) enrichment statistics in Chapter 4 using *in silico* datasets. We additionally

made use of our proposed scoring technique by applying it to real expression data in Chapter 5. Although the module and model scores are simple, they still provide a robust means of identifying biologically meaningful results. Module and model score will also aid as a guide for biologists, to help choose biologically meaningful network modules for further investigation and suitable network algorithms. The findings of this investigation complement those of earlier studies. Many of the higher scoring modules which have a significant association with a particular biological process were found to be consistent with the results obtained by Rintala *et al*. when the same datasets were used (Rintala et al. 2011).

Our results were based on internal validation indices and external validation model scores, indicating that the performance of algorithms vary with different datasets. This suggests that the scores are not biased to any particular network algorithm. In addition, we used significance values of a highly enriched biological category to evaluate the algorithms in contrast to the percentage of gene coverage within a module that is associated with an over-represented BP in GO analysis which was demonstrated in previous studies (Richards et al., 2008). Returning to the second hypothesis posed at the beginning of this study, it is now possible to state that the scoring strategies do provide a quantitative measure to identify biologically meaningful modular networks.

In conclusion, we have shown in this thesis that consensus learning strategies do provide better value on benchmark gene expression datasets for the network inference problem. These findings enhance our understanding of consensus networks through an ensemble of predicted edge interactions, delivering high confidence results with synergistic effect. The present study confirms those previous findings and contributes additional evidence in the field of computational and systems biology, verifying that integrating network edges from diverse network inferences improves the breadth and accuracy of predictions.

6.2 Limitations of the study

A number of important limitations from this study need to be considered. The most important ones are listed below:

- The present study combines predictions from five popular frequency-based statistical inference algorithms. Although these algorithms are able to capture linear and non-linear dependencies between any two random gene variables, the current study has only focused on information theory based models. However, more sophisticated algorithms, like Bayesian models, need to be included for building consensus.
- The predicted edge interactions are directed, providing the path for the information to flow from source gene to target gene. However, a limitation of our method is that it does not differentiate between activation and inhibition.
- One of the limitations of the consensus model is that it not does not incorporate sophisticated algorithms like differential equation models or Boolean models. However, in future work, implementation of a variety of network inference algorithms will encapsulate many regulatory interactions and improve the breadth and accuracy of the model.
- The proposed module and model scores that quantify the biological relevance of cluster modules using ground truth from GO enrichment analysis are empirical and require further experimental investigation, in order to support the general applicability of the measures.
- The module score does not reveal information on the transcriptional program within the modules. To overcome this limitation, these module scores can be coupled with targeted network analysis to determine their transcription factor (TF) activity.

6.3 Future work

This research has thrown up many questions in need of further investigation.

- To further investigate consensus predictions from real gene expression datasets which are not documented in the curated database, we will detect communities or modular hubs in the consensus network. For example; the network plot below detects nodes, which are connected to many other genes called hubs that are associated with high node degree.

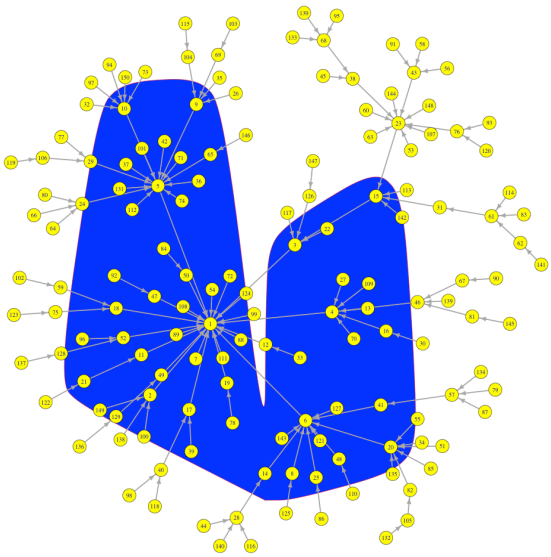


Figure 6.1: Sample network showing high degree nodes called hubs highlighted in blue.The yellow nodes signify genes.

The underlying assumption is that with a high node degree, this is a plausible TF. To further investigate this, we can employ Hidden Variable Dynamic Modeling (HVDM) (Barenco et al. 2006), a differential equation based modeling technique that uses supervised learning to predict putative TF targets.

- Incorporate more sophisticated algorithms - like Bayesian models – into the consensus building. Such algorithms encapsulate more complex interactions, like

activation or inhibition, between gene pairs (Friedman et al. 2000; Needham et al. 2007) .

- The consensus method has so far only used gene expression data from a simple eukaryote; its ability to predict the regulatory interactions from higher eukaryotes and other organisms requires further investigation.
- Integrating high-throughput data on TF-target interactions (e.g. ChIP-Seq) in addition to gene expression data will enhance our understanding of the regulation of the transcriptional network, which could improve the performance of the method and provide a more comprehensive view of GRNs.
- It should be noted that many of the consensus predicted interactions from real gene expression data of yeast (Chapter 5), have not been documented (Table A1 in Appendix-A) in the curated database and will require further experimental investigation.
- Module score proposed would help biologists to choose biological meaningful modular networks of choice to facilitate future research for investigating the transcriptional regulatory interactions associated within the modular network.

Appendix A

In this appendix, comparative analysis of a variety of existing network inference algorithms, which has been used earlier (section 3.3, Chapter 3) to study real biological networks using benchmark gene expression data. The aim of this appendix is to show by evidence that the individual network inference algorithms vary in performance using different types of benchmarked in silico (simulated) gene expression data that includes steady state and time series. The variations in the performance of each the network inference algorithm using the same expression dataset support our hypothesis that gene interactions predicted by each of the network algorithm are inconsistent. The unique gene interaction list predicted from consensus network using real gene expression datasets (Chapter 5) is also discussed later in this appendix.

Here, we compared the performance measures of various existing network inference approaches for their ability to predict true interactions. In order to provide a thorough comparison of the performance of each network inference method, we show ROC curves for each benchmarked *in silico* dataset.

A.1 SynTReN size 100

In Figure A.1, ROC curves are shown for each network inference method, using benchmark yeast GRN data of size 100 generated using SynTReN. The ROC curves plot true positive rate against false positive rate at various significance thresholds. It was observed from the plots that the performance of individual algorithms varied and the one performing best have a tendency to move nearer the true positive scale, compared to ROC curves of other

individual network inference algorithms. We calculated Area Under ROC curves (AUROC) to quantify performance measures in each case.

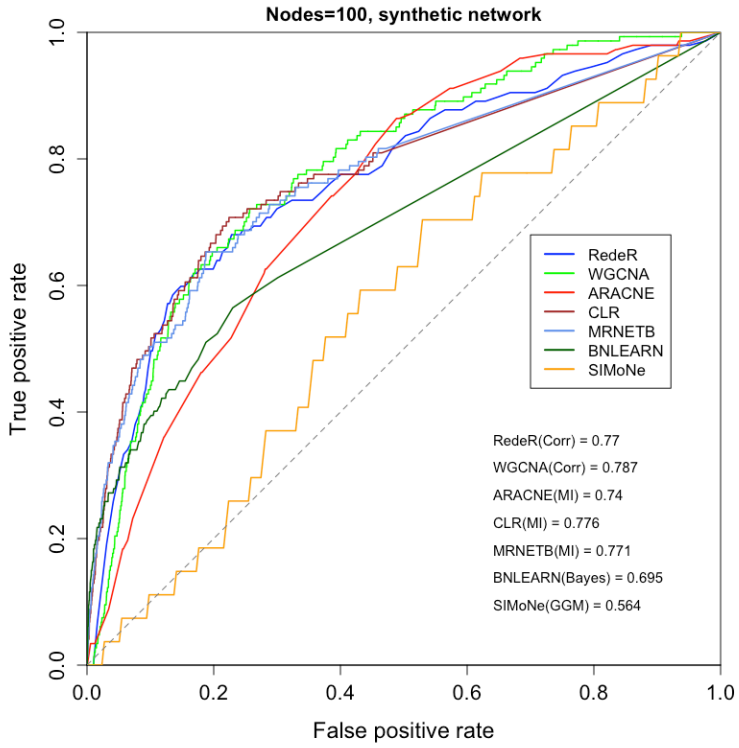


Figure A.1: ROC curves and corresponding AUROC values for different network inference approaches using a benchmarked *in silico* dataset of size (nodes) 100 and sample size 10 (perturbation experiments) generated from SynTReN (Van den Bulcke et al. 2006). Abbreviations: Corr-Correlation; MI-Mutual information; Bayes-Bayesian; GGM-Gaussian graphical model.

From these AUROC measures, the performance of each network inference method was assessed. In Figure A.1 one can see that even with a small number of samples (perturbation experiments), the correlation based WGCNA and mutual information based CLR algorithms, performed well compared to the other methods, yielding AUROC values of 0.787 and 0.776 respectively, Notably, the Bayesian method (BNLEARN) was ranked 6th behind the

correlation and mutual information based methods, which can perhaps be attributed to the low sample number.

Figure A.2 compares the average performance measures for medium (size 100) and large sized (size 500) networks. For medium sized networks (100 genes), it is evident that correlation based networks performed well compared to other methods, with RedeR giving the best performance, as quantified by a median AUROC score of 0.865. However, for large sized (500 genes) networks, WGCNA outperformed the other inference methods with a

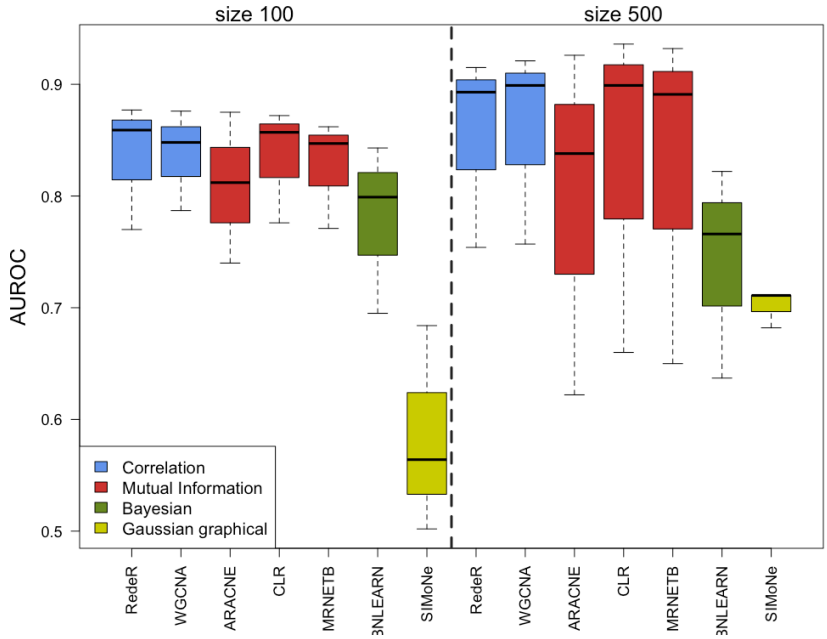


Figure A.2: Comparative average performance scores of different network inference approaches using benchmarked *in silico* datasets generated from SynTReN of size 100 and size 500, for the following sample sizes (number of different perturbation experiments): 10, 100 and 500.

median AUROC of 0.90, suggesting that the correlation approach is more capable of predicting true interactions for larger networks. Surprisingly, the performance of the Bayesian methods was comparatively poor in relation to the correlation and mutual information based methods on both sized networks, albeit with a better performance than

SIMoNe, which had median AUROCs of 0.56 (medium size network) and 0.68 (large size network).

A.2 DREAM size 10

To further enhance our comparative analysis, we employed benchmark networks of various experimental systems from the Dialogue on Reverse Engineering Assessment and the Method (DREAM4) challenge. These provide benchmarks which have been used to assess more than 30 network inference algorithms (Marbach et al. 2010; Greenfield et al. 2010). The aim of the DREAM4 challenge is to reverse engineer gene regulatory networks using simulated steady state and time series data from small sized (10 genes), and medium sized (100 genes) networks (see Table 3.1).

The boxplots in Figure A.3 compare the AUROC scores obtained for the different network inference algorithms when applied to a variety of steady state and time series data of size 10 (10 genes), as illustrated in Table 3.1. In particular, the steady state data consists of different simulated knockdown (KD), knockout (KO), and multifactorial (MF) experiments, whilst the time series (TS) data is a single experiment. For more details on the simulated experiments refer to the Methods section 3.2.4.3. It was observed from Figure A.3 that the performance of each network inference algorithm varies, depending on the type of expression data (steady state or time series) and the experimental perturbations. These results indicate that for small size (10 genes) networks, mutual information based algorithms yield better performance measures for all kinds of data types. For example ARACNE outperforms other approaches

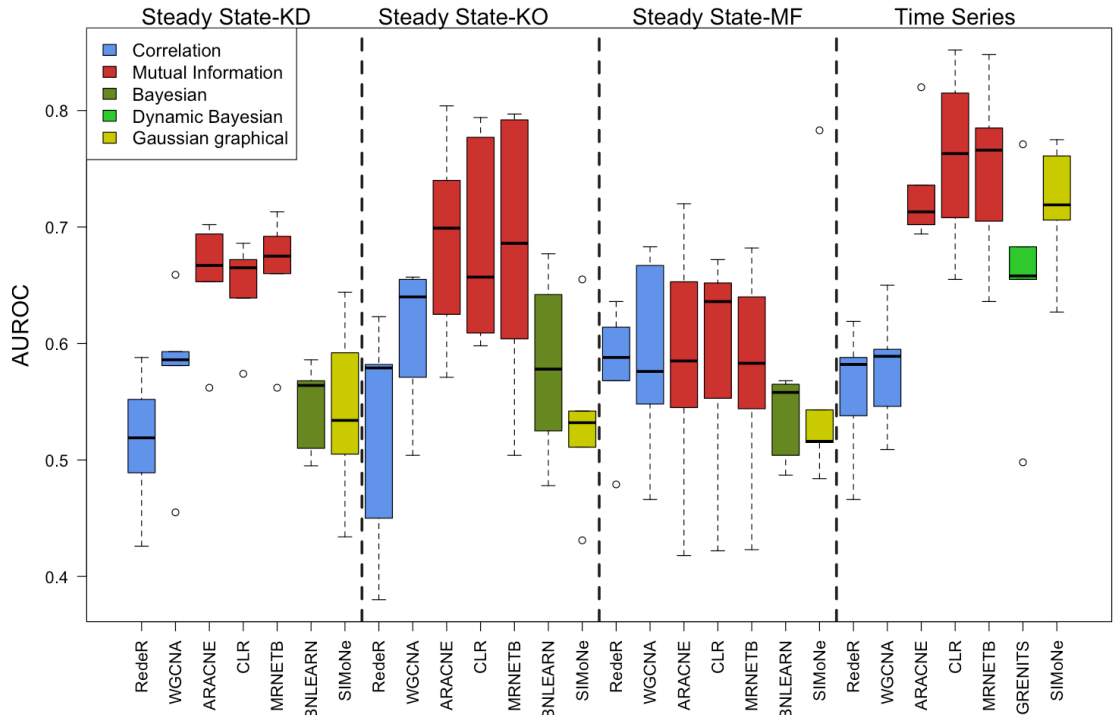


Figure A.3: Performance scores of different network inference approaches using benchmarked DREAM4 challenge *in silico* datasets of size 10. Both steady state (KO-Knockout; KD-Knockdown; MF-Multifactorial) and time series data (Time Series) were used.

when applied to steady state data, in particular the KD and KO experiments with median AUROC scores of 0.67 and 0.70 respectively. It also gives the fourth best score with time series data, with an AUROC of 0.71. In contrast, the correlation based methods WGCNA performed fourth best (AUROC of 0.64) with KO, and RedeR performed third best (AUROC of 0.59) with MF steady state experiments. SIMoNe performed surprisingly well with time series data (AUROC of 0.72) in compared to various types steady-state datasets. It is interesting to note that with steady state data, Bayesian networks performed comparably to the other methods; however, they performed well with time series data.

A.3 DREAM size 100

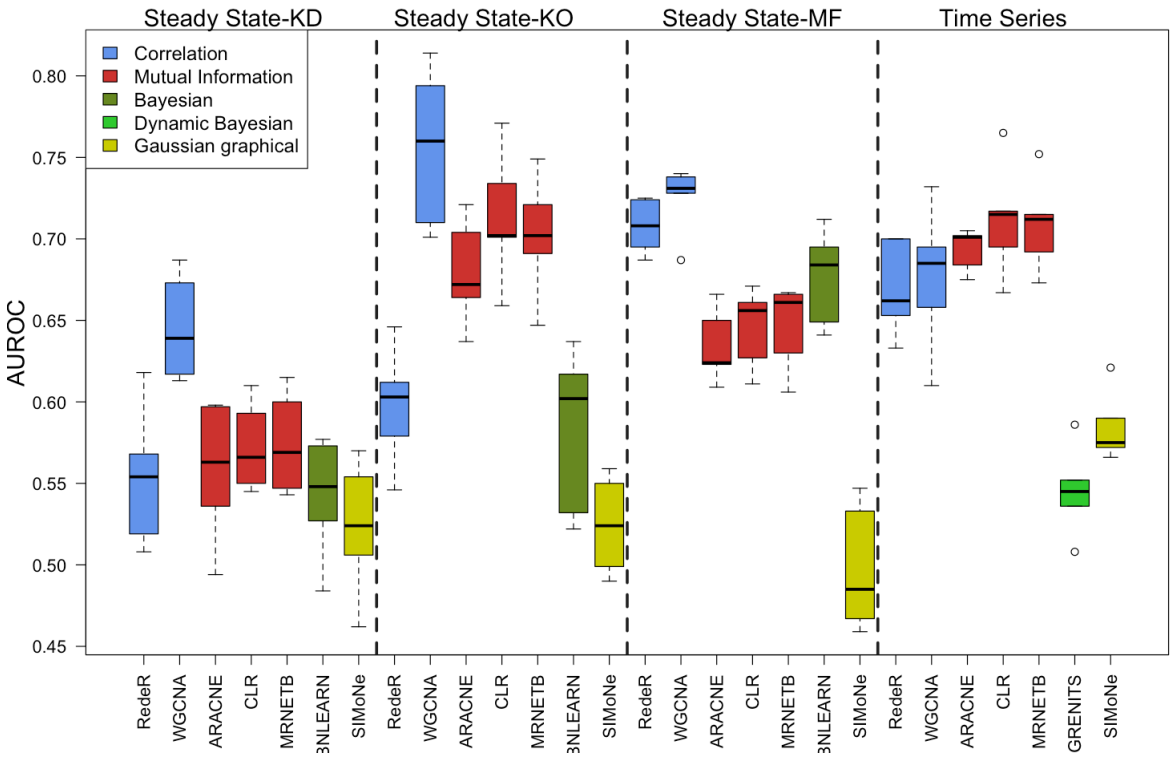


Figure A.4: Performance scores of different network inference approaches using benchmarked DREAM4 challenge *in silico* datasets of size 100. Both steady state (KO-Knockout; KD-Knockdown; MF-Multifactorial) and time series data (Time Series) were used.

In Figure A.4, the boxplots compare the AUROC scores for another medium sized network (100 genes) under similar simulated experimental conditions as that of the size 10 DREAM4 data. It is apparent from these plots that the correlation based networks - in particular WGCNA - performed consistently well in this case for most steady state data tested, and fourth best with time series data. The most striking result to emerge was the performance of the correlation and Bayesian network (BNLEARN) with steady state data (in particular the MF experimental data) where it was superior to all mutual information algorithms. However, the GRENITS and SIMoNe performance was comparatively poor for time series data. Surprisingly, CLR, MRNETB and ARACNE performed well with time series data,

highlighting the power of mutual information and correlation based inference algorithms in using temporal gene expression patterns to predict true regulatory interactions.

A.4 Conclusion from *in silico* experiments

In summary, the above findings from *in silico*-based experiments suggest that none of the network algorithms are able to perform equally across the same type of gene expression data used in terms of AUROC measures. This means that regulatory interactions predicted from each of the network inference algorithms vary, generating heterogeneous GRNs.

A.5 Unique predictions from real gene expression dataset

The unique predicted gene interactions from the consensus models (FCPT) in Chapter 5 using real gene expression dataset from *S.cerevisiae* that are statistically significant (q<0.05) are not included curated SGD database that require further experimental investigation, as illustrated in Table A.1.

Table A.1: Consensus-predicted unique edge-lists those are not common across individual network inference algorithms using real gene expression data from *S.cerevisiae*.

1	YAL054C-YGL078C	41	YDR033W-YLR397C	81	YGR067C-YKL120W
2	YAL054C-YNL209W	42	YDR077W-YJR063W	82	YGR067C-YKR024C
3	YBL003C-YPL051W	43	YDR085C-YBL004W	83	YGR187C-YGR234W
4	YBL004W-YDR033W	44	YDR085C-YDR536W	84	YGR234W-YGR187C
5	YBL004W-YDR085C	45	YDR085C-YLR106C	85	YGR234W-YNL075W
6	YBL075C-YHL016C	46	YDR173C-YHL016C	86	YGR248W-YLR354C
7	YBL075C-YKL009W	47	YDR253C-YLL062C	87	YHL016C-YBL075C
8	YBR026C-YLR153C	48	YDR324C-YBR213W	88	YHL016C-YDL059C
9	YBR072W-YLR354C	49	YDR324C-YER090W	89	YHL016C-YDR173C
10	YBR088C-YGL080W	50	YDR528W-YNL111C	90	YHR005C-YDL024C
11	YBR088C-YPL051W	51	YDR536W-YDR085C	91	YHR005C-YLR328W
12	YBR092C-YIR012W	52	YDR536W-YKL068W-A	92	YHR005C-YPR006C

13	YBR092C-YNL113W	53	YDR536W-YLR164W	93	YHR019C-YOR019W
14	YBR104W-YNL111C	54	YEL040W-YLR300W	94	YHR049W-YMR049C
15	YBR208C-YJL153C	55	YEL057C-YLR048W	95	YHR049W-YOR388C
16	YBR213W-YDR324C	56	YEL057C-YNL301C	96	YHR066W-YMR061W
17	YBR213W-YHR163W	57	YEL070W-YER015W	97	YHR089C-YDL183C
18	YBR249C-YJL148W	58	YEL070W-YLL024C	98	YHR163W-YBR213W
19	YBR296C-YLR354C	59	YEL070W-YMR241W	99	YHR163W-YNL012W
20	YBR299W-YNL237W	60	YER015W-YEL070W	100	YHR179W-YLL061W
21	YCR034W-YDL079C	61	YER015W-YNL281W	101	YHR183W-YLR205C
22	YCR034W-YDL183C	62	YER043C-YDL079C	102	YHR183W-YPL113C
23	YCR034W-YJL088W	63	YER065C-YIR017C	103	YHR196W-YMR145C
24	YCR087C-A-YGR052W	64	YER065C-YJR097W	104	YIL013C-YDL079C
25	YDL021W-YOL136C	65	YER065C-YNL037C	105	YIL045W-YJL153C
26	YDL024C-YHR005C	66	YER090W-YDR324C	106	YIL045W-YLR174W
27	YDL024C-YKR061W	67	YER090W-YLR397C	107	YIL078W-YJL148W
28	YDL051W-YKL106W	68	YER102W-YFL034C-A	108	YIL078W-YML080W
29	YDL059C-YHL016C	69	YER102W-YOR359W	109	YIR012W-YBR092C
30	YDL059C-YOL083W	70	YFL034C-A-YER102W	110	YIR012W-YOL016C
31	YDL079C-YCR034W	71	YGL078C-YAL054C	111	YIR017C-YER065C
32	YDL079C-YER043C	72	YGL080W-YBR088C	112	YJL033W-YMR093W
33	YDL079C-YGR154C	73	YGL080W-YJR016C	113	YJL060W-YNL151C
34	YDL079C-YIL013C	74	YGL080W-YPL266W	114	YJL088W-YCR034W
35	YDL183C-YCR034W	75	YGL205W-YJR070C	115	YJL088W-YOR215C
36	YDL183C-YHR089C	76	YGL205W-YNL151C	116	YJL116C-YGL225W
37	YDL183C-YPL266W	77	YGL225W-YJL116C	117	YJL116C-YLL028W
38	YDL218W-YKL021C	78	YGR052W-YCR087C-A	118	YJL116C-YML080W
39	YDL229W-YOR010C	79	YGR052W-YMR061W	119	YJL148W-YBR249C
40	YDR033W-YBL004W	80	YGR055W-YLL062C	120	YJL148W-YIL078W
121	YJL148W-YLL062C	168	YLR153C-YBR026C	218	YNL111C-YOR134W
122	YJL153C-YBR208C	169	YLR164W-YDR536W	219	YNL113W-YBR092C
123	YJL153C-YIL045W	170	YLR174W-YIL045W	220	YNL142W-YNL301C
124	YJL153C-YLR354C	171	YLR177W-YKL081W	221	YNL144C-YPL051W
125	YJL153C-YOR180C	172	YLR177W-YKL125W	222	YNL151C-YGL205W
126	YJL200C-YKL188C	173	YLR186W-YPL131W	223	YNL151C-YJL060W
127	YJR016C-YGL080W	174	YLR205C-YHR183W	224	YNL209W-YAL054C
128	YJR016C-YOR180C	175	YLR205C-YKL125W	225	YNL237W-YBR299W
129	YJR043C-YOL064C	176	YLR205C-YPL030W	226	YNL281W-YER015W
130	YJR047C-YNL072W	177	YLR216C-YML047C	227	YNL301C-YEL057C
131	YJR063W-YDR077W	178	YLR223C-YLL062C	228	YNL301C-YNL142W
132	YJR070C-YGL205W	179	YLR300W-YEL040W	229	YNL327W-YOR233W
133	YJR097W-YER065C	180	YLR328W-YHR005C	230	YNR044W-YOR222W
134	YKL009W-YBL075C	181	YLR346C-YKR061W	231	YNR050C-YKL133C
135	YKL021C-YDL218W	182	YLR346C-YOR347C	232	YOL016C-YIR012W
136	YKL068W-A-YDR536W	183	YLR354C-YBR072W	233	YOL064C-YJR043C
137	YKL068W-YMR175W	184	YLR354C-YBR296C	234	YOL083W-YDL059C
138	YKL081W-YLR177W	185	YLR354C-YGR248W	235	YOL136C-YDL021W
139	YKL106W-YDL051W	186	YLR354C-YJL153C	236	YOR010C-YDL229W
140	YKL107W-YML047C	187	YLR354C-YML047C	237	YOR010C-YMR217W
141	YKL120W-YGR067C	188	YLR354C-YMR145C	238	YOR019W-YHR019C

142	YKL125W-YLR177W	189	YLR397C-YDR033W	239	YOR134W-YNL111C
143	YKL125W-YLR205C	190	YLR397C-YER090W	240	YOR180C-YJL153C
144	YKL128C-YOR222W	191	YLR432W-YMR093W	241	YOR180C-YJR016C
145	YKL133C-YLR092W	192	YLR432W-YOR271C	242	YOR180C-YPR074C
146	YKL133C-YNR050C	193	YML047C-YKL107W	243	YOR215C-YJL088W
147	YKL188C-YJL200C	194	YML047C-YLR216C	244	YOR222W-YKL128C
148	YKR024C-YGR067C	195	YML047C-YLR354C	245	YOR222W-YNR044W
149	YKR061W-YDL024C	196	YML047C-YOR271C	246	YOR233W-YNL327W
150	YKR061W-YLR346C	197	YML080W-YIL078W	247	YOR271C-YLR432W
151	YKR076W-YLL061W	198	YML080W-YJL116C	248	YOR271C-YML047C
152	YLL008W-YMR145C	199	YML080W-YLL062C	249	YOR347C-YLR346C
153	YLL024C-YEL070W	200	YMR049C-YHR049W	250	YOR348C-YMR175W
154	YLL028W-YJL116C	201	YMR061W-YGR052W	251	YOR348C-YPL240C
155	YLL028W-YLR142W	202	YMR061W-YHR066W	252	YOR359W-YER102W
156	YLL061W-YHR179W	203	YMR093W-YJL033W	253	YOR359W-YLL062C
157	YLL061W-YKR076W	204	YMR093W-YLR432W	254	YOR388C-YHR049W
158	YLL062C-YDR253C	205	YMR145C-YHR196W	255	YPL030W-YLR205C
159	YLL062C-YGR055W	206	YMR145C-YLL008W	256	YPL051W-YBL003C
160	YLL062C-YJL148W	207	YMR145C-YLR354C	257	YPL051W-YBR088C
161	YLL062C-YLR223C	208	YMR175W-YKL068W	258	YPL051W-YNL144C
162	YLL062C-YML080W	209	YMR175W-YOR348C	259	YPL081W-YPL131W
163	YLL062C-YOR359W	210	YMR217W-YOR010C	260	YPL113C-YHR183W
164	YLR048W-YEL057C	211	YMR241W-YEL070W	261	YPL131W-YLR186W
165	YLR092W-YKL133C	212	YNL012W-YHR163W	262	YPL131W-YPL081W
166	YLR106C-YDR085C	213	YNL037C-YER065C	263	YPL240C-YOR348C
167	YLR142W-YLL028W	214	YNL072W-YJR047C	264	YPL266W-YDL183C
168	YLR153C-YBR026C	215	YNL075W-YGR234W	265	YPL266W-YGL080W
169	YLR164W-YDR536W	216	YNL111C-YBR104W	266	YPR006C-YHR005C
170	YLR174W-YIL045W	217	YNL111C-YDR528W	267	YPR074C-YOR180C

Appendix B

In this appendix, supplementary material showing functional modules emerged in GO enrichment analysis for biological process (BP) that are not top ranked but yet show biologically meaningful results from different network algorithms at cluster module sizes (4, 8, 12 and 16) using in silico (Chapter 4), and real expression data (Chapter 5) are presented.

B.1 *In silico* expression data

Table B.1: Functional modules predicted from RedeR for different cluster module sizes that show statistically significant (p<0.05) association with biological process (BP) in GO enrichment analysis for size 100 data. Count signifies total number of genes within a module that are associated with a BP.

RedeR					
Module	**GO.ID**	**Process**	**Count**	**P.value**	**Module.score**
		4 Modules			
M1	GO:0009987	cellular process	30	0.013328	1.875
M2	GO:0009987	cellular process	43	0.0038657	2.413
M3	GO:0044763	single-organism cellular process	15	0.0069951	2.155
M4	GO:0044238	primary metabolic process	9	0.022303	1.652
		8 Modules			
M1	GO:0009987	cellular process	17	0.031685	1.499
M2	GO:0009987	cellular process	25	0.039677	1.401
M7	GO:0044763	single-organism cellular process	13	0.026261	1.581
M8	GO:0044238	primary metabolic process	9	0.022303	1.652
		12 Modules			
M1	GO:0009987	cellular process	17	0.031685	1.499
M10	GO:0044763	single-organism cellular process	13	0.026261	1.581
M11	GO:0044238	primary metabolic process	9	0.022303	1.652
M2	GO:0044763	single-organism cellular process	16	0.0081977	2.086
		16 Modules			

M1	GO:0044249	cellular biosynthetic process	10	0.0019637	2.707
M12	GO:0044763	single-organism cellular process	13	0.026261	1.581
M13	GO:0044249	cellular biosynthetic process	7	0.049355	1.307
M14	GO:0044238	primary metabolic process	9	0.022303	1.652

Table B.2: Functional modules predicted from WGCNA for different cluster module sizes that show statistically significant (p<0.05) association with biological process (BP) in GO enrichment analysis for size 100 data.

WGCNA					
Module	**GO.ID**	**Process**	**Count**	**P.value**	**Module.score**
		4 modules			
M1	GO:0009987	cellular process	50	0.0005464	3.262
M2	GO:0009987	cellular process	33	0.0081633	2.088
M3	GO:0044249	cellular biosynthetic process	9	0.032565	1.487
		8 modules			
M1	GO:0050896	response to stimulus	7	0.024363	1.613
M2	GO:0044699	single-organism process	13	0.012128	1.916
M4	GO:0044281	small molecule metabolic process	6	0.026879	1.571
M5	GO:0044699	single-organism process	13	0.027023	1.568
M6	GO:0044249	cellular biosynthetic process	9	0.032565	1.487
M8	GO:0009987	cellular process	25	0.0098082	2.008
		12 Modules			
M11	GO:0009987	cellular process	24	0.011809	1.928
M12	GO:0044763	single-organism cellular process	8	0.016608	1.78
M2	GO:0044699	single-organism process	12	0.017671	1.753
M9	GO:0044281	small molecule metabolic process	6	0.00014107	3.851
		16 Modules			
M14	GO:0009987	cellular process	17	0.041987	1.377
M2	GO:0044249	cellular biosynthetic process	9	0.0015962	2.797
M3	GO:0008152	metabolic process	18	0.010944	1.961

Table B.3: Functional modules predicted from SIMoNE for different cluster module sizes that show statistically significant (p<0.05) association with biological process (BP) in GO enrichment analysis for size 100 data.

SIMoNE					
Module	**GO.ID**	**Process**	**Count**	**P.value**	**Module.score**
		4 Modules			
M1	GO:0009987	cellular process	54	0.0047656	2.322
M2	GO:0044238	primary metabolic process	12	0.021288	1.672
M3	GO:0009987	cellular process	22	0.017778	1.75
M4	GO:0044249	cellular biosynthetic process	8	0.0060416	2.219
		8 Modules			
M1	GO:0044699	single-organism process	19	0.029014	1.537
M2	GO:0044249	cellular biosynthetic process	10	0.00056502	3.248
M7	GO:0009987	cellular process	22	0.017778	1.75
M8	GO:0044249	cellular biosynthetic process	8	0.0060416	2.219
		12 Modules			
M11	GO:0044249	cellular biosynthetic process	8	0.0060416	2.219
M2	GO:0044249	cellular biosynthetic process	9	0.0015962	2.797
M3	GO:0044699	single-organism process	20	0.015156	1.819
		16 Modules			
M14	GO:0044249	cellular biosynthetic process	8	0.0060416	2.219
M2	GO:0034645	cellular macromolecule biosynthetic process	8	0.00065753	3.182
M4	GO:0044699	single-organism process	18	0.016891	1.772

Table B.4: Functional modules predicted from RedeR for different cluster module sizes that show statistically significant (p<0.05) association with biological process (BP) in GO enrichment analysis for size 500 data.

RedeR					
Module	**GO.ID**	**Process**	**Count**	**P.value**	**Module.score**
		4 Modules			
M1	GO:0009987	cellular process	101	0.00012642	3.898
M2	GO:0009987	cellular process	214	6.18E-10	9.209
M3	GO:0009987	cellular process	97	1.21E-07	6.918
M4	GO:0009987	cellular process	57	2.20E-05	4.657
		8 Modules			
M1	GO:0009987	cellular process	81	0.00213	2.672
M2	GO:0009987	cellular process	173	2.33E-09	8.633
M3	GO:0009987	cellular process	97	1.21E-07	6.918

M4	GO:0009987	cellular process	39	0.00059569	3.225
M5	GO:0044763	single-organism cellular process	14	0.00169	2.772
M6	GO:0009987	cellular process	18	0.025147	1.6
M7	GO:0044699	single-organism process	43	0.00022829	3.642
M8	GO:0044710	single-organism metabolic process	6	0.01356	1.868
		12 Modules			
M1	GO:0008152	metabolic process	42	0.039675	1.401
M10	GO:0044699	single-organism process	43	0.00022829	3.642
M11	GO:0009987	cellular process	17	0.035313	1.452
M12	GO:0044710	single-organism metabolic process	6	0.01356	1.868
M2	GO:0009987	cellular process	148	4.08E-08	7.39
M3	GO:0009987	cellular process	25	0.03536	1.451
M4	GO:0009987	cellular process	47	0.00016491	3.783
M5	GO:0009987	cellular process	39	0.00059569	3.225
M6	GO:0009987	cellular process	50	0.00057404	3.241
M7	GO:0009987	cellular process	35	0.011763	1.929
M8	GO:0044763	single-organism cellular process	14	0.00169	2.772
		16 Modules			
M1	GO:0008152	metabolic process	42	0.039675	1.401
M11	GO:0044699	single-organism process	43	0.00022829	3.642
M12	GO:0009987	cellular process	17	0.035313	1.452
M14	GO:0044699	single-organism process	26	0.00090634	3.043
M15	GO:0044710	single-organism metabolic process	6	0.01356	1.868
M2	GO:0009987	cellular process	121	2.25E-07	6.649
M3	GO:0009987	cellular process	25	0.03536	1.451
M4	GO:0009987	cellular process	21	0.012808	1.893
M5	GO:0009987	cellular process	39	0.00059569	3.225
M6	GO:0009987	cellular process	46	0.0011195	2.951
M7	GO:0009987	cellular process	30	0.016388	1.785
M8	GO:0009987	cellular process	26	0.0065528	2.184
M9	GO:0044763	single-organism cellular process	14	0.00169	2.772

Table B.5: Functional modules predicted from WGCNA for different cluster module sizes that show statistically significant (p<0.05) association with biological process (BP) in GO enrichment analysis for size 500 data.

WGCNA					
Module	**GO.ID**	**Process**	**Count**	**P.value**	**Module.score**
		4 Modules			
M1	GO:0009987	cellular process	242	1.12E-12	11.952
M2	GO:0009987	cellular process	30	0.0024823	2.605
M3	GO:0009987	cellular process	180	6.70E-10	9.174
M4	GO:0044699	single-organism process	18	0.021332	1.671
		8 Modules			
M1	GO:0009987	cellular process	72	0.0010803	2.966
M2	GO:0009987	cellular process	118	1.85E-08	7.733
M3	GO:0009987	cellular process	30	0.0024823	2.605
M4	GO:0009987	cellular process	97	4.07E-06	5.39
M5	GO:0009987	cellular process	60	0.00038522	3.414
M6	GO:0044699	single-organism process	18	0.021332	1.671
M7	GO:0009987	cellular process	52	0.017013	1.769
M8	GO:0008152	metabolic process	24	0.006315	2.2
		12 Modules			
M1	GO:0009987	cellular process	37	0.044512	1.352
M10	GO:0044699	single-organism process	18	0.021332	1.671
M11	GO:0009987	cellular process	53	0.014391	1.842
M12	GO:0008152	metabolic process	24	0.006315	2.2
M2	GO:0009987	cellular process	35	0.0071474	2.146
M3	GO:0009987	cellular process	83	1.16E-06	5.934
M4	GO:0009987	cellular process	28	0.0045984	2.337
M5	GO:0009987	cellular process	26	0.014872	1.828
M6	GO:0009987	cellular process	66	0.00017649	3.753
M7	GO:0009987	cellular process	32	0.011286	1.947
M8	GO:0009987	cellular process	60	0.00038522	3.414
		16 Modules			
M1	GO:0065007	biological regulation	16	0.028706	1.542
M10	GO:0009987	cellular process	60	0.00038522	3.414
M12	GO:0044237	cellular metabolic process	9	0.027596	1.559
M13	GO:0044699	single-organism process	18	0.021332	1.671
M14	GO:0009987	cellular process	53	0.014391	1.842
M15	GO:0008152	metabolic process	24	0.006315	2.2
M16	GO:0044710	single-organism metabolic process	7	0.0035082	2.455
M3	GO:0009987	cellular process	35	0.0071474	2.146
M4	GO:0009987	cellular process	83	1.16E-06	5.934
M6	GO:0009987	cellular process	26	0.014872	1.828
M7	GO:0009987	cellular process	51	0.0004553	3.342

M8	GO:0009987	cellular process	32	0.011286	1.947
M9	GO:0009987	cellular process	22	0.015901	1.799

Table B.6: Functional modules predicted from SIMoNE for different cluster module sizes that show statistically significant (p<0.05) association with biological process (BP) in GO enrichment analysis for size 500 data.

SIMoNe					
Module	**GO.ID**	**Process**	**Count**	**P.value**	**Module.score**
		4 Modules			
M1	GO:0009987	cellular process	390	9.14E-23	22.039
M2	GO:0050794	regulation of cellular process	12	0.045754	1.34
M3	GO:0009987	cellular process	32	0.019077	1.719
M4	GO:0044699	single-organism process	21	0.0018785	2.726
		8 Modules			
M1	GO:0009987	cellular process	299	6.17E-19	18.21
M2	GO:0050794	regulation of cellular process	12	0.045754	1.34
M3	GO:0009987	cellular process	22	0.01644	1.784
M4	GO:0009987	cellular process	32	0.019077	1.719
M5	GO:0044699	single-organism process	28	0.0042347	2.373
M6	GO:0044699	single-organism process	21	0.0018785	2.726
M7	GO:0009987	cellular process	27	0.0040442	2.393
M8	GO:0044699	single-organism process	17	0.032582	1.487
		12 Modules			
M1	GO:0009987	cellular process	223	5.43E-13	12.266
M10	GO:0044699	single-organism process	21	0.0018785	2.726
M11	GO:0009987	cellular process	27	0.0040442	2.393
M12	GO:0044699	single-organism process	17	0.032582	1.487
M2	GO:0009987	cellular process	32	0.01024	1.99
M3	GO:0044238	primary metabolic process	20	0.00036663	3.436
M4	GO:0050794	regulation of cellular process	12	0.045754	1.34
M5	GO:0009987	cellular process	18	0.044906	1.348
M6	GO:0009987	cellular process	22	0.01644	1.784
M7	GO:0009987	cellular process	29	0.025759	1.589
M8	GO:0044237	cellular metabolic process	10	0.037272	1.429
M9	GO:0044699	single-organism process	28	0.0042347	2.373
		16 Modules			
M1	GO:0009987	cellular process	191	2.59E-10	9.587
M10	GO:0065007	biological regulation	9	0.016835	1.774
M11	GO:0044763	single-organism cellular process	12	0.025992	1.585
M13	GO:0044699	single-organism process	21	0.0018785	2.726
M14	GO:0009987	cellular process	27	0.0040442	2.393
M15	GO:0044699	single-organism process	17	0.032582	1.487

M16	GO:0008152	metabolic process	14	0.019517	1.71
M2	GO:0009987	cellular process	30	0.0025205	2.599
M3	GO:0044238	primary metabolic process	20	0.00036663	3.436
M4	GO:0051704	multi-organism process	7	0.0033724	2.472
M5	GO:0009987	cellular process	18	0.044906	1.348
M6	GO:0009987	cellular process	22	0.01644	1.784
M7	GO:0009987	cellular process	28	0.032471	1.489
M8	GO:0044237	cellular metabolic process	10	0.037272	1.429

B.2 Real expression data

The real gene expression data of *S.cerevisiae* (Chapter 5) in the tables (B.7-B.9) illustrated above, show that many modular networks from different network algorithms, are associated with common biological processes. However, the module score helps determine quantitatively, which modules are highly associated with a particular functional process.

Table B.7: Functional modules predicted from RedeR for different cluster module sizes that show statistically significant (p<0.05) association with biological process (BP) in GO enrichment analysis with real gene expresson data. Count signifies total number of genes within a module that are associated with a BP.

RedeR					
Module	**GO.ID**	**Process**	**Count**	**P.value**	**Module.score**
		4 Modules			
M1	GO:0009987	cellular process	99	0.02062	1.686
M2	GO:0044710	single-organism metabolic process	39	0.005315	2.274
M3	GO:0009987	cellular process	59	0.045812	1.339
M4	GO:0009987	cellular process	72	0.0031065	2.508
		8 Modules			
M1	GO:1901564	organonitrogen compound metabolic process	12	0.045243	1.344
M2	GO:0055114	oxidation-reduction process	8	0.01292	1.889
M3	GO:0009987	cellular process	61	0.02188	1.66
M4	GO:0009987	cellular process	59	0.045812	1.339
M5	GO:0044699	single-organism process	16	0.01704	1.769
M6	GO:0055114	oxidation-reduction process	9	0.0048029	2.318
M7	GO:0008152	metabolic process	48	0.00028985	3.538
M8	GO:0009987	cellular process	25	0.01483	1.829
		12 Modules			

M1	GO:0042254	ribosome biogenesis	6	0.02575	1.589
M10	GO:0044281	small molecule metabolic process	9	0.00014201	3.848
M11	GO:0008152	metabolic process	48	0.00028985	3.538
M12	GO:0009987	cellular process	25	0.01483	1.829
M2	GO:0055114	oxidation-reduction process	8	0.01292	1.889
M3	GO:1901564	organonitrogen compound metabolic process	7	0.043432	1.362
M4	GO:0009987	cellular process	61	0.02188	1.66
M5	GO:0009987	cellular process	51	0.0092284	2.035
M8	GO:0055114	oxidation-reduction process	9	0.0048029	2.318
		16 Modules			
M1	GO:0042254	ribosome biogenesis	6	0.02575	1.589
M11	GO:0044281	small molecule metabolic process	9	0.00014201	3.848
M12	GO:0008152	metabolic process	39	0.0029889	2.524
M14	GO:0009987	cellular process	16	0.048776	1.312
M15	GO:0044238	primary metabolic process	9	0.044606	1.351
M2	GO:0055114	oxidation-reduction process	8	0.01292	1.889
M3	GO:1901564	organonitrogen compound metabolic process	7	0.043432	1.362
M4	GO:1901360	organic cyclic compound metabolic process	35	9.34E-06	5.03
M5	GO:0009987	cellular process	51	0.0092284	2.035
M7	GO:0044238	primary metabolic process	15	0.0032435	2.489
M9	GO:0055114	oxidation-reduction process	7	0.011097	1.955

Table B.8: Functional modules predicted from WGCNA for different cluster module sizes that show statistically significant (p<0.05) association to biological process (BP) in GO enrichment analysis with real gene expresson data.

WGCNA					
Module	**GO.ID**	**Process**	**Count**	**P.value**	**Module.score**
		4 modules			
M1	GO:0006807	nitrogen compound metabolic process	57	0.00014146	3.849
M2	GO:0009987	cellular process	104	0.00088951	3.051
M3	GO:0044281	small molecule metabolic process	23	0.041258	1.384
M4	GO:0044699	single-organism process	31	0.0014306	2.844
		8 Modules			
M1	GO:0006807	nitrogen compound metabolic process	45	0.010534	1.977
M2	GO:0009987	cellular process	83	0.0019025	2.721

M4	GO:0006807	nitrogen compound metabolic process	12	0.00014087	3.851
M5	GO:0044281	small molecule metabolic process	22	0.0095026	2.022
M8	GO:0044699	single-organism process	21	0.0016517	2.782
		12 Modules			
M1	GO:0043170	macromolecule metabolic process	47	0.047859	1.32
M10	GO:0044699	single-organism process	17	0.0019065	2.72
M12	GO:0000749	response to pheromone involved in conjugation with cellular fusion	7	5.92E-11	10.228
M2	GO:0009987	cellular process	15	0.041922	1.378
M3	GO:0009987	cellular process	71	0.00042137	3.375
M4	GO:0044699	single-organism process	16	0.040331	1.394
M6	GO:0044281	small molecule metabolic process	17	0.012511	1.903
		16 Modules			
M1	GO:0006725	cellular aromatic compound metabolic process	35	7.74E-05	4.112
M11	GO:0009987	cellular process	64	0.00029501	3.53
M12	GO:0044281	small molecule metabolic process	9	0.00017853	3.748
M13	GO:0044699	single-organism process	18	0.0011432	2.942
M14	GO:0044281	small molecule metabolic process	15	0.042452	1.372
M16	GO:0000749	response to pheromone involved in conjugation with cellular fusion	6	2.61E-10	9.583
M4	GO:0044763	single-organism cellular process	16	0.041899	1.378
M8	GO:0006796	phosphate-containing compound metabolic process	6	0.01928	1.715

Table B.9: Functional modules predicted from SIMoNE for different cluster module sizes that show statistically significant (p<0.05) association to biological process (BP) in GO enrichment analysis with real gene expresson data.

SIMoNe					
Module	**GO.ID**	**Process**	**Count**	**P.value**	**Module.score**
		4 Modules			
M1	GO:0006807	nitrogen compound metabolic process	49	0.004724	2.326
M2	GO:0044238	primary metabolic process	20	0.028659	1.543
M3	GO:0044281	small molecule metabolic process	12	0.00019442	3.711
M4	GO:0009987	cellular process	170	0.017545	1.756
		8 Modules			
M1	GO:0006807	nitrogen compound metabolic process	42	0.0056137	2.251
M3	GO:0008152	metabolic process	97	0.0031956	2.495
M4	GO:0044699	single-organism process	70	0.0013771	2.861
M7	GO:0044281	small molecule metabolic process	7	0.0027278	2.564
		12 modules			
M1	GO:0006807	nitrogen compound metabolic process	42	0.0010776	2.968
M3	GO:0071444	cellular response to pheromone	6	0.0067577	2.17
M4	GO:0009987	cellular process	76	0.0065446	2.184
M5	GO:0009987	cellular process	77	0.0016435	2.784
		16 Modules			
M1	GO:0006807	nitrogen compound metabolic process	48	0.0002311	3.636
M4	GO:0009987	cellular process	49	0.010335	1.986
M6	GO:0009987	cellular process	54	0.034146	1.467
M9	GO:0055114	oxidation-reduction process	14	0.022185	1.654

Bibliography

Albert R, B. AL, 2002. Statistical Mechanics of Complex Networks. *Reviews of Modern Physics*, 74, pp.47–97.

Androulakis, I., Yang, E. & Almon, R., 2007. Analysis of Time-Series Gene Expression Data: Methods, Challenges, and Opportunities. *Annu. Rev. Biomed. Eng.*, pp.1–23.

Bansal, M. et al., 2007. How to infer gene networks from expression profiles. *Molecular systems biology*, 3(78), p.78.

Bansal, M., Della Gatta, G. & di Bernardo, D., 2006. Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics (Oxford, England)*, 22(7), pp.815–22.

Barabási, A.-L. & Oltvai, Z.N., 2004. Network biology: understanding the cell's functional organization. *Nature reviews. Genetics*, 5(2), pp.101–13.

Barenco, M. et al., 2006. Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome biology*, 7(3), p.R25.

Barton, S.J. et al., 2013. Correction of unexpected distributions of P values from analysis of whole genome arrays by rectifying violation of statistical assumptions. *BMC Genomics*, 14(1), p.161.

Beal, M.J. et al., 2005. A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics (Oxford, England)*, 21(3), pp.349–56.

Bennett, M.R. et al., 2008. Metabolic gene regulation in a dynamically changing environment. *Nature*, 454(7208), pp.1119–1122.

di Bernardo, D. et al., 2005. Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. *Nature biotechnology*, 23(3), pp.377–83.

Bilal, E. et al., 2015. A crowd-sourcing approach for the construction of species-specific cell signaling networks. *Bioinformatics*, 31(4), pp.484–491.

Blais, A. & Dynlacht, B.D., 2005. Constructing transcriptional regulatory networks. *Genes & development*, 19(13), pp.1499–511.

Bland, M., 2013. Do Baseline P-Values Follow a Uniform Distribution in Randomised Trials? *PLoS ONE*, 8(10), pp.1–5.

Bolshakova, N. & Azuaje, F., 2003. Cluster validation techniques for genome expression data. *Signal Processing*, 83(4), pp.825–833.

Borenstein, M. & Rothstein, H., 2007. Introduction to Meta-Analysis. , (C). Available at: www.Meta-Analysis.com.

Borovkov, K., 2003. Markov chains. In *Elements of Stochastic Modelling*. WORLD SCIENTIFIC, pp. 75–128.

Breslow, D., Cameron, D. & Collins, S., 2008. A comprehensive strategy enabling high-

resolution functional analysis of the yeast genome. *Nature methods*, 5(8), pp.711–718.

Van den Bulcke, T. et al., 2006. SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC bioinformatics*, 7, p.43.

Butte, a J. & Kohane, I.S., 2000. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 426, pp.418–29.

Carey, M.F., Peterson, C.L. & Smale, S.T., 2009. Chromatin immunoprecipitation (ChIP). *Cold Spring Harbor protocols*, 2009(9), p.pdb.prot5279.

Cassman, M., Arkin, A. & Doyle, F., 2007. *Systems biology: International research and development*, Springer.

Castro, M. a a et al., 2012. RedeR: R/Bioconductor package for representing modular structures, nested networks and multiple levels of hierarchical associations. *Genome biology*, 13(4), p.R29.

Chang, L.-C. et al., 2013. Meta-analysis methods for combining multiple expression profiles: comparisons, statistical characterization and an application guideline. *BMC bioinformatics*, 14(vi), p.368.

Chen, Z., 2011. Is the weighted z-test the best method for combining probabilities from independent tests? *Journal of Evolutionary Biology*, 24(4), pp.926–930.

Cherry, J.M. et al., 2012. Saccharomyces Genome Database: the genomics resource of budding yeast. *Nucleic acids research*, 40(Database issue), pp.D700–5.

Chiquet, J. et al., 2009. SIMoNe: Statistical Inference for MOdular NEtworks. *Bioinformatics (Oxford, England)*, 25(3), pp.417–8.

Choi, J.K. et al., 2003. Combining multiple microarray studies and modeling interstudy variation. *Bioinformatics*, 19(Suppl 1), pp.i84–i90.

Costa, I.G., Carvalho, F.D.A.T. De & Souto, M.C.P. De, 2004. Comparative analysis of clustering methods for gene expression time course data. *Society*, 631, pp.623–631.

Csardi, G., 2010. Package "igraph." *http://igraph.sourceforge.net*.

D'haeseleer, P., D, P. & D'haeseleer, P., 2005. How does gene expression clustering work? *Nature biotechnology*, 23(12), pp.1499–501.

Dabney A, Storey JD, W.G., 2013. qvalue: Q-value estimation for false discovery rate control. , p.R package version 1.28.0.

Deeks, J., Higgins, J. & Altman, D., 2008. Analysing Data and Undertaking Meta-Analyses. *Cochrane Handbook for …*, pp.1–43.

DerSimonian, R. & Laird, N., 1986. Meta-analysis in clinical trials. *Controlled clinical trials*, 7(3), pp.177–88.

Desvergne, B., Michalik, L. & Wahli, W., 2006. Transcriptional regulation of metabolism. *Physiological reviews*, 86, pp.465–514.

Donetti, L. & Muñoz, M. a, 2004. Detecting network communities: a new systematic and efficient algorithm. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(10), p.P10012.

Edwards, D., 2000. *Introduction to graphical modelling*, John Wiley & Sons, Ltd.

Eisen, M. & Spellman, P., 1998. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the ...*, (22), pp.12930–12933.

Erdmann, E., 2011. Strengths and Drawbacks of Voting Methods for Political Elections. *d.umn.edu*.

Faith, J.J. et al., 2007. Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS biology*, 5(1), p.e8.

Fattah, P. a, 2013. Clustering Validation. In pp. 1–4.

Fawcett, T., 2006. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), pp.861–874.

Fisher, R.., 1932. Statistical Methods for Research Workers. *Oliver and Boyd, Edinburgh.*

Friedman, N. et al., 2000. Using Bayesian networks to analyze expression data. *Journal of computational biology : a journal of computational molecular cell biology*, 7(3-4), pp.601–20.

Gasch, A.P. & Werner-Washburne, M., 2002. The genomics of yeast responses to environmental stress and starvation. *Functional & integrative genomics*, 2(4-5), pp.181–92.

Geistlinger, L., 2008. Detection of differentially regulated genes : Course Analysis Stouffer Merge and Time Course Analysis. *Bioinformatics (Oxford, England)*, 00(00), pp.1–6.

Gevaert, O., Van Vooren, S. & De Moor, B., 2007. A framework for elucidating regulatory networks based on prior information and expression data. *Annals of the New York Academy of Sciences*, 1115, pp.240–8.

Goldstein, D., 2005. Comparison of meta-analysis to combined analysis of a replicated microarray study. *Meta-analysis and Combining Information in Genetics*, pp.1–29.

Greenfield, A. et al., 2010. DREAM4: Combining genetic and dynamic information to identify biological networks and dynamical models. *PloS one*, 5(10), p.e13397.

Gutteridge, A. et al., 2010. Nutrient control of eukaryote cell growth: a systems biology study in yeast. *BMC biology*, 8, p.68.

Haiyuan Yu, Nicholas M Luscombe, J.Q. and M.G., 2003. Genomic analysis of gene expression relationships in transcriptional regulatory networks. *Trends in genetics : TIG*, 19(8), pp.417–22.

Handl, J., Knowles, J. & Kell, D.B., 2005. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15), pp.3201–3212.

Harbison, C.T. et al., 2004. Transcriptional regulatory code of a eukaryotic genome. *Nature*, pp.99–104.

Harris, J.E. et al., 2008. Publishing nutrition research: a review of nonparametric methods, part 3. *Journal of the American Dietetic Association*, 108(9), pp.1488–96.

Hartemink, A. & Gifford, D., 2001. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pacific Symposium on Biocomputing*, pp.422–433.

Hatzimanikatis, V. et al., 2004. Metabolic networks: enzyme function and metabolite structure. *Current opinion in structural biology*, 14(3), pp.300–6.

Hecker, M. et al., 2009. Gene regulatory network inference: data integration in dynamic models-a review. *Bio Systems*, 96(1), pp.86–103.

Hennig, C., 2013. fpc: Flexibile procedures for clustering. *Book fpc: Flexibile procedures for clustering*, 1, pp.1–122.

Hess, A. & Iyer, H., 2007. Fisher's combined p-value for detecting differentially expressed genes using Affymetrix expression arrays. *BMC genomics*, 8, p.96.

Hung, H.M.J. et al., 1997. The Behavior of the P-Value When the Alternative Hypothesis is True. *Biometrics*, 53(1), p.11.

Ion Mandoiu, A.Z., 2008. *Bioinformatics Algorithms: Techniques and Applications*, John Wiley & Sons, Ltd.

Joshi, A., Beck, Y. & Michoel, T., 2014. Multi-species network inference improves gene regulatory network reconstruction for early embryonic development in Drosophila. *arXiv preprint arXiv:1407.6554*, pp.1–21.

Kauffman, S. a, 1969. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3), pp.437–67.

Khatri, P. & Drăghici, S., 2005. Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics (Oxford, England)*, 21(18), pp.3587–95.

Kim, S.Y., Imoto, S. & Miyano, S., 2003. Inferring gene networks from time series microarray data using dynamic Bayesian networks. *Briefings in bioinformatics*, 4(3), pp.228–35.

Knijnenburg, T. a et al., 2009. Fewer permutations, more accurate P-values. *Bioinformatics (Oxford, England)*, 25(12), pp.i161–8.

Kumar, L. & Futschik, M., 2007. Mfuzz : A software package for soft clustering of microarray data Bioinformation. *Bioinformation*, pp.5–7.

Langfelder, P. & Horvath, S., 2008. WGCNA: an R package for weighted correlation network analysis. *BMC bioinformatics*, 9, p.559.

Li, Y. & Ghosh, D., 2014. Meta-analysis based on weighted ordered P-values for genomic data with heterogeneity. *BMC bioinformatics*, 15(1), p.226.

Liang, S., Fuhrman, S. & Somogyi, R., 1998. REVEAL, a general reverse engineering algorithm for inference of geneticnetwork architectures. *Pacific Symposium on Biocomputing*, 29, pp.18–29.

Ma'ayan, A., 2011. Introduction to Network Analysis in Systems Biology. *Science Signaling*, 4(190), pp.tr5–tr5.

Maetschke, S.R. et al., 2014. Supervised, semi-supervised and unsupervised inference of gene regulatory networks. *Briefings in bioinformatics*, 15(2), pp.195–211.

Marbach, D. et al., 2010. Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences of the United States of America*, 107(14), pp.6286–91.

Marbach, D. et al., 2012. Wisdom of crowds for robust gene network inference. *Nature*

methods, 9(8), pp.796–804.

Margolin, A. a et al., 2006. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC bioinformatics*, 7 Suppl 1, p.S7.

Markowetz, F. & Spang, R., 2007. Inferring cellular networks--a review. *BMC bioinformatics*, 8 Suppl 6, p.S5.

Mendoza, M.R. & Bazzan, A.L.C., 2012. On the Ensemble Prediction of Gene Regulatory Networks: A Comparative Study. *2012 Brazilian Symposium on Neural Networks*, pp.55–60.

Meyer, P.E. et al., 2010. Information-Theoretic Inference of Gene Networks Using Backward Elimination. In *The International Conference on Bioinformatics and Computational Biology*.

Meyer, P.E. et al., 2007. Information-theoretic inference of large transcriptional regulatory networks. *EURASIP journal on bioinformatics & systems biology*, 2007(i), p.79879.

Meyer, P.E., Lafitte, F. & Bontempi, G., 2008. minet: A R/Bioconductor package for inferring large transcriptional networks using mutual information. *BMC bioinformatics*, 9(1), p.461.

Miller, J. & Stagljar, I., 2004. Using the yeast two-hybrid system to identify interacting proteins. *Methods in molecular biology (Clifton, N.J.)*, 261, pp.247–62.

Mitra, K. et al., 2013. Integrative approaches for finding modular structure in biological networks. *Nature reviews. Genetics*, 14(10), pp.719–32.

Mohammed, S., 2013. Comparative analysis of network algorithms to address modularity with gene expression temporal data. In *ACM Conference on Bioinformatics, Computational Biology and Biomedical Informatics Proceedings*. Washington, USA: ACM Digital Library, pp. 876–882.

Mohammed, S., Akman, O.E. & Yang, Z.R., 2014. A consensus approach to predict regulatory interactions. In *2014 7th International Conference on Biomedical Engineering and Informatics*. Dalian, China: IEEE, pp. 769–775.

Morandi, E. et al., 2008. Gene expression time-series analysis of camptothecin effects in U87-MG and DBTRG-05 glioblastoma cell lines. *Molecular cancer*, 7(155), p.66.

Morris, S.B. & DeShon, R.P., 2002. Combining effect size estimates in meta-analysis with repeated measures and independent-groups designs. *Psychological Methods*, 7(1), pp.105–125.

Morrissey, E.R., 2013. GRENITS:Gene Regulatory Network Inference Using Time Series. *Systems Biology Doctoral Training Centre*.

Needham, C.J. et al., 2007. A primer on learning in Bayesian networks for computational biology. *PLoS computational biology*, 3(8), p.e129.

Ness, S., 2006. Basic microarray analysis. *Bioinformatics and Drug Discovery*, 316.

Newman, M., 2004. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6), p.066133.

Newman, M. & Girvan, M., 2004. Finding and evaluating community structure in networks. *Physical Review E*, 69(2), p.026113.

Nichols, T.E. & Holmes, A.P., 2002. Nonparametric permutation tests for functional neuroimaging: a primer with examples. *Human brain mapping*, 15(1), pp.1–25.

Niida, A. et al., 2010. A novel meta-analysis approach of cancer transcriptomes reveals prevailing transcriptional networks in cancer cells. *Genome informatics. International Conference on Genome Informatics*, 22, pp.121–31.

Oltvai, Z.N. & Barabási, A., 2002. Life's Complexity Pyramid. *Science*, 298, pp.763–764.

P. Erdös, A.R., 1959. On random graphs I. *Publicationes Mathematicae (Debrecen)*, 6, pp.290–297.

Palla, G. et al., 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043), pp.814–8.

Penfold, C. a. & Wild, D.L., 2011. How to infer gene networks from expression profiles, revisited. *Interface Focus*, 1(6), pp.857–870.

Petranovic, D. et al., 2010. Prospects of yeast systems biology for human health: integrating lipid, protein and energy metabolism. *FEMS Yeast Research*, 10, pp.1046–1059.

R Development Core Team (2011), 2011. R: A language and environment for statistical computing. *R Foundation for Statistical Computing, Vienna, Austria.*, ISBN 3-900.

Ravasz, E. et al., 2002. Hierarchical organization of modularity in metabolic networks. *Science (New York, N.Y.)*, 297(5586), pp.1551–5.

Ravasz, E. & Barabási, A.-L., 2003. Hierarchical organization in complex networks. *Physical Review E*, 67(2), pp.1–7.

Richards, A.L. et al., 2008. A comparison of four clustering methods for brain expression microarray data. *BMC bioinformatics*, 9, p.490.

Rintala, E. et al., 2011. Transcriptional Responses of Saccharomyces cerevisiae to Shift from Respiratory and Respirofermentative to Fully Fermentative Metabolism. *Omics : a journal of integrative biology*, 15(7-8), pp.461–76.

Rodwell, G.E.J. et al., 2004. A transcriptional profile of aging in the human kidney. *PLoS Biology*, 2(12).

Sales, G. & Romualdi, C., 2011. parmigene--a parallel R package for mutual information estimation and gene network reconstruction. *Bioinformatics (Oxford, England)*, 27(13), pp.1876–7.

Schaffter, T., Marbach, D. & Floreano, D., 2011. GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics (Oxford, England)*, 27(16), pp.2263–70.

Science, B.A., Provost, F. & Fawcett, T., 2011. *Biosensors for Health, Environment and Biosecurity* P. A. Serra, ed., InTech.

Scutari, M., 2009. Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3).

Serra, P.A. ed., 2011. *Biosensors for Health, Environment and Biosecurity*, InTech.

Shi, Z., Derow, C.K. & Zhang, B., 2010. Co-expression module analysis reveals biological processes, genomic gain, and regulatory mechanisms associated with breast cancer progression. *BMC systems biology*, 4, p.74.

Simon, I. et al., 2001. Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell*, 106, pp.697–708.

De Smet, R. & Marchal, K., 2010. Advantages and limitations of current network inference methods. *Nature reviews. Microbiology*, 8(10), pp.717–29.

Smyth, G.K., 2005. limma: Linear Models for Microarray Data. In *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. New York: Springer-Verlag, pp. 397–420.

Smyth, G.K., 2004. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical applications in genetics and molecular biology*, 3(1), p.Article3.

Steele, E. et al., 2009. Literature-based priors for gene regulatory networks. *Bioinformatics (Oxford, England)*, 25(14), pp.1768–74.

Steele, E. & Tucker, A., 2008. Consensus and Meta-analysis regulatory networks for combining multiple microarray gene expression datasets. *Journal of biomedical informatics*, 41(6), pp.914–26.

Steuer, R. et al., 2002. The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics (Oxford, England)*, 18 Suppl 2, pp.S231–40.

Stouffer S.A, S.E.. et al, 1949. The American Soldier: Adjustment During Army Life. Volume I. *JAMA: The Journal of the American Medical Association*, 140(14), p.1189.

Stuart, J.M. et al., 2003. A gene-coexpression network for global discovery of conserved genetic modules. *Science (New York, N.Y.)*, 302(5643), pp.249–55.

Tamayo, P. et al., 1999. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences of the United States of America*, 96(6), pp.2907–12.

Tang, B. et al., 2012. Hierarchical Modularity in ERα Transcriptional Network Is Associated with Distinct Functions and Implicates Clinical Outcomes. *Scientific reports*, 2, p.875.

Tavazoie, S. et al., 1999. Systematic determination of genetic network architecture. *Nature Genetics*, 22(3), pp.281–285.

Tseng, G.C., Ghosh, D. & Feingold, E., 2012. Comprehensive literature review and statistical considerations for microarray meta-analysis. *Nucleic acids research*, 40(9), pp.3785–99.

Venables, B., 2000. correlation matricies: getting p-values? Available at: https://stat.ethz.ch/pipermail/r-help/2000-January/009758.html.

Wang, Y. et al., 2006. Inferring gene regulatory networks from multiple microarray datasets. *Bioinformatics (Oxford, England)*, 22(19), pp.2413–20.

Wang, Y.K. et al., 2013. Integration of steady-state and temporal gene expression data for the inference of gene regulatory networks. *PloS one*, 8(8), p.e72103.

Weckwerth, W., 2003. Metabolomics in systems biology. *Annual review of plant biology*, 54, pp.669–89.

Werhli, A. V, Grzegorczyk, M. & Husmeier, D., 2006. Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks. *Bioinformatics (Oxford, England)*, 22(20), pp.2523–31.

Yu, G. et al., 2012. clusterProfiler: an R package for comparing biological themes among gene clusters. *Omics : a journal of integrative biology*, 16(5), pp.284–7.

Yu, H. & Gerstein, M., 2006. Genomic analysis of the hierarchical structure of regulatory networks. *Proceedings of the National Academy of Sciences of the United States of America*, 103(40), pp.14724–31.

Yu, J. et al., 2004. Advances to Bayesian network inference for generating causal networks from observational biological data. *Bioinformatics (Oxford, England)*, 20(18), pp.3594–603.

Zakin, D. V., 2011. Optimally weighted Z-test is a powerful method for combining probabilities in meta-analysis. *Journal of Evolutionary Biology*, 24(8), pp.1836–1841.

Zou, K.H., Tuncali, K. & Silverman, S.G., 2003. Correlation and simple linear regression. *Radiology*, 227(3), pp.617–622.

Zou, M. & Conzen, S.D., 2005. A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics (Oxford, England)*, 21(1), pp.71–9.