Reverse engineering of biological signaling networks via integration of data and knowledge using probabilistic graphical models

Dissertation
In partial fulfillment for the degree of
Doctorate in Bioinformatics (Dr. rer. nat.)
in Mathematics and Natural Science Faculty
Rheinische Friedrich-Wilhelms- Universität Bonn
Germany

Submitted by
Paurush Praveen
Dighwara (India)

May 2014

Gutachter (Principal Advisor): Prof. Dr. Holger Fröhlich
Gutachter (Second-Advisor): Prof. Dr. Martin Hofmann-Apitius

Angefertigt mit Genehmigung
der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

Laboratory:
Algorithmic Bioinformatics
Bonn-Aachen International Center for Information Technology
Dahlmann str 2, 53113 Bonn Bonn, Germany

Thoughts without content are empty, intuitions without concepts are blind.
The understanding can intuit nothing, the senses can think nothing.
Only through their unison can knowledge arise.

- Immanuel Kant

Declaration

I herewith affirm that this dissertation is an original work, except where indicated through the proper use of citations and references. Any uses made within it of the works of other authors in any form are properly acknowledged at the point of their use. A full list of the references employed has been included.

Paurush Praveen

Date:

Place: Bonn

Acknowledgements

A successful work is the acme of abounding efforts by many people, some directly involved and some others quietly encouraging and supporting behind the scene. I would like to hereby express my sincere gratitude to everyone who has taken a role-in to make this effort an efficacious one.

First and foremost I would like to express my thanks to my principal adviser, Prof. Dr. Holger Fröhlich for his astute guidance, amaranthine patience and consistent support. With his experience, suggestions and motivations, he kept me in pursuit throughout the doctoral work. Without his foster I would not have reached the project goals. I thank him for providing me an opportunity to work under him on the topic.

I sincerely thank Prof. Dr. Martin Hofmann-Apitius; my co-adviser, for reviewing my work and advising me with his valuable comments and suggestions. His vision for my scientific pursuance, encouragement and impeccable believe in every dimension was instrumental and morale boosting.

I feel beholden to Prof. Dr. Andreas Weber and Prof. Dr. Ulrich Jaehde who agreed to review my thesis and be a part of my doctoral committee.

I would like to express my words of gratitude to Prof. Dr. Armin Cremers and Prof. Dr. Mathias Jarke, for granting me the B-IT Research School scholarship to pursue my research. I acknowledge the 'State of Nordrhein-Westfalen (Germany)' for the research funding provided via the graduate school

I would like to thank my group members of ABI@B-IT specially; Khalid for providing his enormously important suggestions, Yupeng, Gloria, Satya, Nikhil, Afshin and Gurnoor for useful discussions and support. The computer infrastructure support I received from the Mr. Thomas Theil and his group members was enormous. Finally, I would like to thank my parents, family and all friends for their enthusiasm, and unflagging support.

Abstract

Motivation

The postulate that biological molecules rather act together in intricate networks, pioneered systems biology and popularized the study on approaches to reconstruct and understand these networks. These networks give an insight of the underlying biological process and diseases involving aberration in these pathways like, cancer and neuro degenerative diseases. These networks can be reconstructed by two different approaches namely, data driven and knowledge driven methods. This leaves a critical question of relying on either of them. Relying completely on data driven approaches brings in the issue of overfitting, whereas, an entirely knowledge driven approach leaves us without acquisition of any new information/knowledge. This thesis presents hybrid approach in terms of integration of high throughput data and biological knowledge to reverse-engineer the structure of biological networks in a probabilistic way and showcases the improvement brought about as a result.

Accomplishments

The current work aims to learn networks from perturbation data. It extends the existing Nested Effects Model (NEMs) for pathway reconstruction in order to use the time course data, allowing the differentiation between direct and indirect effects and resolve feedback loops. The thesis also introduces an approach to learn the signaling network from phenotype data in form of images/movie, widening the scope of NEMs, which was so far limited to gene expression data. Furthermore, the thesis introduces methodologies to integrate knoowledge from different existing sources as probabilistic prior that improved the reconstruction accuracy of the network and could make it biologically more rational. These methods were finally integrated and for reverse engineering of more accurate and realistic networks.

Conclusion

The thesis added three dimensions to existing scope of network reverse engineering specially Nested Effects Models in terms of use of time course data, phenotype data and finally the incorporation of prior biological knowledge from multiple sources. The approaches developed demonstrate their application to understand signaling in stem cells and cell division and breast cancer. Furthermore the integrative approach shows the reconstruction of AMPK/EGFR pathway that is used to identify potential drug targets in lung cancer which were also validated experimentally, meeting one of the desired goals in systems biology.

Contents

Chapter 1
Introduction

1.1 Biological Systems

Biological systems are remarkably complex. They owe their functionality and behavior not only to their components (genes, proteins and other small molecules) but to the synergistic output of complex interactions among them (Kitano, 2002). Therefore, deciphering a biological system like cell needs not only the elucidation of the individual entities but also of the interplay involved therein. To achieve this, two possible ways (Figure 1.1) have been proposed ; top-down and bottom-up (Bruggeman and Westerhoff, 2007; Kitano, 2002). In the current work the focus is on the reverse engineering of systems which is primarily considered as top-down approach. The top-down approach treats the entire system as a black box and then tries to identify the details of the components of the systems following a reductionist approach. It uses data as the starting point and statistical data mining approaches are applied for a comprehensive understanding of the biological system. The second way is the bottom-up approach, starting from the lowest level of system organization. First, the details of the individual units of the system are collected and then the entire system is constructed using these system components.

1.2 Networks in Biological systems

As it is about the interactions of the constituents rather than merely the constituent itself, at a comprehensive level cellular systems can be conceivably represented as graphs or networks. These networks abstractly represent a biological system, capturing their core characteristics. In these circuits nodes represent the molecular entities and edges connecting pairs of vertices correspond to the relation between them. The overall cellular system is an overlay of such networks. Cellular systems comprise many diverse components and component interactions. There are signal transduction, transcriptional and metabolic networks. These different network types are not distinct from each other, but they are interconnected and operate together. Every category has its own design principle and biological behavior (Palsson, 2006; Alon, 2006). Therefore it is a prerequisite to go through these categories and their properties before making an attempt to reconstruct them and understand their functioning.

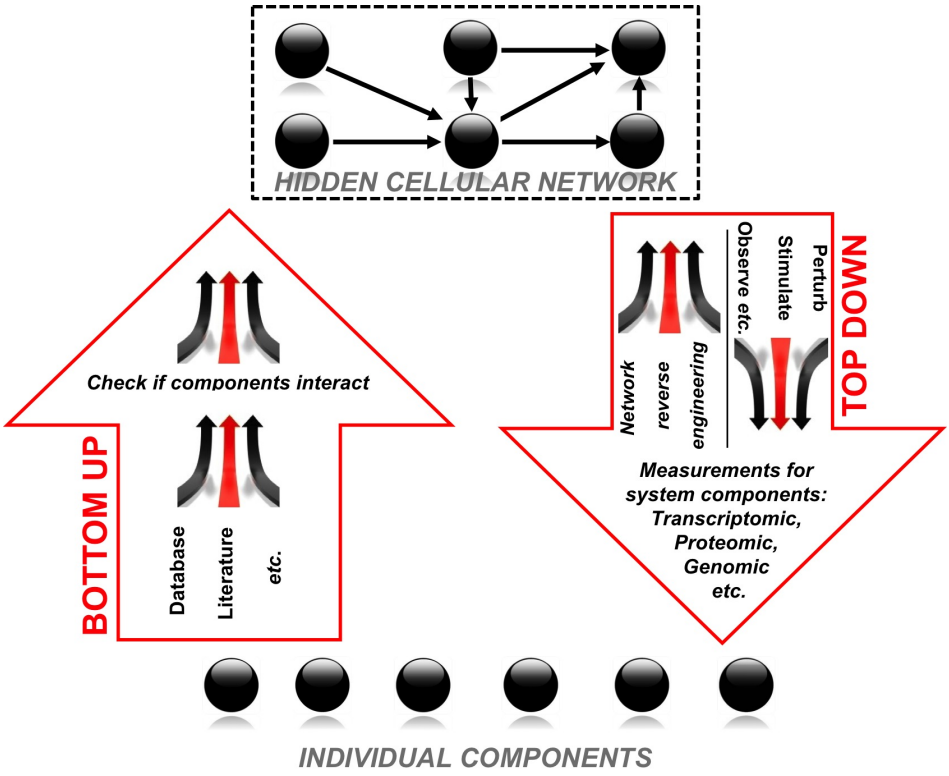

Figure 1.1.: Modeling approaches in systems biology. Bottom-up systems biology is knowledge driven; by contrast, top-down systems biology is systemic-data driven.

1.2.1. Categories of biological networks

Signaling Networks: In order to respond to changes in their immediate environmental stimuli, cells must be able to receive and process signals. Signal transduction pathways govern a cell's response to extra-cellular stimuli, including, e.g., how a cell adapts its transcriptional regulatory program in response to specific environmental changes. Signaling pathways (Figure 1.2) refer to the set of biochemical processes using which cells respond to internal or external cues (Albert and Oltvai, 2007). The entire life history of cell starting from its proliferation, differentiation, functioning unto its death is orchestrated by signaling pathways. Signaling pathways in contrast to metabolic pathways convey information. They process and encode along with receiving and transmitting information. Thus signaling involves a set of cascades taking place between the receptor and effector.

Cell signals are mostly chemical in nature. Receptors (most often trans-membrane proteins) bind to signaling molecules and subsequently transmit the signal via protein actions like ion channel opening. Signal transduction systems can be simple, like- transfer of ions resulting in the electrical potential difference of the cell, propagating the signal in the cell or more complex signal transduction (Gutkind, 2000). Activation of receptors can trigger the synthesis of small molecules called second messengers, which initiate and coordinate intra-cellular signaling pathways via enzyme activation via phosphorylation It allows for intricate control of protein function. At any one time, a cell is receiving and responding to numerous signals, and multiple

signal transduction pathways are operating in its cytoplasm. Many points of cross talks exist among these pathways. For instance, a single second messenger or protein kinase might play a role in more than one pathway. Through this network of signaling pathways, the cell is constantly integrating all the information it receives from its external environment.

Metabolic Networks: Metabolism in cellular systems produce energy, amino acids, and other precursors required for the growth and maintenance of a cell. The set of biological networks associated with such activities fall into the category of metabolic networks. Metabolic networks envisage the various chemical reaction involved in anabolism (construct molecules from smaller units) and catabolism (breaking bigger molecules into smaller units and release energy) [1]. These networks bear a lot of small molecules and intermediate small molecules (Palsson, 2006). Such networks execute chemical changes of the involved molecules. As an example consider glycolysis, where glucose ($C_6H_{12}O_6$) is converted into pyruvate ($CH_3COCOO^- + H^+$). The process releases energy used to form the high-energy compounds ATP (Adenosine Tri Phosphate) and NADH (reduced Nicotinamide Adenine Di-Nucleotide) (Nelson et al., 2008).

Metabolic network mostly have been built manually through a four step process including an initial reconstruction from gene-annotation coupled with information from online databases e.g. KEGG (Kanehisa et al., 2014).

Transcriptional Networks: Transcriptional regulatory networks control the transcription state of a genome. In general, they describe the connections between environmental cues and transcriptional responses. They are comprised of nodes, the genes and their regulators, joined together by edges, which represent physical and/or regulatory interactions. Physical interactions between genes and TFs (Transcriptional Factors) can be delineated using two conceptually and practically different strategies that are highly complementary (MacNeil and Walhout, 2011). In addition to representing physical interactions, GRN edges can also represent regulatory relationships that can, for instance, be inferred by correlating gene expression profiles between genes and potential regulators.

Transcriptional networks are involved in the regulation the expression of thousands of genes involved in different biological processes. These networks act as a control systems for various biological processes like cellular development. Acting as a hardwired control system, they invoke responses through sequential steps in the form of genomic regulatory codes. The role of these systems is to specify and regulate the sets of genes that must be expressed in specific spatial and temporal patterns. In physical terms, these control systems consist of many thousands of modular DNA sequences. Each such module receives and integrates multiple inputs, in the form of regulatory proteins (activators and repressors) that recognize specific sequences within them (Davidson and Levin, 2005). The end result is the precise transcriptional control of the associated genes. Some regulatory modules control the activities of the genes encoding regulatory proteins.

It is worth mentioning that though these are different types of networks they do not always exist in isolation but may be involved in a cross talk . For example, the pluripotent state in

[1] http://www.chem.qmul.ac.uk/iupac/bioinorg; Accessed: April 2013

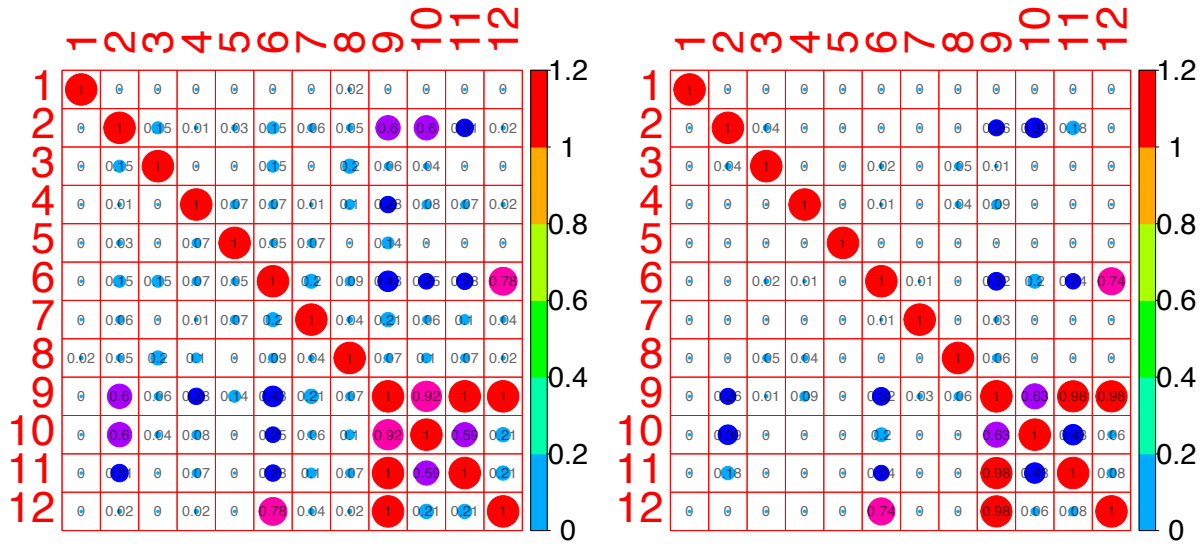

Figure 1.2.: Figure showing sharing of nodes and edges across different networks. (left) Proportion of nodes and (right) edges shared across 12 KEGG pathways. Indicating the properties of biological network for cross-talk with shared nodes (left) and modularity with shared edges (right). A detailed list of the involved pathways and the formula for the computation of this plot is available in appendix A

embryonic stem cells is maintained by a network of transcription factors and is influenced by specific signaling pathways (Ng and Surani, 2011) or glycolytic enzymes acting as transcriptional regulator (whan Kim and Dang, 2005). The former involves a cross talk of transcriptional and signaling network, whereas the later illustrates the case of transcriptional and metabolic networks. This increases the complexity in classification of biological networks.

1.2.2. Properties of biological networks

Robustness: Robustness is a ubiquitous property of biological systems (Kitano, 2004). Robustness is the property of a system to by virtue of which, it maintains its functionality caused by internal or external perturbations (Rizk et al., 2009). In both engineering and biology, a system must function under all likely interventions that come with the inherent properties of the components and the environment (Alon, 2003) and sustain them. The networks operating in the biological system must keep the concentrations of their components within tightly defined bounds despite intra and extra cellular disturbances. The topology of these networks renders the output of the pathway invariant against a large class of possible adverse fluctuations like-changes in energy states or total protein concentrations (Steuer et al., 2011).

Dynamics: The molecular interactions within signaling networks operate in the order of seconds to minutes at timescale while for trancriptional networks, it can be longer (Alon, 2006, page:11). The components of a signaling pathway exhibit a dynamically coordinated behavior in time and space for specificity in their response Kholodenko (2006). They respond to changes in their environment and cell state, and they execute these responses on timescales that can

be observed via genomic technologies. The specificity in signal response is contributed by the timing, amplitude and duration of signaling.

Cross-talk: Though life processes are diverse, yet the number of biological pathways especially signaling pathways is limited. The cross talks across different types of networks has already been presented in the previous section. This part deals in a more generalized way about cross talks. There is a sharing of some signaling pathways or a part of it across biological processes (Figure 1.2). To illustrate, we consider MAPK cascades, which can be activated/inactivated by many stimuli/perturbations (Figure 1.3)(Raman et al., 2007). They can regulate diverse cellular phenomena ranging from proliferation, differentiation to apoptosis. Hence, common components of a signaling pathway come-up with elicit effects in the same signaling pathway. A characteristic feature of cellular signaling in eukaryotic cells is that components are frequently shared among pathways, providing a potential for cross-talk. However, this can also lead to an inappropriate response, if stimulus specific signals transmitted through one pathway inadvertently cross-activate the other(s). Several known mechanisms can enable signaling pathways with shared components to respond specifically to any one stimulus. Spatial insulation can be achieved by localizing the pathways to different cellular compartments or by incorporating the shared component into distinct macromolecular complexes through scaffolding molecules.

Specificity: Though, the biological space is shared by many signaling pathways and cells respond to a wide variety of stimuli, yet they maintain specificity (Kholodenko, 2006). Specificity enables transmission of different signals by common components of signaling pathway, in response to corresponding stimuli; eliciting distinct outputs (Komarova et al., 2005). A classical example for specificity is the MAPK pathway (*mitogen -activated protein kinase*) (Figure 1.3). EGF induces transient MAPK activation, which results in cell proliferation, whereas a sustained MAPK activation by NGF changes the cell fate and induces cell differentiation (Marshall, 1995; Murphy and Blenis, 2006). Thus, cells respond to a myriad of stimuli using a limited number of signaling pathways. These pathways do not simply transmit, but process, encode and integrate signals. Specificity can be attributed to spatio-temporal activation profiles of the same repertoire of signaling proteins (Hoffmann et al., 2002). This property of specificity paves the way to understand the system behavior on specific stimuli/perturbation.

Modularity: A module in a network is a set of nodes that have strong interactions within themselves and a common function. A module has defined input nodes and output nodes that control the interactions with the rest of the network of interconnected nodes, each of which has a state that depends on the integrated inputs from other nodes. A module also has internal nodes that do not significantly interact with nodes outside the module (Alon, 2003). The potential reason for network modularity is that individual modules serving a defined biological function developed as one block during the evolutionary process. (Alon, 2003).

Such circumstances make biological networks an interesting as well as challenging domain to study. These concepts, together with the current technological revolution in biology, may eventually allow characterization and understanding of cell-wide networks, with great benefit to medicine as well as develop an understanding the laws of nature operating and evolving biological systems (Alon, 2003).

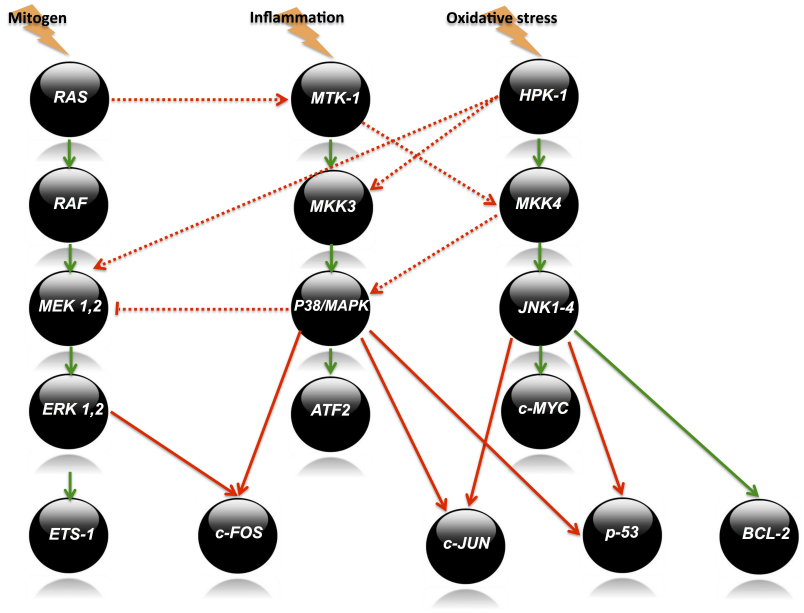

Figure 1.3.: MAPK signaling pathways organized in modular cascades in which activation of upstream kinases by cell surface receptors lead to sequential activation of a MAPK module. The figure shows the major MAPK pathway components and their targets for different stimulations. In green we have the normal signal flow pathways and red lines indicate combined action (AND or OR). Dotted lines indicate signaling cross-talks between MAPK modules. The figure thus represents the modular, specific and cross talk nature in MPK pathways as an example. Redrawn from www.cellsignaling.com [Accessed: March 2013]

1.3 Reverse Engineering

The reconstruction of mapping the intelligible structure of network among molecular components from data is termed as data driven network inference (Stolovitsky and Califano, 2007). Reverse engineering (or deconvolution) is the process of elucidating the structure of the system by reasoning backward from observation of its behavior (Hartemink, 2005). Network inference via reverse engineering is one of the challenges in Bioinformatics. In a way reverse engineering analyzes the behavior of a system to characterize its architecture. It can monitor these profiles and can traceback the pathways, helping one to experimentally validate the predictions (Stolovitzky et al., 2007). Depending on the data used for inferring the network, which, principally, may either come from DNA microarray, RNA-seq, proteomics or ChIP-chip experiments, or combinations thereof, the biological interpretation of an edge in these networks is dependent thereon. For expression data, inferred interactions may preferably indicate transcription regulation, but can also correspond to protein-protein interactions (Emmert-Streib et al., 2012; Tegnér et al., 2003).

The ambition to achieve systematic, comprehensive and accurate reverse engineering of biological networks makes systems biology much more demanding for experimental biologists than the current practice of biology. A methodical experiment has to be performed and at the same time the quality of measurement and data produced should be high to be used as a reference point for simulation, modeling and prediction (Kitano, 2002; Hache et al., 2009).

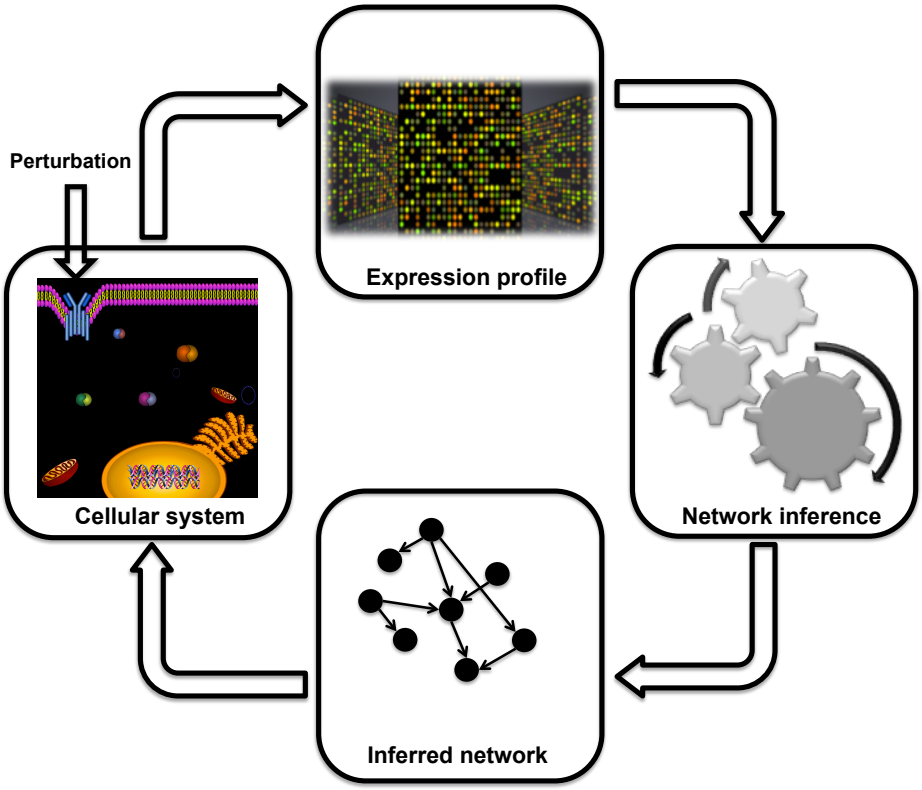

Figure 1.4.: Schematic diagram for the work-flow in reverse engineering of cellular networks via perturbation technique.

1.3.1. Experimental techniques

A modern way to reverse engineering cellular networks is to perturb the cellular system and observe its response. A specific perturbation perturbations forces the cell to find new equilibrium points and hence can activates specific pathways ultimately leading to corresponding expression profiles. However non-perturbation techniques are most often encountered due to their simplicity. Canvassed below are the various experimental techniques and sources of data used for the purpose.

Non-perturbation techniques

The simplest experiment to generate data for network reverse engineering is measuring the components of biological systems (genes/proteins) in different conditions. The conditions can be time (Morrissey et al., 2010; Wang et al., 2006) or environment (Dhaeseleer et al., 2000;

Carrera et al., 2009). These methods do not need any targeted perturbation of involved components. For example- an expression data for a cell under different conditions of environmental stress outline the pattern that define the connections between the genes/proteins (Carrera et al., 2009). (Penfold and Wild, 2011)

Perturbation based techniques

Genetic interactions are best inferred when the genes explore a substantial dynamical range. Traditionally, this has been achieved by systematic perturbations in simple organisms (e.g., by large-scale gene knockouts or exogenous constraints) (Esvelt and Wang, 2013), which are not easily obtained in more complex cellular systems (Basso et al., 2005). Perturbation data can be used to deduce the biological networks via reverse engineering work-flow. The first step in such reverse engineering of cellular system is the perturbation of a functioning system (di Bernardo et al., 2005). However, as single perturbation cannot lead to conclusions in biological systems and reveal the underlying network. For this we need to perturb different genes (entities) to explicitly understand the dynamics of involved components (Stelniec-Klotz et al., 2012).

In organisms such as yeast such perturbations are easier, but in higher organisms/cells (eukaryotic), it gets experimentally complicated. RNAi is the offers a practical way to perturb multiple genes in these systems (Wang et al., 2011). Systematic RNA interference (RNAi) perturbations allows performing such perturbation for specific genes in a cell (Tewari et al., 2004). During the experiment a double-stranded RNA (with a sequence complementary to a gene of interest) is introduced into a cell / organism, where it is recognized as exogenous genetic material and activates the RNAi pathway (See BOX-1.1 for details) leading to a drastic decrease in the expression of a targeted gene. This activity can be then measured in different ways (see section 1.3.2).

RNA interference (RNAi) allows simultaneous screening of hundreds to thousands of genes in a high content manner. This screening can be achieved by first performing a targeted RNA interference followed by measuring the expression values for the entire cell. Network inference often has the goal of generating testable hypotheses regarding biological interplay (Peér and Hacohen, 2011). RNAi makes this possible by mapping the RNAi functional network to that of the protein interaction networks (Ramanuj Dasgupta, 2004). It can help to identify new regulators overlooked in RNAi screens thus generating testable hypothesis on gene function.

1.3.2. Data from experiments

The effect of above mentioned experimental techniques measure the system in terms of activities of biomolecules (genes/proteins) for reverse engineering of the system. However, it is important to measure the changes brought about by these perturbations simultaneously for a large number of genes to get overall impact of perturbation on the cellular system. Gene expression measurements offers the most obvious solution to the issue as per the “Central Dogma of Molecular Biology” (Figure 1.5). They represent the level of transcriptional activity of each gene under observation. High throughput technologies like microarrays (Brown and Botstein, 1999), allow the researchers to monitor average mRNA concentrations in a cell population on a genome-wide scale. Microarrays offer opportunities to identify gene deletion consequences on entire genomes.

Although, the activity measurement of involved molecules is mainly accomplished with microarray techniques, other techniques like ChiP, SAGE and RNA-seq are also popular to measure gene activity at mRNA level. The differential activity of cell can also be measured in terms of protein expression. At protein level, techniques like RPPA (Pierobon et al., 2011), mass spectrometry and western blotting is used. Although other high throughput techniques are available, the current work is mainly based on microarray data.

Figure 1.5.: The Central Dogma of Molecular Biology: The information flow from DNA information to proteins.

Microarray Data

Microarrays (Schena et al., 1995) are widely used for the expression profiling of thousands of genes simultaneously. It involves attaching probes (gene sequences) by robotic machines on a pre-determined spots onto a chip. The mRNA produced as a result of activity of corresponding gene in the cell of interest, binds to these probes which are complementary to the mRNA‘s. To make the measurements of the mRNA‘s, they are labeled with a fluorescent dye. Certain active genes produce more mRNA that attach to the chip and produce brighter areas whereas spots that are not bright evidence lower activity of corresponding genes. The intensity of fluorescence thus reflects the quantified presence of the mRNA‘s and ultimately the gene expression.

1.4 Objectives

The objectives of the current work are as follows:

1. Extend Nested Effects Model (NEM) approach to time course data.
2. Develop a method to adapt NEMs to be able to use phenotype data.
3. Develop a method to integrate different sources of biological knowledge as probabilistic consensus prior for network inference.
4. Apply and test the effect of informative prior in network inference via NEM.

1.5 Document road-map

The present document has been sectioned into 8 chapters. This chapter discussed the biological aspects of introduction to the biological networks and reverse engineering along with challenges and goals of this thesis. The upcoming chapters outline as follows:

- **Modeling Approaches:** Chapter 2 gives an overview of recent approaches to map the dependency structure between genes. The chapter gives an overview of reverse engineering of biological systems via probabilistic graphical models and their state of art.
- **Nested Effects Model:** Chapter 3 describes Nested Effects Models to infer network from perturbation data. The chapter describes the principle and mathematical formalism of nested effects models and its different formulations.
- **dynoNEM: Dynamic Nested Effects Model:** The 4^{th} chapter discusses a novel approach called dynoNEM (Dynamic Nested Effects Models) to apply Nested Effects Models on time-course perturbation data. The method was developed during current doctoral work. The further improvement in the approach with MCMC based sampling is also discussed in this chapter.
- **Movie-NEM: dynoNEM with image data** Chapter 5 proposes the use of image features from movie as a source of time series data to reverse engineer biological network. The novel MovieNEM approach is introduced in this chapter.
- **Developing informative prior:** The 6^{th} chapter talks about the integrating information from different heterogeneous sources of biological knowledge into one probabilistic consensus prior. Such a prior can be used to improve network inference.
- **Applying prior knowledge** The seventh (7) chapter displays the application of the probabilistic consensus prior and NEM on Non-Small Lung Cancer perturbation data.
- **Conclusion and Outlook:** The 8^{th} and final chapter brings out the high level conclusive messages and an exploration of future outlook in the direction. The chapter outlines the accomplishments of the work and the possible future orientation of the research done during the doctoral work.

The current chapter overviewed the biological concepts regarding biological network and technical details from a biologist point of view. The properties and of signaling pathways and their importance were discussed. This was followed by a brief description of experiments and data used to understand the structure and functioning of the signaling system in cell. Finally the chapter came up with the plans and goals for the current work. In the next chapter; an overview of methods to model these signaling systems is presented with a detailed description of Nested Effects Model (NEM) as it is the in-focus tool for this dissertation.

BOX 1.1: RNA Interference (RNAi)

RNA interference (RNAi) is a powerful tool to perform loss-of-function genetic screens (Fire et al., 1998; Hannon, 2002). Certain small dsRNAs, such as short interfering RNAs (siRNAs) and microRNAs (miRNAs) found in mammalian cells, regulate gene expression via gene-silencing enzymatic complexes. RNAi offers certain advantages over insertional mutation like- speed, flexibility and convenience.(Boehm and Hahn, 2011)

The work-flow for RNAi (Figure 1.6 drawn using pathway builder[a])starts with the introduction of a double stranded RNA (dsRNA) into the cell. These are recognized and processed into small interfering RNAs (siRNAs) by Dicer. Dicer is a double-stranded-RNA-specific ribonuclease from the RNase III protein family. The double stranded siRNAs are passed to the RNA-induced silencing complex (RISC), and the complex becomes activated by unwinding of the duplex. Activated RISC complexes can regulate gene expression at many levels. Almost certainly, such complexes act by promoting RNA degradation with siRNAs and translational inhibition with microRNAs. However, similar complexes probably also target chromatin remodeling. In plants, amplification of the silencing signal can also occur. The ways in which miRNAs cause silencing of their target mRNAs are still debated. The mechanisms involved are likely to include: inhibition of translation; triggering removal of the poly(A) tail from mRNAs (de-adenylation); disruption of cap-tail interactions; and degradation of mRNAs by exonucleases and further translational inhibition by micro RNAs (Hannon, 2002). The difference in the cell can then be accessed based on microarray data.

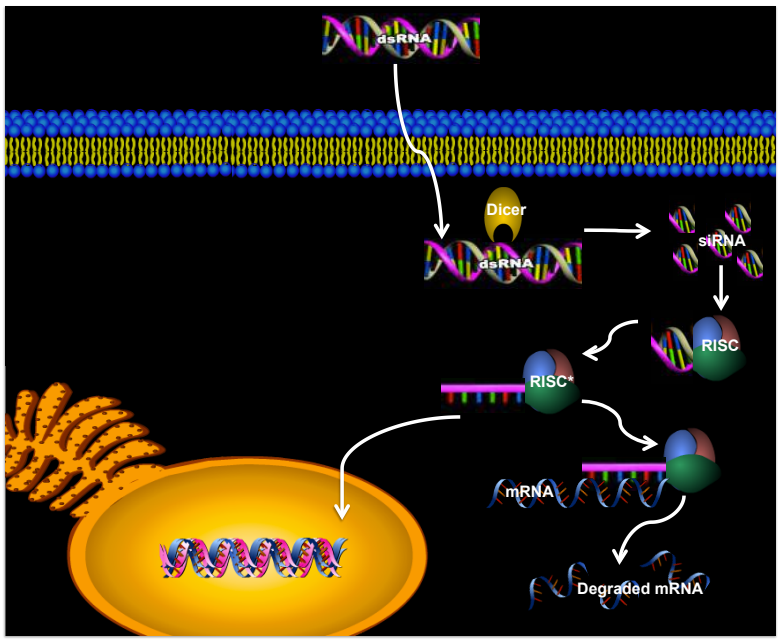

Figure 1.6.: Diagram representing overall scheme of RNAi process. Redrawn from Hannon et al. using pathway builder

[a]www.proteinlounge.com ; Accessed February 2013

Chapter 2
Reverse Engineering of Biological Networks

This chapter will deal with different paradigms of network reverse engineering popular in bioinformatics and computational biology. The concept involved, state of art and conclusive messages for these approaches will be explained in brief here. Although the thesis focuses on Nested Effects Models, the chapter covers other approaches for the sake of completeness. For every approach the formalism, state of art and critical points are entailed in the current chapter. Bayesian Networks has been dealt rigorously in the present chapter as the basics for Nested Effects Models (NEMs).

2.1 Introduction

The idea that biological molecules rather act together in intricate networks, pioneered systems biology. This postulate has popularized the study on approaches to reconstruct and understand networks. In the first chapter an introduction to biological systems together with different experimental techniques that can be used to measure their activity were discussed. Reconstructing networks from such data can offer insights in the functioning of biological systems. However, it is an under-determined problem, as the number of interactions that can be inferred exceeds the number of independent measurements (De Smet and Marchal, 2010). At the same time, the properties of networks like dynamical behavior, modularity, cross-talk etc. (See 1.2.2) make the task challenging. Different state-of-the-art tools for network inference use specific assumptions and simplifications to deal with under determination, and this influences the inference. The outcome of network inference therefore varies between tools and can be highly complementary. In the current chapter the methods and tools available for network reconstruction are discussed. For the ease of discussion the methods are categorized into different classes based on the mathematical formalism adopted in there.

2.2 Approaches

The current state-of-the-art approaches for network inference rely on specific assumptions. Each of these methods therefore, differ in terms of strategy, mathematical schema and ultimately the inferred network (De Smet and Marchal, 2010). This section presents these categories and their underlying framework with illustration.

2.2.1. Clustering and correlation based approaches

Though clustering is not precisely a network inference algorithm, it lends a framework to analyze gene expression and group genes exhibiting similar expression profile into clusters (Eisen et al., 1998). Each cluster contains a set of genes whose activities follow similar trend (up-regulation and down-regulation). Such genes are said to be co-expressed. The rational behind the use of clustering as a network inference tool is that there is a high chance of the genes in a cluster to be functionally related or co-regulated (Dhaeseleer et al., 2000). This gives rise to the concept of co-expression networks. Co-expression networks are constructed by computing first a correlation matrix for genes and then a similarity score for each pair of genes based on their expression pattern distances. This similarity can serve as the weight for the corresponding edge.

Gene co-expression network have led to many interesting findings like discovery of conserved genetic modules (Dhaeseleer et al., 2000; Stuart et al., 2003; Oldham et al., 2006), study T-helper cell differentiation (Elo et al., 2007) and chronic fatigue syndrome (Presson et al., 2008). Commendable efforts have also been made to study multiple microarray data sets with such approaches (Lee et al., 2004). COXPRESdb; a databases of gene co-expression networks for some model mammals have been constructed from large numbers of microarray data sets (Obayashi et al., 2013). These models were further extended with weighted gene co-expression networks (Horvath and Dong, 2008). The connected genes in co-expression networks often show their relatedness in terms of enrichment for Gene Ontology categories (Horvath and Dong, 2008; Stuart et al., 2003), indicating their functional relatedness in biological space. Ultimately, this leads to the conclusion that such networks are biologically meaningful to a certain extent.

Furthermore, a conditional model belonging to the class of correlation based models is the 'Gaussian Graphical Model' (GGM) (Dempster, 1972). This is based on the assumption of multivariate normal distribution of data. GGMs consider an edge between two vertices given the rest of the observation, thus imparting the attribute of conditional models and hence the partial correlation is used to define the edge between two vertices (Toh and Horimoto, 2002).

GGMs have also been used to reconstruct regulatory network from time series data (Liu et al., 2012). In addition GGMs being based on partial correlation provide a stronger measure for dependence Markowetz and Spang (2007). Correlation based networks indicate that two genes are co-regulated, participate in common pathway, share a biological process, function or location or even directly bind to one another (Hartemink, 2005). Such networks do show the functional relatedness of bio-molecules, however they do not explicitly narrate the nature and directionality of the function. This undermines the power of network based biological studies.

What co-expression network usually result into is an undirected graph (Ruan et al., 2010), therefore not indicating the causation and direction of regulation. However, an edge inferred by a GGM could be causal, but it is not guaranteed to be. Furthermore, co-expression networks might not distinguish direct gene interactions from indirect ones with an exception to GGMs. Simple correlation based networks do not confirm a direct interaction among the co-expressed genes, as genes separated by one or more intermediaries (indirect interaction)

can still follow co-expression (Bansal et al., 2007). Gaussian graphical models (GGMs) circumvent indirect association effects by evaluating conditional dependencies in multivariate Gaussian distributions. Another interesting aspect of such inference methods is that they include neighborhoods overlooked in cluster analysis (Horvath and Dong, 2008) leading to an intriguing geometric feature. Brunet *et al.* compared such networks with established networks in yeast and concluded that such network may resemble gene regulatory networks but less protein-protein interaction (PPI) or physical interaction networks (Xulvi-Brunet and Li, 2010).

The computational simplicity makes co-expression network based network inference a commonly used approach. It offers an effective method for predicting gene functions and the relationship between them at coarsely resolved scale. However, their limitations as discussed above attenuate this methods in terms of understanding biological systems.

2.2.2. Information theory based approaches

Information theoretic approaches offer an alternative to classical linear dependency measures based on Pearson's correlation. The advantage of the mutual information as the most prominent example of information theoretic dependency measures is that it is able to capture non-linear correlations between variables (MacKay, 2003). It considers the entropy; $H(X), H(Y)$ and joint entropy; $H(X,Y)$ of the involved pair of variables (i), here genes (Equation 2.1). MI or mutual information for a pair of genes is then defined as their combination

$$H_i = \sum_i (P(X_i) log(P(X_i)) \tag{2.1}$$

To further illustrate let us assume two genes G_1 and G_2. Their activity measurements can be supposed to be drawn from random variable say X and Y. For the moment assume the random variables are discrete. The random variable X for G_1 can take the values as equation 2.2 and a similar equation holds for G_2.

$$X = \begin{cases} 0; & with \;\; probability \;\; p \\ 1; & with \;\; probability \;\; 1-p \end{cases} \tag{2.2}$$

Using the equation 2.1 the entropy of $G_!$ ($H(G_1)$) can be given as follows:

$$H(G_1) = p \;\; log(p) + (1-p) \;\; log(1-p) \tag{2.3}$$

Therefore the entropy is a property of the probability distribution. The joint entropy for random variables X and Y can be given as

$$H(X,Y) = \sum_{X,Y} p(X,Y) \;\; log(p(X,Y)) \tag{2.4}$$

The mutual information is defined according to equation2.5 (Shannon and Weaver, 1963; Cover and Thomas, 1991).

$$MI(X,Y) = H(X) + H(Y) - H(X,Y)$$

$$= \sum_{X,Y} p(X,Y) \ log \ \frac{p(X,Y)}{p(X) \ p(Y)} \tag{2.5}$$

In case of continous random variables sums are replaced by integrals over the corresponding probability densities.

$$MI(X,Y) = \int_X \int_Y p(x,y) log \left(\frac{p(x,y)}{p(x)p(y)} \right) dxdy \tag{2.6}$$

Where, p(x,y) is now the joint probability density function of X and Y, and p(x) and p(y) are the marginal probability density functions of X and Y respectively.

Thus, if two variables are independent the MI turns to zero and hence shows no dependence and vice-versa. The mutual information between two variable is a more general measurement of dependence than correlation and measures non linear dependency, often seen in cellular systems (Brunel et al., 2010). It measures the reduction of uncertainty in Y after observing X. Thus, a pairwise MI across expression profile decides the presence or absence of edge in such networks or more precisely statistical independence or dependence of genes on each other (Bansal et al., 2007; Hartemink, 2005).

Interesting MI based approach include the CLR method by Faith *et al.* (Faith et al., 2007) and minet (available as R-package[1]) by Meyer *et al.* (Meyer et al., 2008). However, ARACNe [2] is the most famous member of this family network inference approach (Basso et al., 2005; Margolin et al., 2006). Although it is a MI based approach it can distinguish between direct and indirect relationship. It does not consider the MI alone but together with a principle called 'Data Process Inequality' (DPI). For all pairs of genes (i, j) the mutual information M_{ij} is computed via Kernel density estimation (Duda et al., 2001) (Gaussian kernel density (Steuer et al., 2002)). The DPI comes into picture to eliminate the weak edges. To illustrate, if gene i interacts with j and j with k then ARACNe prunes an edge $i \leftrightarrow k$ if the condition in equation 2.7 holds (Basso et al., 2005). Therefore, for each such triplets the edge corresponding to the lowest mutual information value is eliminated .

$$MI_{i,k} \leq min(MI_{i,j}, MI_{j,k}) \tag{2.7}$$

[1]http://bioconductor.org/packages/release/bioc/html/minet.html; Accessed: February 2013
[2]http://wiki.c2b2.columbia.edu/workbench/index.php/ARACNe; Accessed: February 2013

ARACNe has been used in human B-cell data and could detect dense hubs, it could also outline some validated transcriptional targets of the cMYC proto-oncogene. Later an extension of ARCANe: TimeDelay-ARACNE was proposed to infer network from time-course expression profiles (Zoppoli et al., 2010).

MI based methods specially ARCANe has certain advantages, e.g. it does not rely on $a-priori$ assumptions, does not need any heuristic search and does not require any discretization at expression level (Margolin et al., 2006). Nevertheless, there are certain drawbacks in the approach. The first issue with MI based approaches is owing to the symmetric nature of MI *i.e.* $MI_{i,j} = MI_{j,i}$. This makes the inferred network undirected. Furthermore, such networks cannot be guaranteed to yield causal interactions. However the TimeDelay-ARACNE attempts to overcome the issue to some extent by including the time dimension into the picture (Zoppoli et al., 2010). Another interesting aspect in ARCANe is the DPI. This condition is necessary but not sufficient, that is, the inequality can be satisfied even if (i, k) are directly interacting. Therefore the authors acknowledge that by applying this pruning step using DPI they may be discarding some direct interactions as well (Bansal et al., 2007). ARACNe involves a number of computational approximations and Monte Carlo simulations, which could make the method unstable (Markowetz and Spang, 2007). Sample size requirement is another limitation for such approaches as these methods perform acceptably well with high number of sample size (Hartemink, 2005).

2.2.3. ODE based approaches

In ODE based approaches the observed changes in activity of genes are related to each other via a set of ordinary differential equation (one for every gene). The ODE's describe the (instantaneous) change in each entity as a function of the levels of some network entities. To illustrate let us assume three genes G_1, G_2 and G_3 are interacting in such a way that G_3 is activated by G_2 and G_2. The activity of G_3 in such case will be a function of G_1 and G_2 (Figure 2.1) in terms of their activities (represented with square brackets) (Equation 2.8). Law of mass action is one of the popular ways to model these functions (f) (Dilao and Muraro, 2010). Such models are more often used in bottom up approach, nevertheless they have been applied to infer networks from experimental data (Bansal et al., 2007; Chen et al., 2004; di Bernardo et al., 2005; Li et al., 2008; Locke et al., 2005). These models being deterministic, infer causal relations among genes rather than mere dependencies.

$$\frac{d([G_3])}{dt} = f([G_1] \cdot [G_2]) \tag{2.8}$$

2.2.4. Boolean Networks

Boolean networks were proposed as biological network modeling paradigm by Kaufman *et al.* (Kauffman, 1993) They are a dynamic model of interactions between genes (represented by nodes) in a network. Boolean Networks consist of a directed graph $\mathcal{G}(\mathcal{V}, E)$. $\mathcal{V} \in (Gene_1, Gene_2 ... Gene_n)$ are the vertices of graph representing the genes that act like Boolean variables. Boolean networks are based on the postulate that each vertex in a graph

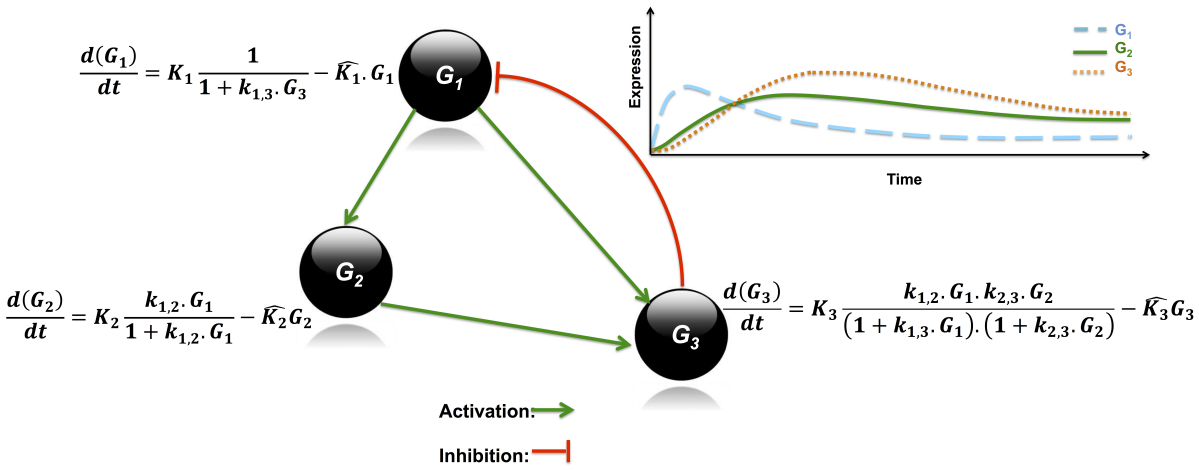

Figure 2.1.: A network with three genes with activities G_1, G_2 and G_3 interacting and the corresponding set ODE for the system. Redrawn from Karlebach *et. al.* k_{ij} represents reaction rate constants for i on j, K represents synthesis rate constant for corresponding genes and $\hat{K}_i$ is the degradation rate constant for gene i. Provided at the left is the activity measurement for the genes.

can attain two alternative levels: 1 or ON (active) and 0 or OFF (inactive). A Boolean function governs every vertex of the graph and they update the entire graph and referred as state transition (Figure 2.2). A graph with n genes can then have 2^n states.

Kauffman *et al.* used such models to analyze regulation and network stability in the yeast transcriptional network (Kauffman et al., 2003). He showed that random regulation functions make the network stable and biologically realistic functions increases the stability. Akutsu and co-workers devised an algorithm to infer genetic networks from state transition tables corresponding to time series of gene expression patterns(Akutsu et al., 1999). It has been applied to study Yeast cell cycle with a literature based Boolean network (Li et al., 2004). Lähdesmäki *et al.* developed an approach to find consistent Boolean networks from data (Lähdesmäki et al., 2003). GINsim [3] (Gene Interaction Network simulation) was introduced by Chaouiya *et al.* as is a computer tool to model and simulate genetic regulatory networks (Gonzalez et al., 2006). The Boolean networks were later extended with multi valued logic (de Jong, 2004; Schlatter et al., 2009). Shmulevich and colleagues extended this model with a hybrid probabilistic Boolean networks to cope up with uncertainties in biology (Shmulevich et al., 2002).

Boolean network are deterministic models that can model the dynamical behavior of a network. They can provide important insights into the network like-existence and nature of steady states, network robustness etc. Inherently, the model assumes the discretization of states (0 or 1). These discretization efforts on genes can lead to loss of information within the network. Furthermore, the modeling of self-down-regulation in genes is difficult with these models. Another issue with such models is their computational expense while analyzing

[3] http://gin.univ-mrs.fr ; Accessed: February, 2013

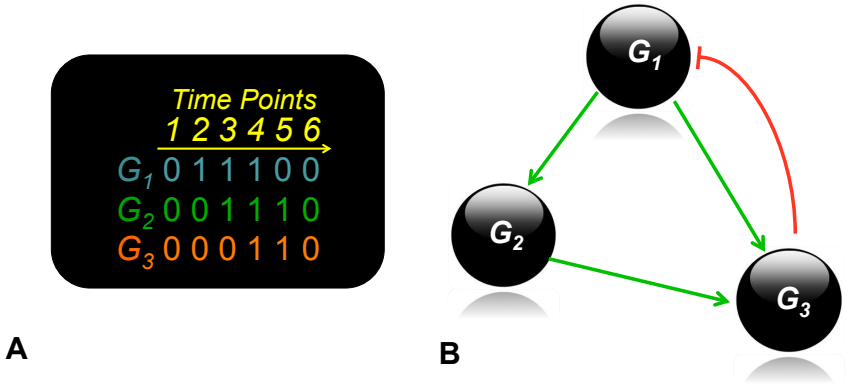

Figure 2.2.: A Boolean network representation of the same data as in figure 2.1 with three genes G_1, G_2 and G_3 representing the trajectory of the system through time together with discretized measurement of gene activities.

large networks, because the number of possible network states grows exponentially with the number of network nodes.

2.2.5. Probabilistic approaches

Deterministic models e.g. Boolean network or ODE models cannot cope with the noise in biological data. The measurement noise from experiments, uncertainty from experimental effects (intervention experiments), together with inherent biological noise (Pedraza and van Oudenaarden, 2005) disallows deterministic models to model biological networks robustly. In real life reverse engineering inference algorithms must cope with uncertainties in the data and relationships (between variables) (Frey and Jojic, 2005). Probability theory offers a mathematically robust way to formulate inference algorithms when reasoning under uncertainty. Therefore, a probabilistic approach becomes a preferred tool to model networks. Probabilistic Graphical Model (PGM) is a framework that combines uncertainty (probabilities) and logical structure (independence constraints) to compactly represent complex, real-world phenomena (Koller and Friedman, 2009). Thus, they offer to absorb the uncertainties and noise in real biological data (Bolouri, 2008).

In a probabilistic (graphical) model, molecules are represented as nodes. The measurements for each node are supposed to be drawn from a random variable. For example, if we consider genes as entities, their expression level can be the corresponding measurements. These models use presumed probability distributions of certain inputs to calculate the implied probability distribution for chosen output (Brémaud, 1998). Probabilistic models differs from a deterministic model, where one can model the relationship between molecules based on data with certainty. These models are popular way to represent a high dimensional system and the complex probability distribution over it in a compact way (Koller and Friedman, 2009). These model embodies the description of the joint probability distribution of all the random variables of interest (Friedman, 2004). The model consists of random variable as nodes and

edges as the relationship (Koller and Friedman, 2009). Probabilistic graphical models offer a common conceptual architecture where biological and mathematical objects can be expressed with a common, intuitive formalism as described below.

A probabilistic graphical model defines the independencies (conditional) among the nodes induced by the graph structure as well as the factorization induced by the graph structure. Bayes' rule is used to perform statistical inference in such models. The central idea of Bayes rule (Equation 2.9) is to update the belief in a hypothesis(X) given the additional evidence (Y) and the background information. X being the variable we care for and Y the evidence. Thus, the equation consists of three parts-posterior probability $P(X|Y)$ and the prior $P(X)$, likelihood $P(Y|X)$ gives the probability of evidence assuming hypothesis to be true. The term $P(Y)$ is independent of the hypothesis X and acts like a scaling factor. The overall function helps to compute the conditional probabilities.

$$p(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \tag{2.9}$$

As the current thesis is very much inspired by Bayesian Networks, following section offers a more detailed description for this model.

Bayesian Networks

Bayesian Networks (BNs) also known as belief networks are one of the most studied and used PGMs. They are used for many applications in science and engineering (Chavez and Cooper, 1990; Pearl, 1988). They provide a systematic representation of dependence among variables in terms of joint probability distribution. A BN builds graph for data for conditional independencies for all orders, thus two vertices are connected if no other vertex subset (S) can explain the conditional statistical dependency. Considering a set of vertices $G_1, G_2...G_n \in \mathcal{V}$ representing random variables in the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, two random variables G_1 and G_2 are conditionally independent given $G_3 \subseteq V \setminus \{G_2, G_2\}$ noted $(G_1 \perp\!\!\!\perp G_2|G_3)$, if the following holds

$$\forall \;\; G_1, G_2, G_3 : P(G_1|G_2, G_3) = P(G_1|G_3) \tag{2.10}$$

While, $\forall G_3$: $P(G_3) > 0$. We will deal with this concept in detail in the upcoming description.

What is Bayesian Network?

A Bayesian network is a graphical model for probabilistic relationships among a set of random variables $(X_1, X_2...X_n) \in \mathcal{V}$. The relationship is encoded as a DAG; Directed Acyclic Graph : $\mathcal{G}(\mathcal{V}, E)$ via conditional independence assertions among the random variables X_i. The DAG actually describes the dependency structure among the variables in a probabilistic way (Pearl, 1988; David Heckerman, 1994).

To illustrate we will consider a biological network as a graph consisting genes $G_1, G_2...G_5$ as vertices (Figure 2.3). Let us assume the expression of each gene is represented as a random variable. The graph depicts G_1 affecting G_2 which in turn affects G_3 and $G4$. These genes the ultimately cause G_5 to respond. A full specification of the joint distribution for these random variables (assuming binary values) requires 31 ($2^n - 1$; (n is the number of vertices)) parameters However around the Bayesian Network model formalism they can be described with fewer parameters. The formalism has been entailed below.

Mathematical Formalism

The node attributes in BNs are defined in terms of a local probability distribution (LPD) and joint probability distribution (JPD). This JPD is consistent with the independence assertions embedded in the graph $\mathcal{G}$. By applying the chain rule of probabilities and independence, the JPD is expressed as a product of conditional probabilities (2.11) (Heckerman et al., 1995a).

$$P(G_n, G_{n-1}...G_1) = P(G_n|G_{n-1}, G_{n-2}...G_1)P(G_{n-1}, G_{n-2}...G_1) \tag{2.11}$$

This can be generalized as follows for the graph in figure 2.3

$$P(\mathcal{G}) = \prod_{1}^{n} P(G_n|G_{n-1}, G_{n-2}...G_1) \tag{2.12}$$

The factorization of JPD over the graph structure. i.e. the local probability distribution for a given node i only depends on its parent nodes π_i.

$$P(\mathcal{G}) = \prod_{1}^{n} P(G_i|\pi_i) \tag{2.13}$$

On considering the parametrization of local probabilities θ

$$P(\mathcal{G}) = \prod_{1}^{n} P(G_i|\pi_i, \theta) \tag{2.14}$$

The set π_i consists of the parents of the random variable G_i. Thus given the parents a random variable is independent of all other random variables. Therefore, the JPD (joint probability distribution) can be decomposed as the product of conditional probabilities, only if the Markov assumption holds, that is, each variable G_i is independent of its non-descendants, given its parent in the directed acyclic graph $\mathcal{G}$. The conditional independence together with the local probability distribution for random variables, layout the relationships between the variables.

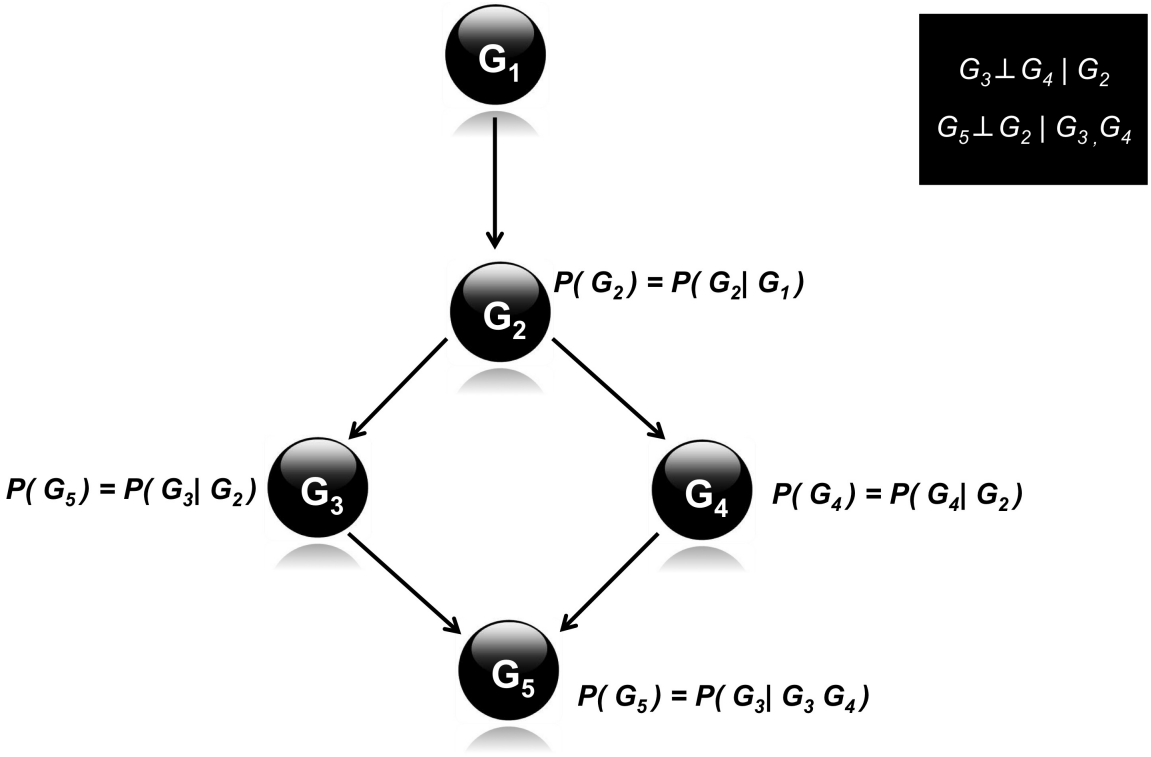

Figure 2.3.: A directed acyclic network with five genes G_1, G_2, G_3, G_4 and G_5 and their conditional independence in box (top-right)

A schematic overview of the theory underlying Bayesian networks is given in Figure 2.3. The conditional independence for the given graph is shown in the figure. The JPD for the given graph say $\mathcal{G}$ is given as equation 2.15.

$$P(\mathcal{G}) = P(G_5|G_4, G_3)P(G_3|G_2)P(G_4|G_2)P(G_2|G_1)P(G_1) \tag{2.15}$$

Why Bayesian Networks

The key feature of Bayesian networks is the fact that they provide a method for decomposing a probability distribution into a set of local distributions. The independence semantics associated with the network topology specifies how to combine these local distributions to obtain the complete joint probability distribution over all the random variables represented by the nodes in the network. This has three important consequences.

- Naïvely specifying a joint probability distribution with a table requires a number of

values exponential in the number of variables. For systems in which interactions among the random variables are sparse, Bayesian networks drastically reduce the number of required values.

- Efficient inference algorithms allow for transmitting information between the local distributions rather than working with the full joint distribution.
- The separation of the qualitative representation of the influences between variables from the numeric quantification of the strength of the influences has a significant advantage for network reverse engineering.

While building a Bayesian network model, one can focus first on specifying the qualitative structure of the domain and then on quantifying the influences. When the model is built, one is guaranteed to have a complete specification of the joint probability distribution. A BN for a given data, can thus be inferred multiple Bayesian Network structures which are likelihood equivalent, i.e. which equally well explain any data. In fact Bayesian Networks form equivalence classes, namely Completed Partially Directed Acyclic Graphs (CPDAGs).

Inferring Network

The key problem that remains is to learn the network structure i.e. the dependencies between random variables given data. Specially in bioinformatics the elucidation of biochemical circuitry (regulatory/signaling etc.) from experimental data is the major objective. The discovered graph $\mathcal{G}$ shows the dependencies across the genes, proteins or bio-molecules. The inference is though difficult can be achieved with careful experiment design and learning strategy (Needham et al., 2007).

To accomplish this, we need to infer a DAG satisfying the factorization (Equation 2.14) for the JPD that generates the data (Neapolitan, 2003). The JPD is not sufficient for the causal inference as it depends on the order of random variables too (Ellis, 2006; Heckerman et al., 1995a), moreover we usually have the data and not the JPD. The search space for entire set of possible networks grows exponentially with the number of random variables, making it an intractable problem to search through all possible networks (Koller and Friedman, 2009). One can test the independence of a pair of nodes for every subset of genes via constraint based learning (Pearl, 2000; Spirtes et al., 2001). However, the bottleneck with such approach is the computational feasibility if the graph is not sparse (Spirtes et al., 2001). Moreover this approach is also sensitive to errors in individual tests. For these reasons score based approaches are preferred.

Constraint based approaches use statistical conditional independence tests to find dependencies and independencies between variables. They then construct a Bayesian network to represent the set of independencies detected. Independence tests are often unreliable for more than a few variables, and constraint-based methods are sensitive to failures in these tests. This is because an edge can be incorrectly left out of a network based on the result of a single failed test.

Search-and-score methods treat structure learning as a model-selection problem. We define a set of possible structures and a scoring function that evaluates how well a structure fits the training data. We then use a search procedure to find a high-scoring structure. The space of Bayesian networks with n nodes has $2O^{(n^2)}$ possible structures (super exponential), and the problem of finding the best structure is in general NP-hard. Consequently, the search method (such as hill climbing or an evolutionary algorithm) will not consider all possible structures, and is not guaranteed to find the optimal structure. In this thesis, we use a search-and- score method to learn structures from a fully observed data set.

The central idea in score based learning is to assign a score to each network structure. We must find the directed acyclic graph $\mathcal{G}$ that best describes the data $\mathcal{D}$. This is performed by choosing a scoring function that evaluates each graph $\mathcal{G}$ (i.e. a possible network topology) with respect to the data $\mathcal{D}$, and then searching for the graph $\mathcal{G}$ that maximizes the score one based on Bayes rule equation 2.16. Here $P(\mathcal{G})$ is the structure prior.

$$P(\mathcal{G}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{G})P(\mathcal{G})}{P(\mathcal{D})} \tag{2.16}$$

The logarithm of posterior probability of the graph given the data was defined as a score S by Peèr *et al.* (Peér et al., 2001) (Equation 2.17).

$$S = log(P(\mathcal{D}|\mathcal{G})) + log(P(\mathcal{G})) \tag{2.17}$$

The marginal likelihood $P(\mathcal{D}|\mathcal{G})$ is the key component of Bayesian scoring metrics. It equals the full model likelihood averaged over parameters of local probability distributions and follows Baye‘s theorem (Heckerman et al., 1995a).

The concept of scores can then be applied to search all possible graph structures. Several search methods have been developed to seek the best fitting network for a data in the search space satisfying the above discussed mathematical formalisms. However, the problems turns unfeasible when the number of parent for each node in the graph exceeds 1 (Chickering et al., 1994). Exhaustively examining every possible graph is computationally expensive (NP complete problem) as the number of possible graph structures is super-exponential in the number of variables. This has motivated the use of heuristic approaches like- Monte Carlo methods (Hesar et al., 2012) which can be further aided by model selection methods (Chickering et al., 1997).

In case of not having large sample data, the structure prior $P(G)$ and parameter prior $P(\theta|G)$ can be handy Heckerman et al. (1995a). The parameter prior has been proposed by many workers based on different assumptions of Bayesian networks namely- equivalence, independence and modularity (David Heckerman, 1994; Cooper and Dietterich, 1992; Buntine, 1991). Structure priors can bias the network inference towards a specific graph space and restrict the search bout this space. They can be non-informative like equal prior (Heckerman et al., 1995b) or informative with expert knowledge (Madigan and Hutchinson, 1995). In case

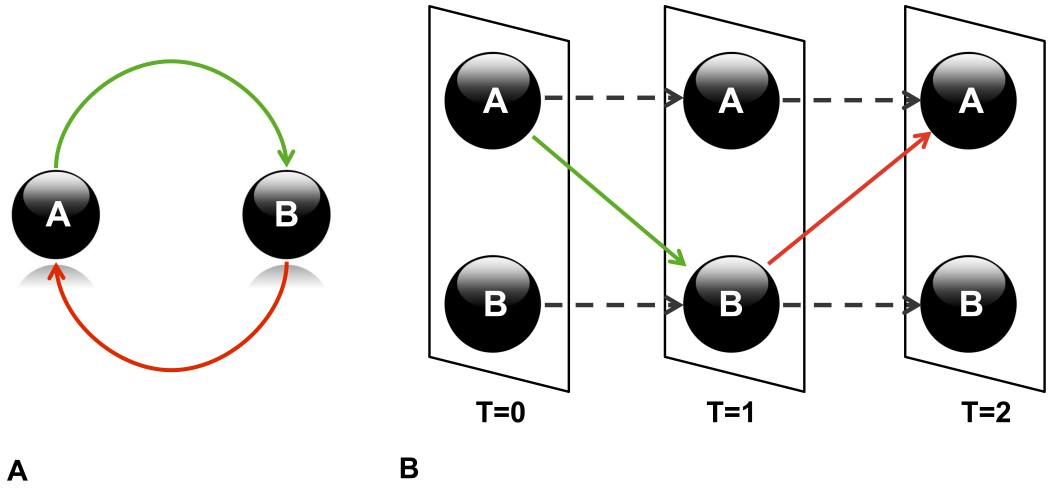

Figure 2.4.: (A) A feedback loop between two nodes A and B in rolled form. (B) Unrolled for the same network over time showing time layers and interactions between the genes across these layers. This forms the principle behind Dynamic Bayesian Networks

of reverse engineering of biological networks use of biological knowledge can prove vital as prior and yield sensible network models. This aspect of Bayesian Network will be exploited in chapter 5 and 6.

Dynamic Bayesian Networks :

Static Bayesian networks cannot handle temporal information. Furthermore, as BN model are based on DAGs, modeling feedback loops; a common phenomenon in biology becomes virtually impossible. The feedback loop in a graph is a cyclic element; turning a parent node into the child for its own child (Figure 2.4 (A)) . This paralyzes the conditional dependence framework of BNs. An extension of BN along the time dimension can overcome such situations and is known as Dynamic Bayesian networks (DBN) (Needham et al., 2007; Murphy, 2002). . They allow variables in a Bayesian network to be time dependent, in order to model time series or sequences (Ghahramani, 1997). In order to understand a dynamic Bayesian network we can simply unroll it across time. Unrolling a network converts it into its equivalent network (Figure 2.4 (B)) organized into layers indexed in terms of time steps (or points) (Perrin et al., 2003). The unrolled network shows how each variable at one time point ($T = t$) affects the variables at the next time point ($T = t + 1$). This extension of BN will be provide a ground for the work in chapter 4 of this thesis.

Application Stories :

Friedman et al. used Bayesian networks to establish regulatory relationships among yeast (*Sachharomyces cerevisae*) genes with time-series data of gene expression (Friedman et al., 2000). The approach was used to map several pathways from expression data (Hartemink et al., 2002a,b; Peér et al., 2001; Yu et al., 2004). Péer *et al.*, applied Bayesian networks to perturbation gene expression data, with a similar but expanded goal: to identify regulatory relationships and

in addition, to predict their nature of activation or inhibition. To do that, they incorporated perturbations into the Bayesian framework. Yeang and Vingron integrated perturbation data with knowledge from the literature into a joint model of regulation and metabolism and created a framework for the prediction of regulatory interactions and pathways in *Escherichia coli* (Yeang and Vingron, 2006). Banjo was developed by the group of Hartemink as a gene network inference software (Hartemink, 2005). It is based on Bayesian networks formalism and implements both Bayesian and Dynamic Bayesian networks. Therefore it can infer gene networks from steady-state gene expression data or from time-series gene expression data (Yu et al., 2004).

Beside being able to use noisy data directly from experiments, BNs offer additional features like the option for a prior to integrate different types of knowledge (Bolouri, 2008) which will be exploited in chapter 6 of this dissertation. Furthermore, they also offer flexibility in terms of node and the effect on it via other nodes even in case of incomplete data and are easy automate (Bolouri, 2008). This makes it an attractive choice to be used for biological network reverse engineering.

The issue that limited the Bayesian network while handling cycles in biological network was also resolve via the use of time differentiated data in Dynamic Bayesian Networks (DBN) Ghahramani (1997); Murphy (2002). The exploitation of time delays in biological networks allowed this by relaxing the acyclicity constraint of BN (Friedman, 1998). Ong *et al.* applied DBNs to learn physiological changes that affect tryptophan metabolism in E. coli (Ong et al., 2002). Perrin *et al.* applied DBNs on the S.O.S. DNA Repair network of E.coli (Perrin et al., 2003).

Inferring gene regulatory networks from high throughput gene expression data is another challenge. The reasons behind this include, fewer data points and also because of the random noise that is present in experimental measurements. We need to find the optimal network based on such data. The number of possible DAGs for the data set can be huge, even for small number of genes. For example, 20 genes can have more than 10^{72} possible DAGs. Therefore, effective methods for learning networks together with smart experimental designs are needed to find a better solution (Kim et al., 2003).

The current chapter presented a brief overview of network reverse engineering methods with a special focus on Bayesian Networks. BNs were discussed in detail as they will serve as the ground for a big part of work done during the thesis as well as for Nested Effects Models (NEMs); the major framework of this thesis. The next chapter will introduce these NEMs

Chapter 3
Nested Effects Model(s)

Following the last chapter on network reverse engineering methods, this chapter presents an overview of Nested Effects Models which will the 'prima focus' for this thesis. The chapter explains the principle and mathematical formalism for NEM. This will serve as the foundation for the next chapters which aim to extend and compliment the NEMs in various dimensions.

3.1 Introduction

A cell when subjected to an external cue or stimuli responds in a complicated way in terms of the activities of various bio-molecules like- gene or proteins. Biological networks or pathways work like water ripples and the effect of disturbance are carried throughout the surrounding bio-molecules according to upstream or downstream relations. The response propagation can reflect the machinery of bio-molecular circuitry ultimately unveiling the network involved. Even if the perturbed bio-molecule cannot be measured directly, other molecules linked to it may be measured and therefore their activity can give an insight into the activity of perturbed gene. Probabilistic graphical models are the representation of the dependency structure between the components of a random variable as described in the last chapter. However, the key in modeling such perturbation measurement based networks is mapping of the downstream effects to perturbation. NEMs are a class of probabilistic methods that explicitly addresses this issue of reverse engineering from such indirect measurements.

3.2 NEM(s)

NEMs (Nested Effects Models) are a class of probabilistic models to reconstruct a pathway based on measurable transcriptional downstream response profiles of individual gene knock-downs. Perturbation typically alters function of the biological system. A targeted perturbation typically brings about an observable change in the steady-state expression/activity levels of every gene in the network in the presence of the perturbation (Ideker et al., 2000; Badaloni et al., 2012). Such changes can be much more informative as it also reveals the effects on the downstream genes (bio-molecules) within the system (Nelander et al., 2008; Wagner, 2002). The perturbed genes are called the *S-genes* (silenced genes) and the expression profile is measured for *E-genes* (effect reporter genes). The NEM framework aims to systematically

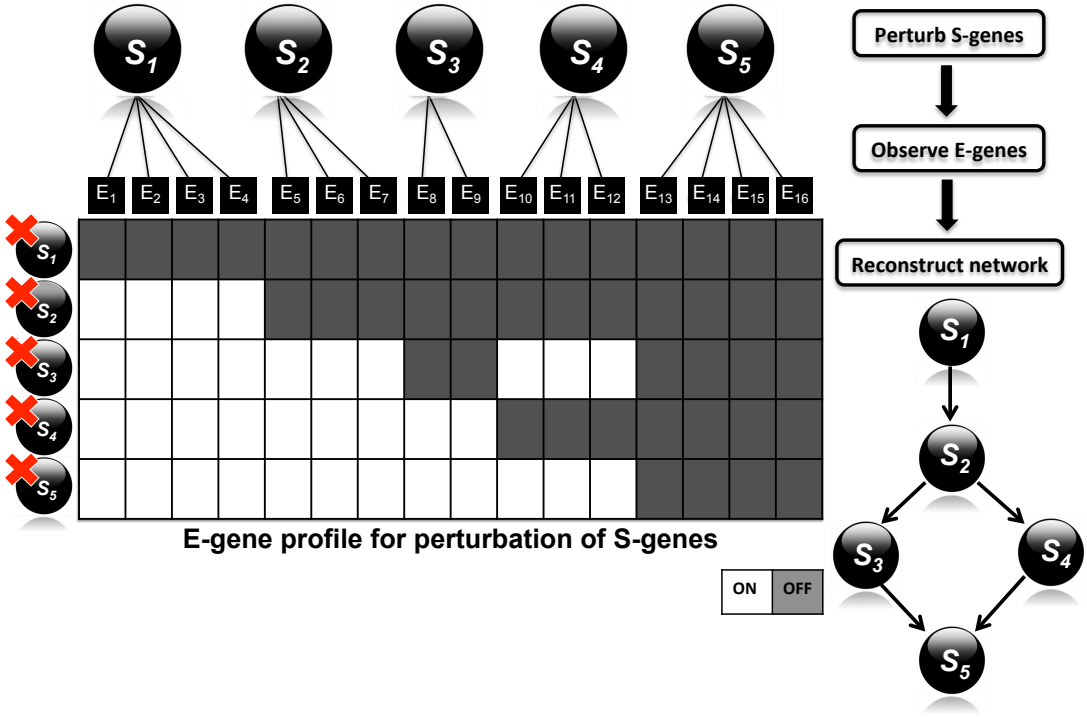

Figure 3.1.: General principle of Nested Effects Model (NEM)

model the upstream signal flow network between S-genes via a directed graph based on the profiles of downstream genes (E-genes). The approach entails the nested effect on E-genes for perturbation of every S-gene.

3.2.1. The approach

The NEM algorithm accepts a set of hypotheses in terms of networks among the perturbed/silenced genes. Each of these hypotheses is a graph $\mathcal{G}(\mathcal{V},\mathcal{E}) : \mathcal{V} \in \mathcal{S}$ and makes use of Bayesian scheme to score the fit to the data. However, the score is not unique to networks as different networks can have identical scores i.e. networks form likelihood equivalence classes. One can then consider the entire list of network and the overall confidence in each possible edge among the silenced genes. First, we present here the framework of NEMs as proposed by Markowetz (Markowetz, 2005), followed by other frameworks and advancements from Fröhlich *et al.* (Fröhlich et al., 2007a), Tresch *et al.* (Tresch and Markowetz, 2008), Zeller *et al.* (Zeller et al., 2009) and Vaske *et al.* (Vaske et al., 2009)

Model semantics

To illustrate the semantics of NEMs let us assume a data matrix $\mathcal{D}$. The matrix has the dimensions E-genes$\times$ S-genes. It shows the phenotypic profiles (of E-genes E) for perturbed

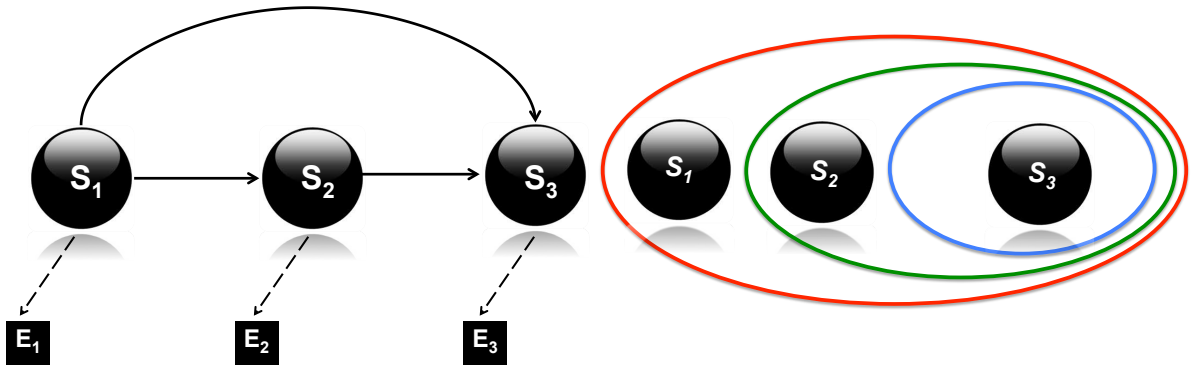

Figure 3.2.: Scheme of perturbation effects and their nested nature representing the principle behind Nested Effects Models. Interventions of S-genes interrupts signal flow through the pathway. S-genes regulate E-genes at secondary level. The E-gene expression effects for downstream S-genes are nested within its upstream S-gene.

genes S. For simplicity we assume a binary data (observed effect and not observed effects). For example, $\mathcal{D}_{ij} = 1$ means E-gene i shows an effect upon silencing of S-gene j and $\mathcal{D}_{ij} = 0$ means E-gene i shows no effect upon silencing of S-gene j.

According to the model as described above, the effects observed when perturbing a gene upstream forms a super-set to the effects observed for all other genes (Figure 3.2). To illustrate, let us assume three genes S_1, S_2 and S_3. Let, S_1 be upstrem of S_2 and S_3 and S_2 is upstream of S_3 as in figure 3.2. Each of these have effects gene (E-genes), E_1, E_2 and E_3 respectively. Perturbing S_1 casts effects on the S-genes shows effect on their downstream genes S_2 and S_3; as they are located downstream. They can be measured in terms of corresponding E-genes. Thus, the effects of perturbing S_1 forms a super set for the effects of S_2 and S_3 and so on (Figure 3.2). The NEM estimates a graph $\mathcal{G}$ that depicts these causal relations across the S-genes as well as the associations between E-genes and S-genes (Tresch and Markowetz, 2008; Markowetz, 2005).

Mathematical formalism

A general 'Effects Model' forms a matrix $\mathcal{M}$ of dimensions $\mathcal{A} \times \mathcal{O}$ where, $\mathcal{A}$ is the action and $\mathcal{O}$ is the observable. Thus, there is a binary value associated with every observable $\mathcal{O}$ for each action $\mathcal{A}$ (Tresch and Markowetz, 2008). For example, if action $\mathcal{A}_j$ has an effect of observable $\mathcal{O}_i$ then $\mathcal{M}_{ij} = 1$; else 0.

Now, let us consider the case of gene perturbations where, every perturbation is an action and one action can imply the other action. This forms our first graph Φ, i.e association among actions. Additionally, we have the observable for each of these actions. These two association can be modeled as an extension of the general effects model defined above, we call this Θ. A Nested Effects Model or NEM, is a combination of these two components thus forming an extension of a general effects model plus the relationships across the actions (Tresch and Markowetz, 2008). It can be given as the product of Φ and Θ.

$$\mathcal{M} = \Phi\Theta \tag{3.1}$$

To further illustrate the NEMs mathematically, let us consider two set of genes $\mathcal{S}$ and E consisting of S and E-genes respectively. Let $\mathcal{D}$ be a binary matrix representing the effect of every perturbation. $\mathcal{D}_{ij} = 1$, if E-gene i shows an effect in a targeted perturbation experiment j, else 0 for no effect (Markowetz, 2005).

$$S_1, S_2, ...S_n \in \mathcal{S}$$

$$E_1, E_2, ...E_m \in E$$

$$\mathcal{D} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

NEMs assume that perturbation of S-gene S yields measurable downstream effects for all E-genes attached to S itself and all E-genes attached to any S-gene reachable from S in the upstream signal flow network. Thus, the information for perturbation is carried to every gene in the network according to connection between the perturbed genes and further communicated to the corresponding downstream genes systematically. For example, in the network in figure 3.2 perturbing gene S_1 will affect the gene S_1 itself and the silencing information will then show effect on genes the downstream genes i.e. E_1. Similarly S_2 being downstream to S_1 will get the silencing information and its downstream genes E_2 will see the change in their expression. Whereas, perturbing S_2 will not affect S_1 as there is no flow of information from S_2 to S_1. This gives rise to the subset relationships among the E-genes (Equation 3.2). Such subset relations being reflexive and transitive define a quasi order on the expression profile data. This brings in the equivalence classes among networks in form of transitively closed graphs (Box 3.1).

$$S_1 \rightarrow S_2 \Leftrightarrow \{i : P_{S_2}(E_{S_2} = 1)\} \subseteq \{i : P_{S_1}(E_{S_1} = 1)\} \tag{3.2}$$

Based on the illustrations explained so far, the entire set of cross talk among signals and effects can be organized in and as two binary matrices namely Φ and Θ. The matrix Φ. The matrix Φ is of dimension $|\mathcal{S}| \times |\mathcal{S}|$, and represents the connections among S-genes (signals). Whereas, the matrix Θ of dimension $|\mathcal{S}| \times |E|$, represents the connection between signals and effects (S and E-genes). $\Phi_{ij} = 1$ shows the presence of an association between signals i and j, and $\Phi_{ij} = 0$ $(i, j \in \mathcal{S})$. We also assume $\Phi_{ii} = 1$. If one assumes transitively closed S-gene graphs Φ (as done in the following), then Θ is just an attachment graph. That means $\Theta_i j = 1$, if and only if E-gene j is attached to S-gene i. We can now illustrate this organization in figure 3.2 with the following Φ and Θ matrices:

$$\Phi = \begin{bmatrix} & S_1 & S_2 & S_3 \\ S_1 & 1 & 1 & 1 \\ S_2 & 0 & 1 & 1 \\ S_3 & 0 & 0 & 1 \end{bmatrix}, \Theta = \begin{bmatrix} & S_1 & S_2 & S_3 \\ E_1 & 1 & 0 & 0 \\ E_2 & 1 & 1 & 0 \\ E_3 & 1 & 1 & 1 \end{bmatrix}$$

For a given data $\mathcal{D}$, Φ is the network hypothesis and NEM aims to infer the posterior to infer the graph among signals (S-genes). Applying Bayes' formula for network hypothesis Φ, we have

$$P(\Theta) = \frac{P(\mathcal{D}|\Phi)P(\Phi)}{P(\mathcal{D})} \tag{3.3}$$

Considering the association of E-genes to S-genes (Θ) the data likelihood becomes as follows

$$P(\mathcal{D}|\Phi) = \prod_i P(\mathcal{D}_i|\Phi, \Theta_i = 1) \tag{3.4}$$

If the effect of each perturbation is known, the model described above can be said to be a complete model. However, in real condition the effect for each perturbation is not known. Therefore, we marginalize over Θ to get overall likelihood as in equation 3.5

$$P(\mathcal{D}|\Phi) = \prod_{k \in E} \sum_{j=1}^{\mathcal{S}} P(\mathcal{D}_i|\Phi, \Theta_{i=j}) P(\Theta_{i=j}|\Phi) \tag{3.5}$$

Simplifying this for perturbation and effects (signals and effects), we have:

$$P(\mathcal{D}|\Phi) = \int_\Theta P(\mathcal{D}|\Phi, \Theta) P(\Theta)$$

$$= \prod_{k \in E} \sum_{s \in \mathcal{S}} \prod_{t \in \mathcal{S}} P(\mathcal{D}_{tk}|\Phi, \Theta_{sk} = 1) P(\Theta_{sk} = 1) \tag{3.6}$$

The Bayesian structure in equation 3.3 allows a structure prior in the model. Such a prior can be used to bias the network search towards prior knowledge (Fröhlich et al., 2007a). This structure prior can be decomposed in terms of individual edge wise priors as in equation 3.7, which are statistically independent. This avoids the overfitting of model when based only on data.

$$P(\Phi) = \prod_{i,j} P(\Phi_{i,j}) \tag{3.7}$$

Computing effect likelihood

The original work of Markowetz *et al.* is based on the assumption of having both positive and negative control together with RNAi perturbation data (Markowetz, 2005). Positive and negative controls refers to the conditions of having data with stimulation (but without RNAi) and without stimulation respectively. This enables to simply count the observed knockdown effects for every replicate in a perturbation experiment as a measurement for observables. However, in absence of positive and negative controls, we cannot use the counting approach to get effects matrix. In most of the cases we have only a single control compared to treatment (RNAi effects).

Fröhlich *et al.* proposed a Beta Uniform Mixture models (BUM models) to address the issue (Fröhlich et al., 2007a, 2008b). The model assumes the data matrix ($\mathcal{D}$) to be composed of p-values obtained via differential gene expression analysis (using methods like $limma$ (Smyth, 2004)) compared to a single control. Under the null hypothesis, the p-values will have a uniform density corresponding to a flat horizontal line. Whereas, under the alternative hypothesis, the p-values will have a distribution that has high density for small p-values and the density will decrease as the p-values increase (Pounds and Morris, 2003). Thus, the overall distribution is a mixture of p-values arising from these two hypotheses, whose shape can be given as equation 3.8. The distribution can be visualized as in figure 3.3. The shape of the probability density function (PDF) $f(x|a,\lambda)$ is defined by a curve with asymptote at $x = 0$ and follows a monotonic decrease to its minimum (Figure 3.3).

$$P(D_{ik}) = \gamma_k + (1-\gamma_k) \cdot f_1(\mathcal{D}_{ik}), \gamma \in (0,1) \tag{3.8}$$

Thus under the alternative hypothesis, we will have a higher density for small P-values and a strong decrease for increasing P-values. Further the mixing coefficient $\gamma \cdot P(\mathcal{D}_{ik}|\Phi,\theta_i)$ can be given as:

$$P(\mathcal{D}_{ik}|\Phi,\theta_i) = \begin{cases} f_1(\mathcal{D}_{ik}) & \text{if } \Phi \text{ shows an effect} \\ 1 & \text{otherwise} \end{cases} \tag{3.9}$$

A better fit has been proposed (Fröhlich et al., 2008b) as a three component mixture of (1) a uniform, (2) a $Beta(1,\beta_k)$ ($\beta_k > 2$), and (3) a $Beta(\alpha_k,1)$ ($\alpha_k < 1$). This distribution can be given as:

$$f(\mathcal{D}_{ik}) = \pi_{1k} + \pi_{2k} Beta(\mathcal{D}_{ik},\alpha_k,1) + \pi_{3k} Beta(\mathcal{D}_{ik},1,\beta_k) \tag{3.10}$$

Where, $\pi_{1k} + \pi_{2k} + \pi_{3k} = 1$ ($\pi_{rk} \geq 0, r = 1,2,3$).

Fröhlich *et al.* proposed that the beta distributions ((2) and (3)) used here are monotonously decreasing convex functions. The minimum for $Beta(1,\beta)$ is always 0, whereas $min\,(Beta(\alpha,1))$ > 0. The density function f_1 reflects the strength of the knock-down effect on E-gene i under the alternative hypothesis. If it is greater 1 the alternative hypothesis would be accepted, and if it is smaller 1 rejected. An Expectation Maximization algorithm can be used to fit such a model (Dempster et al., 1977) and obtain the shape parameters for the distribution. To extract the

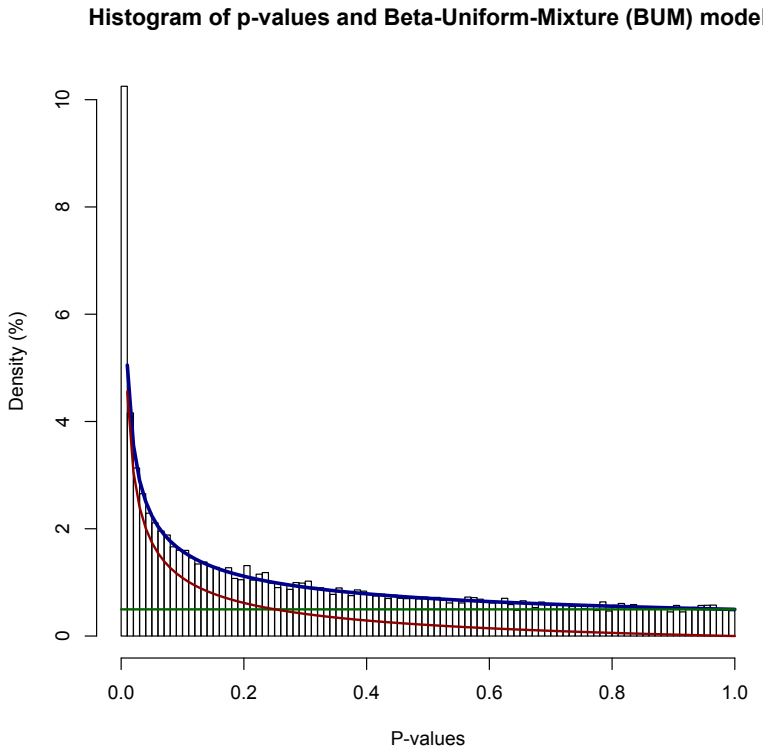

Figure 3.3.: A fitted BUM model to the histogram. The green line indicates the uniform part and red the $Beta$ part. The BUM model is represented by the blue line.

alternative distribution f_1 let us start with $\hat{\pi} = f(1)$ as the maximum uniform part of the BUM model, then we have:

$$f(\mathcal{D}_{ik}) = \frac{f(\mathcal{D}_{ik}) - \hat{\pi}}{1 - \hat{\pi}} \tag{3.11}$$

We will use this model while generating artificial data for our simulation studies in upcoming chapters.

Learning network structure

The S-gene graph in NEMs represents a quasi order [1] between S-genes. The number of possible quasi orders increases exponentially with the number S-genes. Therefore, exhaustive enumeration of the whole space of possible S-gene graphs is only possible for a small number of S-genes. For example, for 4 S-genes ($n = 4$) there are 355 possible network structures and for n=5 there are 6942 possible network structures. This brings in the heuristic methods into the picture. A set of methods have been proposed to handle the issue (Markowetz and Spang, 2007; Fröhlich et al., 2007a). The simplest among these methods are the pairwise and triplet inference methods (Markowetz and Spang, 2007).

[1] http://oeis.org/search?q=quasi+order&sort=&language=english&go=Search ; Accessed: February 2014

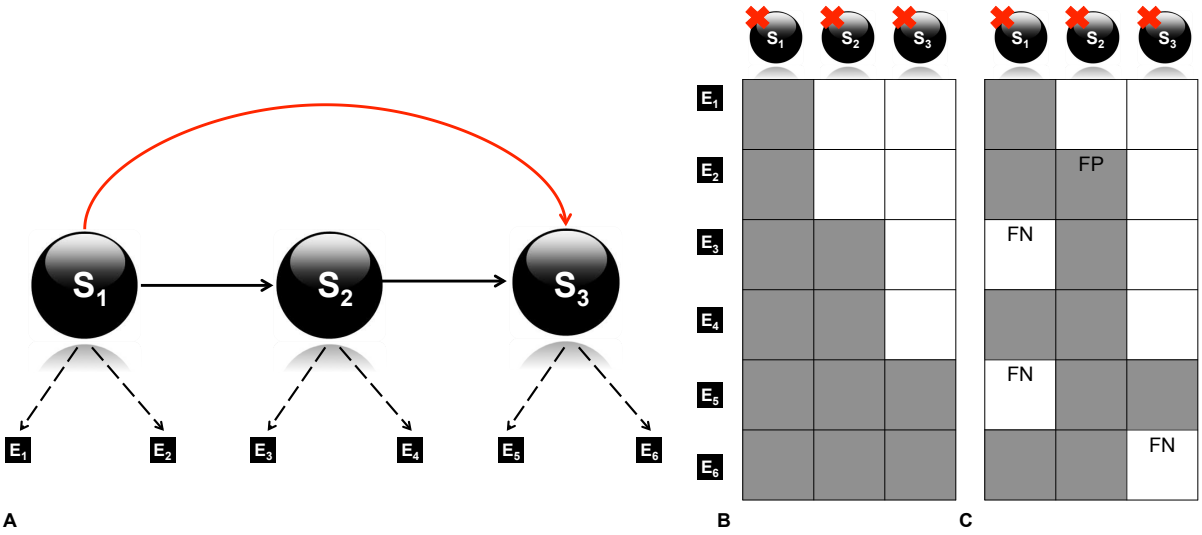

Figure 3.4.: Transitive closure in the model and the perturbation effect. (A) Shows a transitively closed graph with the shortcut edge ($S_1 \rightarrow S_3$) in red (B) Expected effect of perturbation: without noise (C) Effect of noise in observed data. Redrawn from Markowetz et al. 2007.

The pairwise relations can be inferred by choosing between four models for each gene pair S_1 and S_2. In this case we can have only four possible models (Equation 3.12). For each pair a Bayesian score for all these possible models can be computed and the one with *maximum aposteriori* (MAP): (M_{S_1,S_2}) is selected. The advantage is the reduction of search space. The search space for n genes is $\binom{n}{2}.4$. This function grows quadratically with n. For example, in case of 2 genes the search space will be 4 and in case of 3, 4 and 5 genes it will be 12, 24 and 40 respectively.

$$S_1, S_2 \Rightarrow \begin{cases} S_1 \rightarrow S_2 \\ S_1 \leftarrow S_2 \\ S_1 \leftrightarrow S_2 \; (indistinguishable) \; or \\ S_1 \not\leftrightarrow S_2 (no - relation) \end{cases} \tag{3.12}$$

Although the pairwise inference method is faster, it treats the involved edges independently. In transitive graphs e.g. figure 3.4 (A), where we have a path $S_1 \rightarrow S_2 \rightarrow S_3$, it is possible to have shortcuts between S_1 and S_3 as an edge $S_1 \rightarrow S_3$. In real data, noise (FP and FN in figure 3.4(C)) can cause non detection of edges or detection of spurious edges. For example, in figure 3.4 the FP and FN in the observed data can cause non detection of the edge $S_1 \rightarrow S_3$ as their perturbation effects do not overlap. Dealing with a collection of sub-models consisting triplets of interacting vertices can address the issue to some extent.

The triplet method, offers a slight extension in terms of considering triplets of nodes in a model compared to pairs. It extends the inference method beyond the assumption of

independence between edges. In case of triples the models space for n gene is $\binom{n}{3}.29$. This grows cubical as $O(n^3)$. Thus, for a three gene model, 29 possible quasi order structures equivalently: transitively closed graphs can be scored to select the MAP model (M_{S_1,S_2,S_3}). Later to combine the triplets in a model to one graph an edge-wise model averaging is used. Since all possible triplets are scored, for each edge we can compute the frequency, in how many sub-models it appears. This implies the count of an edge in a model (model's confidence in an edge) (Markowetz and Spang, 2007). Thus for each edge the confidence function is given as:

$$f(S_1, S_2) = \frac{1}{n-2} \sum_{z \notin \{S_1,S_2\}} 1[S_1 \to S_2 \in M_{(S_1,S_2,S_3)}] \tag{3.13}$$

Where, $1[.]$ indicates the presence of corresponding edge in a model $M_{(S_1,S_2,S_3)}$. The overall graph is constructed from edges with confidence greater than a defined threshold. The triplet approach takes into account that all possible triplets. Hence, there is no assumption of independence; an advantage over the pairwise method. The transitive closure $\mathcal{G}^* = (\mathcal{V}, \mathcal{E}^*)$ of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ requires $(i,k) \in \mathcal{E}^* \Leftrightarrow (i,j) \in \mathcal{E}$ and $(j,k) \in \mathcal{E}\ \forall\{i,j,k\} \in \mathcal{V}$ (BOX 3.1). This means that if there is a path from S_1 to S_n via some other nodes say $(S_2, \ldots S_{n-1})$, a path directly reaching S_n cannot be disregarded. Although triplet methods do not always guarantee to yield a transitively closed graph, they reach much closer to a quasi-order compared to the pairwise approach (Markowetz and Spang, 2007). Another issue can be inclusion of spurious edges owing to the consideration of smaller data and the noise in pairwise method is overcome by the triples' method.

Module network approach was proposed by Fröhlich *et al.* to address the issue of large networks in NEMs (Fröhlich et al., 2007a). The module network approach is based on the idea that a network consists of smaller sub-graphs called modules. The data set is first split into sets of at most 4 nodes that serve as modules via hierarchical clustering based on p-value density profiles. The network across these S-genes is learned via exhaustive search. To join the modules Fröhlich et al. proposed two approaches (1) successively link pairs of modules together using the pairwise model (Fröhlich et al., 2007a) and (2) guide the link of module pairs via the increase of the log-likelihood of the complete network (Fröhlich et al., 2008b). It establishes connection between modules by connecting pairs of nodes from different modules. The module network approach can be applied to learn larger networks and is biologically more sensible as the modular nature is evident in biological networks (see chapter 1).

3.2.2. Connection of NEMs to Bayesian Networks

The formulation for NEMs (equation 3.6) as a Bayesian network was presented by Zeller *et al.* in 2009 (Zeller et al., 2009). The BN setting requires to assume an acyclic graph between S-genes.

To illustrate, the NEM graph can be seen as two component system, first being the observed E-genes and second the perturbed S-genes. The S-genes component in the NEM represent

BOX 3.1: Transitive Closure in Graphs

The transitive closure of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a graph $\mathcal{G}^* = (\mathcal{V}, \mathcal{E}^*)$ such that $\mathcal{E}^*$ contains an edge (u, v) if and only if $\mathcal{G}$ contains a path from u to v. In other words, the transitive closure of a graph is a graph which contains an edge (u, v) whenever there is a directed path from u to v. Algebraically, the transitive closure of a network can be expressed as an infinite sum of the true direct network and all indirect effects along paths of increasing lengths, which can be written in a closed form as an infinite-series sum (Feizi et al., 2013). The transitive reduction of an acyclic graph $\mathcal{G}$ is the unique smallest graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^*)$, i.e., with the least number of edges, such that $(\mathcal{G}^*) = \mathcal{G}$. Intuitively, this means that the transitive closure is preserved by the reduction, i.e., no information about reachability is lost. For an acyclic graph $\mathcal{G}$ it can be shown that the $\mathcal{G}^*$ can be obtained by removing each redundant edge $\mathcal{E}_{u,v} \in \mathcal{E}$ from the original graph $\mathcal{G}$ for which there is an indirect path, i.e., not including edge $\mathcal{E}_{u,v}$, between u and v in $\mathcal{G}$ (Bonaki et al., 2012). Figure 3.5 shows the transitive reduction of a directed graph having 5 vertices.

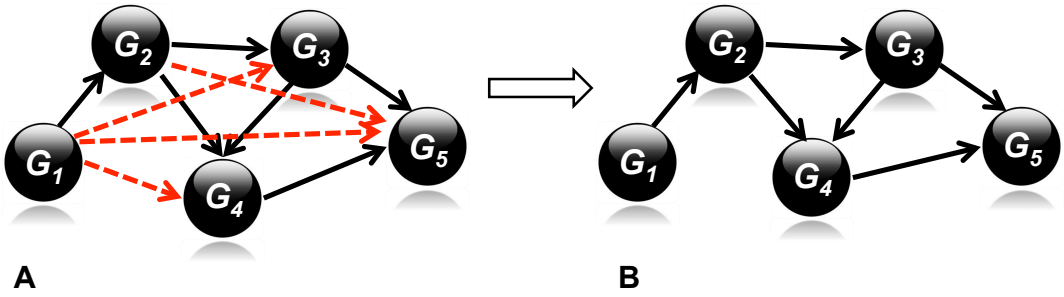

Figure 3.5.: Diagram representing the phenomenon of transitive closure in graphs. Here the graph A is reduced to B via transitive reduction. The red dashed edges in graph A represent the transitive information flow in terms of indirect edges

latent variables (not observed directly). This S-gene component ($\mathcal{S}$) is rather observed through the effect component; the E-genes (E). For simplicity, let us assume the effect component to be a binary variable, i.e. they carry a value 1 and 0 in active and inactive states respectively. These two components which can be combined into a single graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, such that:

$$V(\mathcal{G}) = \mathcal{S} \cup E \tag{3.14}$$

Thus, the overall graph $\mathcal{G}$ is a DAG (Directed Acyclic Graph) consisting of the 2 levels in hierarchy (1) latent variable S , and (2) observed effect component, representing the Φ and Θ components respectively in the original NEM framework. In the graph $\mathcal{G}$ (Figure 3.6). An effect e ($e \in E$) is observed when it is reachable from a signal s ($s \in \mathcal{S}$). For example in figure 3.6. E_1 is reachable from S_1 as well as S_2 so a signal from either of these vertices will lead to an

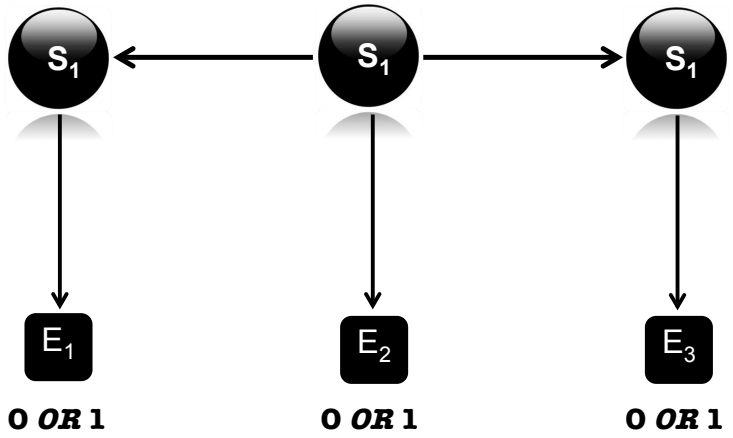

Figure 3.6.: The Bayesian Network view of NEMs, with assumed Boolean observation (observable E-genes) for simplicity

observation of effect of E_1. On the other hand, E_2 is reachable only from S_2, therefore, there an effect on E_2 can only be observed with the signal S_2.

Now, let $\mathcal{D}_{se}$ be the observation in our data for effect e_{se} i.e. e caused by signal s. The local probabilities $\mathcal{L} = P(\mathcal{D}_{se}|e_{se})$, assuming the data independence, the likelihood for NEM framework can be given as

$$P(\mathcal{D}|\mathcal{G},\mathcal{L}) = \prod_{s\in S}\prod_{e\in E} P(\mathcal{D}_{se}|e_{se}) \tag{3.15}$$

From a BN perspective, $\mathcal{G}$ consists of a set of random variables $\mathcal{V}$ and local probability distribution parameters $\mathcal{L}$. Thus, in NEM we have to model $\mathcal{S}(S \subset \mathcal{G})$ given the observed effects e ($e \in E, E \subset \mathcal{G}$). For the NEM graph $\mathcal{G}$ a node x (where, $x \in S \cup E$) is unaffected as long as its immediate parents are affected (Equation 3.16):

$$P(x = 1|pa(x)) \begin{cases} 1 & : if \;\; max(pa(x)) = 1 \\ 0 & : otherwise \end{cases} \tag{3.16}$$

When a node (here S-gene) is perturbed ($S \rightarrow S^*$), the state of each node changes based on the connections of the graph. This, in turn, determines the conditional probability distribution of the observable nodes based on the data. For each data $\mathcal{D}_i$ (observed data under perturbation of S-gene i) and observable $e \in E$ (Zeller et al., 2009) we have equation 3.17. Where, the first part scores the likelihood of a perturbation of e relative to the opposite (i.e. when e is not perturbed).

$$P(\mathcal{D}_i) = \prod_{e \in E | \mathcal{D}_i e=1} \frac{p(\mathcal{D}_{ie}|e=1)}{p(\mathcal{D}_{ie}|e=0)} . \prod_{e \in E} p(\mathcal{D}_{ie}|e=0) \tag{3.17}$$

For a set of n perturbations we have a set of corresponding data $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2 ... \mathcal{D}_n\}$. Assuming independence the data observations:

$$P(\mathcal{D}) = \prod_{i=1}^{n} \mathcal{D}_i \tag{3.18}$$

The optimal graph can be computed via maximization of this score discussed in Bayesian Network section of last chapter (section 2.2.5).

$$P(\mathcal{G}|\mathcal{D}) = \int P(\mathcal{G}, \Theta|\mathcal{D}) d\Theta$$

$$\propto P(\mathcal{G}) \int P(\mathcal{D}|\mathcal{G}, \Theta) P(\Theta) d\Theta \tag{3.19}$$

Considering the Bayesian Network setting introduced by Cooper and Herskovits (Cooper and Herskovits, 1991) (See chapter 2) marginal $p(\mathcal{D}|\mathcal{G})$ can be calculated in closed form for binary observations. In this context additional beta distribution priors (with shape parameters α and β) for the binomial distributions, which observations are supposed to follow, can be defined. Accordingly, the Cooper and Herskovits formula reads as:

$$P(\mathcal{D}_1 \dots \mathcal{D}_n|\mathcal{G}) = \prod_{j=1}^{N} \prod_{e \in \varepsilon} \prod_{i \in (0,1)} \frac{\Gamma(N_{e,i,0} + \alpha_i)\Gamma(N_{e,i,0} + \beta_i)\Gamma(\alpha_i + \beta_i)}{\Gamma(N_{e,i,0} + \beta_i + \beta_i)\Gamma(\alpha_i)\Gamma(\beta_i)}$$

$$\propto \prod_{j=1}^{N} \prod_{e \in \varepsilon} \prod_{i \in (0,1)} \frac{\Gamma(N_{e,i,0} + \alpha_i)\Gamma(N_{e,i,0} + \beta_i)}{\Gamma(N_{e,i,0} + \beta_i + \beta_i)} \tag{3.20}$$

For the case of normally distributed observations the Cooper-Herskovits formula can be derived as:

$$P(\mathcal{D}_1 \dots \mathcal{D}_n|\mathcal{G}) = \prod_{e \in \varepsilon} \prod_{i \in (0,1)} \left(\frac{1}{2\pi}\right)^{N_{ek}/2} \sqrt{\left(\frac{v}{v + N_{e,k}}\right)} \frac{\Gamma((\alpha + N_{e,k})/2)}{\Gamma(\alpha/2)}.$$

$$\frac{|\beta|^{\alpha/2}}{|\beta + s_{e,k} + (v_{nek}/(v + N_{e,k}))(\bar{x_{e,k}} - \mu)^2|^{(\alpha+N_{e,k})/2}}$$

$$\propto \prod_{e \in \varepsilon} \prod_{i \in (0,1)} \sqrt{\left(\frac{v}{v + N_{e,k}}\right)} \frac{\Gamma((\alpha + N_{e,k})/2)}{|\beta + s_{e,k} + (v_{nek}/(v + N_{e,k}))(\bar{x_{e,k}} - \mu)^2|^{(\alpha+N_{e,k})/2}} \tag{3.21}$$

It is a major achievement of NEMs to restrict the topology of the underlying graphical structure in a sensible yet highly efficient way, thus, tremendously reducing the size of the search space. There is an arbitrary "core" network consisting of signal nodes, and there is a very sparse "marginal" network connecting the signals to the effects. It is, however, by no means necessary that the core network and the signal nodes coincide.

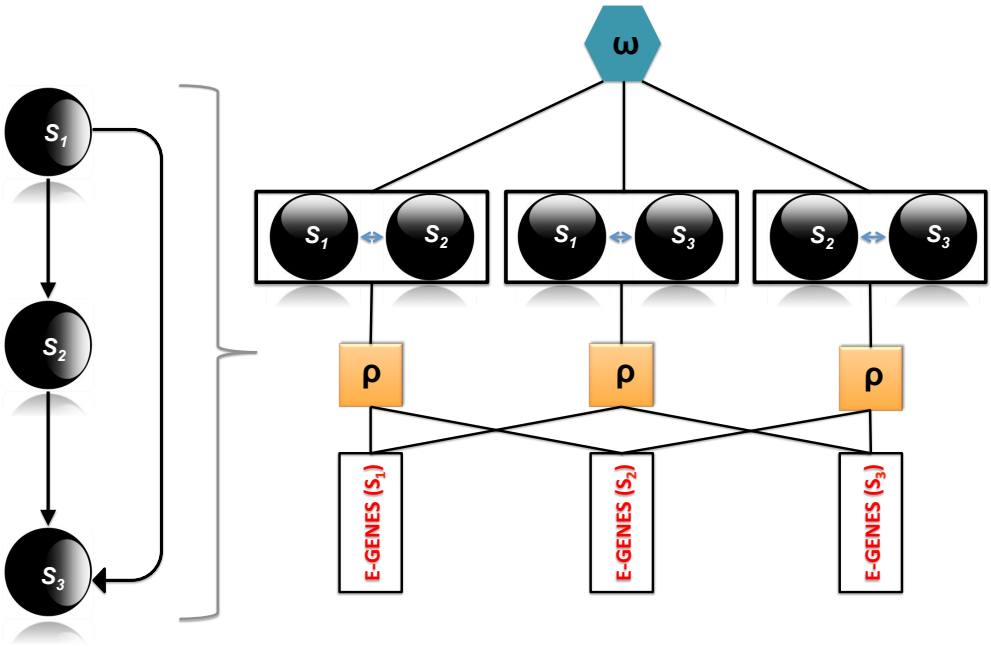

Figure 3.7.: The factor graph model for NEMs. ω (in blue box) represents the transitive factor and ρ (orange box) represents the interaction factors. On the left is the graph (consisting of S_1, S_2 and S_3) for which the factor graph is drawn

3.2.3. Factor Graph View

Vaske *et al.* translated the NEMs into Factor Graph model (Vaske et al., 2009). It is a way to express NEMs as a factor graph model, which itself can be understood as a generalization of a Bayesian Network. A factor graph is a bipartite graph that expresses which variables are arguments of which local functions, via factorization of the function (Kschischang et al., 2001). It is represented in terms of a hyper-graph where a factor node can connect more than

one normal node. The factor graph model for NEM introduces two kinds of factor nodes namely- (1) transitive factor (ω) for $S-gene$ interactions and (2) interaction factor (ρ) for $S-gene \leftrightarrow E-gene$ interaction (Figure3.7). The NEM likelihood for a S-gene graph can be now represented as equation 3.22

$$P(\mathcal{D}|\Phi) = \sum_{\mathcal{S}} \prod_{\gamma_{eA},\gamma_{eB}} P(\gamma_{eS_1},\gamma_{eS_2}|\Phi_{S_1,S_2},\theta_{eS_1,S_2})P(\mathcal{S}_{e,S_1}|\gamma_{eS_1})P(\mathcal{S}_{e,S_2}|\gamma_{eS_2}) \tag{3.22}$$

The prior for the S-gene network can therefore be given as:

$$P(\Phi) = \left(\prod_{\mathcal{S}} \omega_{\mathcal{S}}(\Phi_{S_1S_2},\Phi_{S_2S_3},\Phi_{S_2S_3})\right)\left(\prod_{S_1,S_2\in\mathcal{S}} \rho_{S_1,S_2}(\Phi_{S_1,S_2})\right) \tag{3.23}$$

Thus the value 0 for ω leads to a intransitive graph and 1 to transitive interactions.

3.3 NEM: Methodology Advancements

NEMs are designed to learn signaling cascades among perturbed genes via mapping of high-dimensional indirect effects (Markowetz and Spang, 2007; Markowetz, 2005). They estimate the interaction among unobservable variables (namely S-genes) from a set of observable downstream variables. Triplet method was introduced to infer network in the NEM framework (Markowetz, 2005) as we described in section 3.2.1. Nevertheless, computing cost for such inference method remains high. Markowetz *et al.* also introduced a model prior to penalize topologies in order to reach a more reasonable network.

Fröhlich *et al.* added further advancements to NEM in many ways (Fröhlich et al., 2008b). They also proposed the use of prior assumptions on the network structure via individual edge priors. This could bias the scoring of possible network hypotheses towards the biological realism. Furthermore, the module networks algorithm was introduced to infer networks of larger size (say more than 30 genes). The idea was to reconstruct a larger network from a set of smaller inferred graphs (Froehlich et al., 2007). Additionally their approach to use P-value profile via a beta-uniform mixture was introduced rather than a discretization step of the observed data, as originally proposed by Markowetz *et al.* (Markowetz, 2005), to take the account of effects.

An expectation maximization approach was proposed for faster detection of local maxima for posterior probability in NEM by Niederberger et al. (Niederberger et al., 2012) in form of MC-EMiNEM. This was presented a "counter-part for clustering in interventional data" (Niederberger et al., 2012).

Anchang et al. proposed D-NEM (Dynamic NEMs) with the introduction of time dimension in the NEMs to the modeling of perturbation time series measurements (Anchang et al., 2009). They decomposed observed time delays of multiple step signaling processes into single steps. This was further advanced by Fröhlich et al. by unrolling the signal flow over time in the dynoNEM approach (Fröhlich et al., 2011). It circumvented the need for time consuming Gibbs sampling of D-NEM, making the time series modeling via NEMs computationally more feasible.

Initially the scope on NEM model was limited to omic measurements. Failmezger et al. proposed an approach called MovieNEM (Failmezger et al., 2013). They used the phenotype data in form of time lapse imaging to infer cellular networks. This was one of the novel attempts to learn cellular network by observing phenotypic changes induced by targeted perturbation.

3.4 NEM: Example applications

Several attempts have been successfully made to utilize NEMs. Starting from the pioneering example of the immune response in *Drosophila melanogaster* (Markowetz, 2005), followed by transcriptional network in *Saccharomyces cerevisiae* (Markowetz and Spang, 2007). Fröhlich $etal.$ proposed its usability in learning the ER-α pathway in breast cancer cells (Froehlich et al., 2007; Fröhlich et al., 2008b), and Zeller modeled synthetic lethality interactions network in *S. cerevisiae* (Zeller et al., 2009) and the Rosetta data-set for *S. cerevisiae* (Vaske et al., 2009). We exemplify some of these applications here:

Studying immune response in *Drosophilla*

Markowetz *et al.*'s work on *D. melanogaster* is based on the measurements made by Boutros *et al.* where he used RNAi on selected genes to measure the effects on LPS (Lipopolysaccharides) induced genes via microarrays (Boutros et al., 2002). The data set consists of 16 Affymetrix microarrays: four replicates of control experiments without LPS and without RNAi (negative controls) four replicates of expression profiling after stimulation with LPS but without RNAi (positive controls) and two replicates each of expression profiling after applying LPS and silencing one of the four candidate genes tak, key, rel and $mkk4/hep$.

The inferred network shows a fork below tak, with key and rel on the one side ($tak \rightarrow key \leftrightarrow rel$) and mkk4/hep ($tak \rightarrow mkk4/hep$) on the other. Beside explaining this *'fork-topology'* idea of Boutros *et al.*, the inferred architecture also provided hints for the cross-talk between the two prongs of the fork (Figure 3.8). This can be attributed to the ability of NEMs to precisely model the upstream-downstream effects of perturbation.

Application on human ER-α pathway

Fröhlich *et al.* employed the module network approach (Fröhlich et al., 2008b) to meet the challenges to learn larger network in order to infer a 13 gene signaling involved in breast cancer. The network inferred consisted of 13 genes to be influenced by ER (Estrogen Receptor) status in breast cancer patients. These 13 genes was silenced via siRNAs, respectively, and the effect on gene expression was studied on whole genome cDNA microarrays for human MCF-7

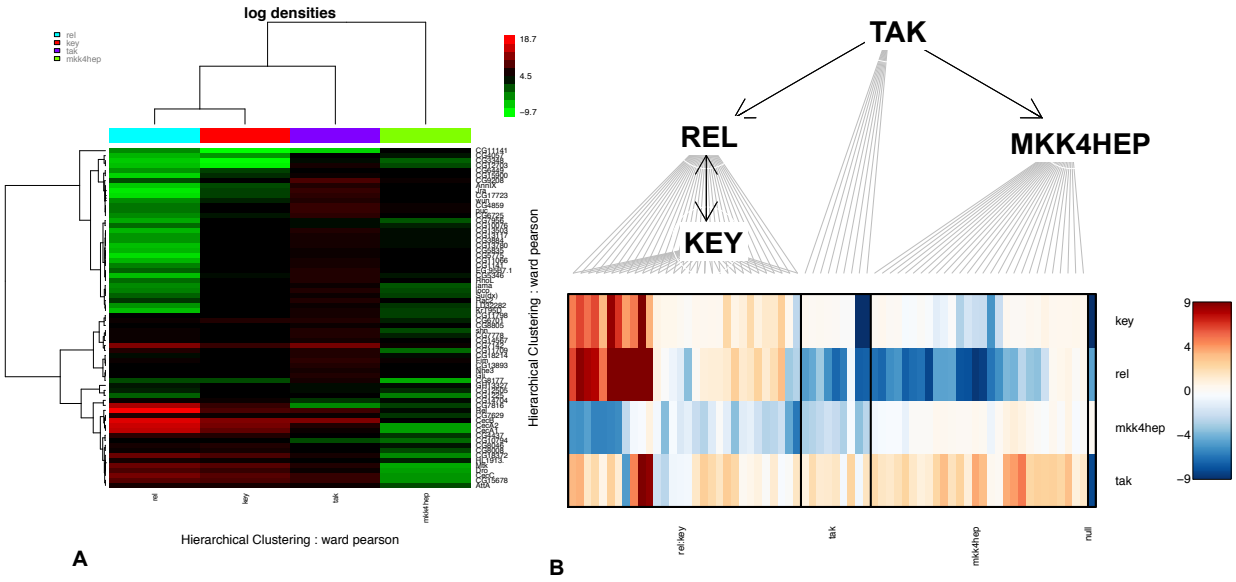

Figure 3.8.: Applying NEM on the Boutros et al. data. (A) The heatmap represent the density and (B) the inferred network with the E-gene map for perturbation

breast cancer cells. The induced edges were found to be robust via bootstrapping and most of them were also confirmed by the literature.

Studying mediator network in yeast *S.cerevisae*

Niederberger *et al.* applied NEM with the MS-EMiNEM algorithm (see section 3.3) to get new insights into the new insight into the signaling network involving the mediator subunits (Niederberger et al., 2012). A mediator is a multiprotein complex that functions as a transcriptional coactivator. They used mediator subunit deletion mutants in yeast. The subunits perturbed were dMed2, dMed15, dMed20, dMed31 together with published intervention studies on Med7, Med10, Med19, Med20 and Med21. The inferred result confirmed the mediator architecture, but also unveiled the organization of internal information flow in the mediator complex pointing out the central role of C terminus of Med7 and Med19, and the peripheral presence of the N-terminus of Med7.

3.5 Summary

Graphical models in general present a multivariate dependency structure among the random variables. They can only answer biological questions if they succeed in reliably and accurately reconstructing biologically relevant features of cellular networks. Unfortunately assessment and benchmarking protocols of inference methods are still sparsely explored. Although gene network inference algorithms are becoming accurate enough to be practically useful, especially if expression data are available for steady state, efforts must be directed in assessing algorithm performances. In recent years, network inference has become as common as clustering for microarray data analysis. The algorithms designed or to be designed for the purpose should

head towards being more 'integrative' by exploiting, protein-protein interaction data, sequence data, protein modification data, metabolic data *etc.* and more in addition to expression profiles, in the inference process (Workman et al., 2006). The mainstay of the thesis is to work in these directions, in order to include other factors like, time, using phenotype data for network inference. Furthermore, we also aim to use biological knowledge from multiple sources to improve accuracy of network inferred, as well as make the inference process aware of the well established interactions.

The chapter outlined the basic framework and an introduction to NEM(s) Nested Effects Model. The details of NEM(s) semantics and formalism were explained here. The next chapter will propose the use of time resolved data via an extension of NEM(s) developed during this thesis called dynoNEM (Dynamic Nested Effects Model(s)).

Chapter 4
dynoNEM: Dynamic Nested Effects Model(s)

This chapter presents an extension on the Nested Effects Models (NEMs). The novelty brought about by the work presented here is a fast and efficient algorithm that can learn a biological network from time point differentiated perturbation experiments. This can help to resolve feedback loops and distinguish between slow and fast interactions and hence differentiate direct edge from indirect edges in the cellular networks. The work has been published as a peer reviewed research article and as a highlight paper at the 18th International Conference on Intelligent Systems for Molecular Biology (ISMB 2011) (Appendix M). The work also extended the existing ***nem*** *R-package with the new* ***dynoNEM*** *software.*

4.1 Motivation

The last chapter concluded with a short introduction to Nested Effects Models (NEMs) showing how can one use data from perturbation experiments to understand the cellular machinery. The secondary effects of the gene perturbations were utilized to reconstruct features of the pathway between upstream genes. The NEM method has been extensively applied and extended to meet different biological goals on different data sets (Markowetz and Spang, 2003). Initially NEMs were proposed to infer the networks for signaling cascades/pathways. These pathways however are fast, eluding time as limiting factor in understanding these pathways (Alon, 2006, page:11). But as discussed in section 1.2, cellular system have different classes of network with different properties. To illustrate we consider a transcriptional network where we have a distinct separation of time scales (Alon, 2006, page:8-11). The time taken transcription factor binding is extremely short (in ms) while transcription and translation of genes can take minutes and change in protein concentration can take even longer. In such networks we observe the factor of time delays which may differ depending on the interactors. Considering these time delays while reverse engineering of networks can change the way one can perceive a biological network and make it biologically more rational.

The time delays in the system can be mapped for a system via temporal data. Such data is a sequence of data instances from measurements at successive moments. This data thus bears the information of the system that generated it and its behavior across time. This information provides valuable insight into the dynamic mechanisms underlying the biological processes being observed. Although, making inferences from such temporal data has statistical

challenges (Werhli et al., 2006), it does open up new perspectives. This enables us to observe not just the time intervals in which the interactions actually take place but can also lead to the understanding of direct and indirect edges together with feedback loops, discussed in the upcoming sections (Bolouri, 2008, page:154).

With the increasing availability of time-resolved data, it is desirable and helpful to integrate the temporal aspects of data into the reverse engineering work-flow. Dynamic Bayesian Network (DBN) (Friedman, 1998; Murphy and Mian, 1999; Bilmes, 2000; Ong et al., 2002; Husmeier, 2003) is one of the pioneering models to learn a network from temporal data. The differential equation systems based modeling of networks from such data was proposed by Nelander et al. (Nelander et al., 2008). State-space regression models were also proposed to address the issue (Rau et al., 2010). However NEMs can utilize high dimensional effects of perturbations as well as the nested structure of these effects.

A naïve approach to model temporal data is the use of static NEMs, which hereafter will be referred as simpleDNEMs (Fröhlich et al., 2011). It considers the time point measurements for perturbations being statistically independent. This makes the time point data equivalent to measurements replicates. The static NEMs can then be applied on the data to infer the network structure from it. However this approach faces the problems finding similar transitively closed graphs (Flesch and Lucas, 2007). This limits the distinction of direct and indirect edges in networks. Second, the simpleDNEMs does not explicitly take into account temporal information and hence its power in modeling time scale separated networks.

Anchang *et al.* proposed a statistical method- Dynamic Nested Effects Model (D-NEM) for the purpose (Anchang et al., 2009). The method is an extension of the NEM methods. D-NEM assumes the time delays for signal propagation steps to be exponentially distributed with rate constants as the major model parameters. The posterior distribution of the network is computed via Gibbs sampling (Casella and George, 1992). The model actually intends to infer the signaling time in the S-gene networks.

The D-NEM approach opens up the issue of temporal data modeling in NEM. It demonstrated the power of temporal information and its ability to understand the time scale separated networks with NEMs. The method distinguishes between the edges within a network in terms of inferred signaling time (Anchang et al., 2009). However, inferring upstream signaling time from downstream effects is biologically less reasonable and less realistic. Furthermore, the use of Gibbs sampling make D-NEM as computationally demanding algorithm in terms of speed (Casella and George, 1992). The forthcoming sections of this chapter will introduce a new approach to model temporal data with NEMs, which is computationally much faster and hence practically feasible.

4.2 dynoNEM

In NEMs the perturbation signal is assumed to propagate deterministically through the whole S-gene network Φ. Without time information, such networks cannot model feedback loops as cyclic pathways fall into the same equivalence class, namely a clique. Such inferences though frequently encountered, are explicitly not feasible with static data. However, a

time series measurements of perturbation effects can potentially overcome these limitations. dynoNEM is an approach to infer network from such data via NEMs. dynoNEMs (**Dyn**amic **N**ested **E**ffects **M**odels) aims to extend the existing NEMs to the modeling of perturbation time series measurements, adding the temporal dimension in network reverse engineering. The approach complements the D-NEM approach of Anchang *et al.*

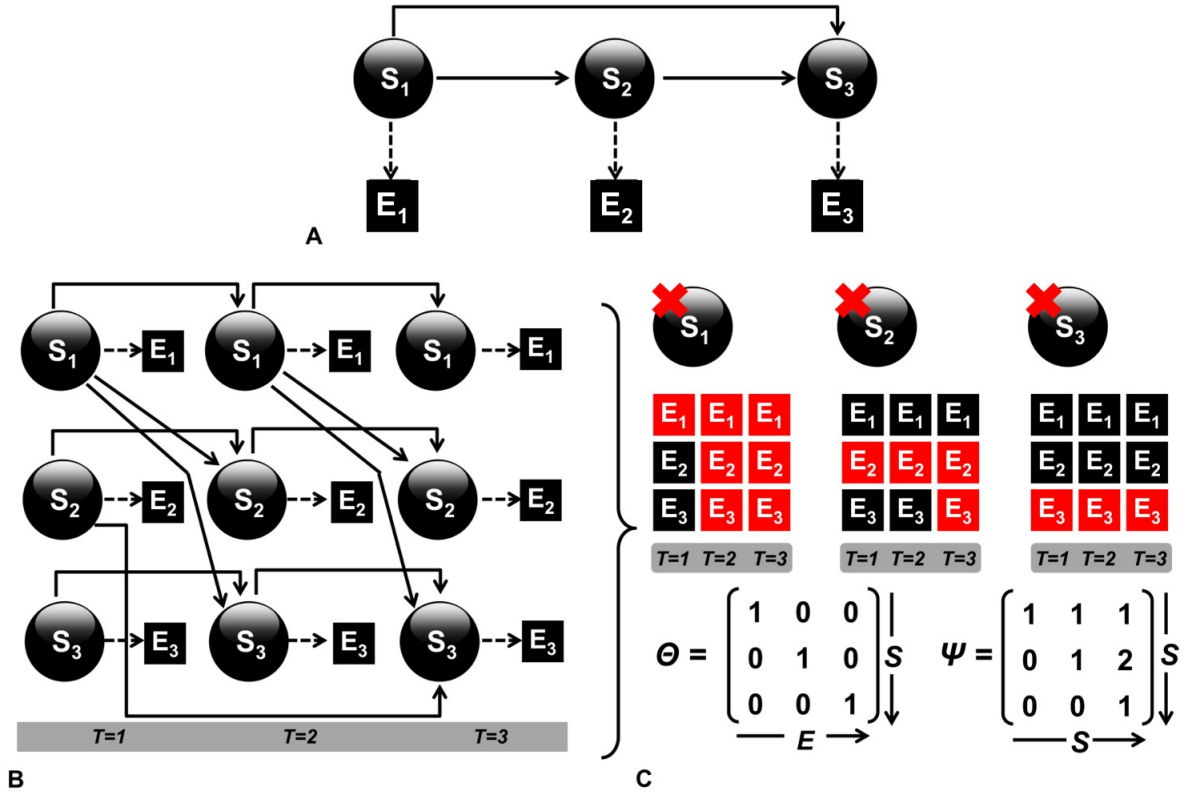

Figure 4.1.: The concept of dynoNEMs. (A) A static NEM with 3 S-genes is parametrized by a directed graph. (B) Network toplogy of dynamic NEMs representing the temporal data unrolled into three layers each representing a time point (C) The predicted effects for perturbation (in red) for same graph along a time (T) with Θ representing the attachment of each E-gene to corresponding S-genes ultimately yielding the weighted adjacency matrix Ψ for connections among S-genes.

4.2.1. Principle

The key idea behind dynoNEM is observing the data as slices along time and edge can occur between nodes across these time slices. This phenomenon can be referred to as unrolling of network along time. The unrolled network shows how a variable (node) affects variable in the next time step. Let us start with a static NEM for 3 genes S_1, S_2 and S_3 as in figure 4.1 (A). Time point data adds time dimensions to the model. The space for NEM is of two types the unobservable S-gene space and observable E-genes. At every time point we have the readings for the observables. Based on these readings a circuit among the unobserved states can be inferred across the time layers (Figure 4.1 (B)). This approach of unrolling the signal flow over

time allowing computation in a way similar to 'Dynamic Bayesian Networks' (Ghahramani, 1997). The effects predicted based o this unrolling shows that the perturbation effect of S_2 on S_3 and its direct descendant E-gene E_3 is reached after 2 time steps and hence $\Psi_{2\,3} = 2$.

Trivially in such unrolled network the states along consecutive layers of time are connected following 1^{st} order Markov assumption (Perrin et al., 2003). This means the state in current time depends on its immediate previous time state. However in biology this condition does not hold necessarily. This is because of time lags. Therefore, dynoNEM considers the higher order Markov assumption distinguishing itself from usual Dynamic Bayesian Networks. In contrast to DBNs in dynoNEMs each edge has a defined time delay. Therefore, in the unrolled network there can appear edges from $t-k$ to $t(k > 1)$. This enable the dynoNEM to allow edges between nodes skipping more than one time point. Furthermore, a classical NEM approach to model in case of temporal data faces problem while operating on similar transitive closures because static NEMs can principally only resolve networks up to these equivalence classes.

Now in context to NEMs, a network among S-genes is defined as an adjacency matrix where an edge is supposed to exist if the adjacency matrix value is 1(i.e. $\Phi_{i,j} = 1$). In case of time series measurement the adjacency matrix Ψ can take on values different from 1, because it represents an edge weighted graph. The following section explains the mathematical theory involved in dynoNEMs.

4.2.2. Mathematical formalism

Let $\mathcal{D}$ be the data to be modeled. For the scenario here the data is a time series data, thus $\mathcal{D} = \{\mathcal{D}_{i,j} | t \in \tau\}$, where τ is the set of time points (τ={1, ..., T}). The measurements in $\mathcal{D}$ can be p-values or some other quantification of effect of S-gene knockdown on the respective E-genes. The data instance $\mathcal{D}_{i,j}(t)$ is the measure of E-gene j on perturbing S-gene i at time point index t (not the time point itself). The static adjacency Φ is converted to a $\mathcal{S} \times \mathcal{S}$ weighted adjacency matrix $\Psi = \Psi_{ij}$, where $\Psi_{ij} = 0$ means no edge and a value $\Psi_{ij} > 0$ implies an influence of node i on E-genes downstream of node j delayed by Ψ_{ij} time steps

As described in section4.2.1 dynoNEM involves an unrolled network along the time series. To achieve this unrolling each S-gene s is clubbed with a Boolean status at every time point t. This describes the perturbation state of the gene at that time point. A perturbed gene s at time point t acquire a state 0 Otherwise (i.e. without any perturbation) the state of S-gene s is 1. A knock-down at any time t thus switches the gene $1 \rightarrow 0$. The operating S-genes circuit that dynoNEM aims to infer, can change the status for the gene. This switching will depend on the location of genes (upstream/ downstream) and the time lag elapsed for a signal to reach from one gene to other. To build upon, the NEM likelihood is give as equation 4.1 which marginalizes as equation 4.2 (for details on NEM formulation please see 3.2.1)

$$P(\mathcal{D}|\Phi, \Theta) = \prod_{i \in \varepsilon} \prod_{k \in \kappa} P(\mathcal{D}_{i\,k}|\Phi, \Theta) \tag{4.1}$$

$$P(\mathcal{D}|\Phi) = \prod_{k\in E}\sum_{j\in\mathcal{S}}\prod_{i\in\mathcal{S}} P(\mathcal{D}_i|\Phi,\Theta_{i=j})P(\Theta_{i=j}|\Phi) \tag{4.2}$$

For the time point observation we have our data $\mathcal{D}$ along three dimensions namely, time (τ), S-genes $\mathcal{S}$ and E-genes E yielding a $|\tau| \times |\mathcal{S}| \times |E|$ matrix. Please recall that in NEMs the data matrix was of dimension $|\mathcal{S}| \times |E|$. Thus, $\mathcal{D}_{ik}(t)$ shows the effect i on pertubation signal k at time point t, (where $i \in E$, $k \in \mathcal{K}$ and $t \in \tau$). Ψ and Θ refer to the matrix representing the weighted S-gene connections and S-gene to E-gene connections respectively. On unrolling the S-gene network along time $t = 1 \rightarrow T \in \tau$, i.e. in case of dynoNEMs the likelihood of data given a network is represented as equation 4.3.

$$P(\mathcal{D}|\Psi) = \prod_{i\in E}\sum_{s\in\kappa}\prod_{k\in\mathcal{K}}\prod_{t=1}^{T} P(\mathcal{D}_{ik}(t)|\Psi,\Theta_{is}=1)Pr(\Theta_{is}=1) \tag{4.3}$$

This extends the NEM (equation 4.1) to time dimensions, in two ways- first introducing the status of each S-gene at a given time point and second, the consideration of S-gene status in each perturbation experiment k.

Since the status of each S-gene depends on the parents and time layer, first part of equation 4.3 can be given as

$$P(\mathcal{D}_{ik}(t)|\Psi,\Theta_{is}=1)$$

$$= \sum_{s(t)\in\{0,1\}} \Big(P(\mathcal{D}_{i\,k}(t)|\mathcal{S}(t)=s(t),\Theta_{i\,s}=1)\cdot Pr(\mathcal{S}(t)=s(t)|Pa(s)(t))\Big) \tag{4.4}$$

$\mathcal{S}(t)$ represents the random variable for unobservable state of signal s at time t. The second part of the equation 4.4 can be further represented as equation 4.5 based on the unrolling of signal flow.

$$Pr(s(t)=0|Pa(s(t))=[r]) = \begin{cases} 1 & \exists p \in Pa(s(t)) : [p]=1 \\ 0 & otherwise \end{cases}$$

$$Pr(s(t)=1|Pa(s(t))=[r]) = 1 - Pr(s(t)=0|pa(s(t)=[r]) \tag{4.5}$$

This means that a gene s is perturbed at a time t given any of its parents are perturbed at previous time step $t-1$. The local likelihoods form the first part in equation 4.4) that defines the likelihood of observing an effect of perturbation s of E-gene i at a time instance t. It follows a BUM model defined by Pounds *et al.* (Pounds and Morris, 2003) (for details please see section 3.2.1). Thus, this local likelihood follows the formalism as described below.

$$P(\mathcal{D}_{i\,k}(t)|\mathcal{S}(t)=s(t),\Theta_{i\,s}=1)=\begin{cases} f_1(\mathcal{D}_{ik}(t)) & [s(t)]=0 \\ 1 & [s(t)]=1 \end{cases} \tag{4.6}$$

Here the density function f_1 defines the effect of perturbation on E-gene i ($i \in E$) under alternative hypothesis of expecting an effect, which can be modeled with a three component BUM model (Fröhlich et al. (2008b), please see section 3.2.1).

simpleDNEM

simpleDNEMs are based on the assumption of statistical independence of time points. With this assumption the measurement of observable get statistically independent at every time point. Thus we have :

$$P(\mathcal{D}_{ik}(t)|\Psi,s(t)=[s(t)],\Theta_{is}=1)=P(\mathcal{D}_{ik}(t)|\Psi,\Theta_{is}=1) \tag{4.7}$$

The RHS can be denoted as equation 4.8.

$$P(\mathcal{D}_{i,k}(t)|\Psi,\Theta_{is}=1)\equiv\prod_{t=1}^{T}P(\mathcal{D}_{ik}(t)\}|\Psi,\Theta_{is}=1) \tag{4.8}$$

This can be further expanded with classical NEM explained last chapter. However, the simpleDNEM approach cannot model the time propagation effect because of the independence assumption. The term 'static NEM' will be synonymously used in the text for simpleDNEMs.

4.2.3. Prior

In the above section, we introduced and defined Ψ, the weighted adjacency matrix as a compact representation of the raw upstream network structure among S-genes and time delays between perturbation events and downstream responses. Learning an upstream signaling cascade now amounts to learning matrix Ψ based on the likelihood in equation 4.3. While scoring candidate network structures Ψ we assume higher edge weights (i.e. observing an effect after longer time lag) to be less likely than small edge weights. Moreover, in general, edges $x \to y$ are redundant, if their time lag is larger or equal than the length of some other path from x to y, where the path length is defined here as the sum of the path's time lags. We have to ensure that such redundant edges, i.e. edges which are principally not observable do not appear. In other words, addition of these edges to Ψ does not change the likelihood of our model.

We specify our demands in form of a prior P(Ψ). Additionally to punishing higher edge weights, P(Ψ) should enable us to include prior knowledge into the network scoring, i.e. to bias the scoring toward known interactions. For this purpose, we assume to have a given matrix of prior probabilities for each edge. Fröhlich et al. (2007) proposed a prior for Ψ. This prior enables the inclusion of prior knowledge in the inference framework by affecting the scoring

systems and will be exploited further in chapter 6 and 7 . The matrix $\hat{\Psi}$ is the prior probabilities for the edges in network matrix Ψ. The prior defined by Fröhlich is given as follows.

$$P(\Psi|\nu) = \prod_{i,j} \frac{1}{2\nu} exp\left(\frac{-|\Psi_{i,j} - \hat{\Psi}_{i,j}|}{\nu}\right) \tag{4.9}$$

where $\nu > 0$ is an adjustable scaling parameter. The parameter ν can be chosen according to the BIC criterion:

$$BIC = -2logP(\mathcal{D}|\Psi) + log(|E|) \sum_{ij} 1\{|\Psi_{i,j} - \hat{\Psi}_{i,j}| > 0\} \tag{4.10}$$

where, $\sum_{ij} 1\{|\Psi_{i,j} - \hat{\Psi}_{i,j}| > 0\}$ is an estimate for the number of parameters in the model.

In the default case (which is employed in our subsequent studies throughout this chapter), we suppose $\hat{\Psi}$ to be an empty graph. This way sparse network structures are favored, inducing a sparsity prior.

4.2.4. Structure learning

As an exhaustive search based approach to infer structure is practically and computationally unfeasible, we need look forward to heuristic methods. Two learning methods were used to learn the network structure in dynoNEMs.

Greedy hill climber

The greedy hill-climbing (GHC) algorithm in search space takes an initial graph, defines a neighborhood via transformation/mutation operation, computes a score for every graph in this neighborhood, and chooses the one which maximizes the score for the next iteration, until the scoring function between two consecutive iterations does not improve. Thus, it accepts a candidate solution if it is better than the current one. Nevertheless, it cannot find the global maxima unless the heuristic landscape is convex. The algorithm gets stuck in a state if it has a better score than any of its children, though the subsequent states of these children may offer a better evaluation or the optimal state (maximum score for objective function i.e the network likelihood) in the entire heuristic landscape (Chickering et al., 1995). Therefore, GHC does not guarantee to find a globally optimal solution.

In case of dynoNEMs, network learning aims to find an optimal weighted adjacency matrix Ψ. The matrix entry consists $\Psi_{i,j}$ as the time delay for interaction between gene i and j such that $\Psi_{i,j} \in (0, \ldots, T)$. The GHC starts with an initial network Ψ and corresponding score S. At every search step all operations (edge weight increase, decrease, edge reversal) are tested for all edges permitting these respective operations in the network (Ψ) to get new matrix (Ψ')(Equation 4.11). For this purpose, we define three search operators (Equation 4.11). The solution that increases the posterior likelihood most, is accepted. The search stops when

the objective function i.e the network likelihood of existing network is better that all the neighboring networks (obtained via above mentioned transformations).

1. edge weight increase
2. edge weight decrease
3. edge reversal (swapping the edge weights)

$$\Psi'_{i,j} = \begin{cases} \Psi_{i,j} + 1 & if \quad \Psi_{i,j} < \tau \\ \Psi_{i,j} - 1 & if \quad \Psi_{i,j} > 0 \\ \Psi_{j,i} & if \quad \Psi_{i,j} \neq \Psi_{j,i} \end{cases} \tag{4.11}$$

The score S' for this new network is computed in terms of likelihood for each operation. The best score $\mathcal{L}'$ and corresponding network is accepted as new score and so the matrix Ψ'. Hence, at each search step, we accept the solution increasing the posterior likelihood most. The final network is achieved when we do not observe any further improvement in the score. This we refer to as Ψ_{best} and the corresponding score $\mathcal{L}_{best}$

The GHC algorithm here requires $O(|\mathcal{S}|^2)$ likelihood evaluations per search step, where each likelihood computation itself has a time complexity of $O(|\tau||E||\mathcal{S}|^2)$. Hence, each search step requires $O(|\tau||E||\mathcal{S}|^4)$ time.

Markov Chain Monte Carlo

Though, greedy hill climber (GHC) algorithms are simple and fast, they have certain limitations (Russell et al., 1996). The GHC can fail to reach the globally optimal network structure as it can get stuck in local maxima (Chickering et al., 1995). The number of possible structures is super-exponential depending on the number of vertices in the graph. This raises the issue of identifiability in structure learning, i.e. true network may not at all or at least not uniquely determinable from limited data. An approximation of the true network is possible based on a subset of possible structures (Friedman and Koller, 2003). Therefore, sampling based search has been advocated to infer dependencies in graph from data in case of Bayesian networks (Daly et al., 2011; Grzegorczyk and Husmeier, 2008). Therefore the dynoNEM was further aided with a Markov Chain Monte Carlo based (MCMC) sampling (W R et al., 1995). A MCMC is implemented here using the Metropolis Hastings algorithm (Robert and Casella, 2005).

MCMC is a class of sampling algorithms. It generates samples x^i while exploring the state space X using Markov chain mechanism i.e. the next state depends of only on current state (order 1)(Andrieu et al., 2003). The purpose of MCMC is to sample from a (possibly high dimensional) target distribution $p(x)$. The working principle of Markov chain Monte Carlo methods can be explained as moving from initial distribution to target distribution via a Markov kernel K (Robert and Casella, 2009). This kernel equals a distribution proposing state x^{t+1} when being in state x^t. For network learning the target distribution is the posterior $p(\Psi|D)$. To sample from this distribution dynoNEM uses Metropolis-Hastings algorithm.

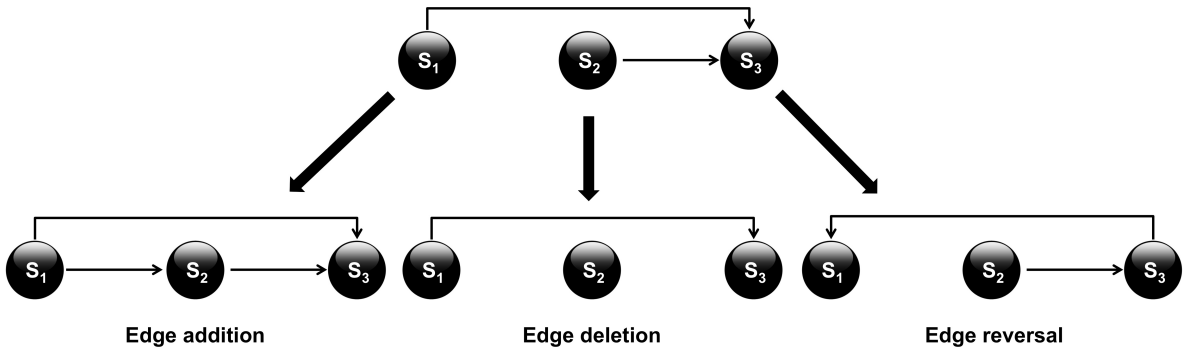

Figure 4.2.: Moves to search the neighborhood of a network with initial network at the top

Metropolis Hastings algorithm is a general approach for producing a correlated sequence of draws from a target distribution. The algorithm start with a random network and is altered by picking an edge and one of the operations in equation 4.11 at random with equal probability (Figure 4.2). The likelihood is computed for each of these network with the dynoNEM likelihood function (Equation 4.3).

The MCMC moves are accepted with a probability equivalent to the Hastings ratio (Equation 4.12) .

$$\alpha = min\left(1, \frac{P(D|\Psi^{new})P(\Psi^{new}|\nu^{new})P(\nu^{new})N(\Psi^{old})}{P(D|\Psi^{old})P(\Psi^{old}|\nu^{old})P(\nu^{old})N(\Psi^{new})}\right) \tag{4.12}$$

Where, $N(\cdot)$ is the number of reachable networks within one MCMC moves from the current network. The convergence for MCMC (Figure 4.3)can be reached quickly via an initialization with GHC (see section above) (Fröhlich et al., 2011). Once the convergence is reached (after burnin) the algorithm reaches the target distribution and sampling begins. To avoid the sampling of correlated networks a thinning with a factor 100 is used. This adds only every 100^{th} network to the sample. Finally an edge wise weighted network is returned.

4.3 Results

To evaluate the performance of the algorithms, extensive set of simulation were performed to study the effect of various parameters on the methods. The algorithms were compared against simpleDNEMs as described in previous sections. Afterwards, an application to learn a transcriptional network playing a role in murine stem cell development is shown (based on data from Ivanova *et al.* (Ivanova et al., 2006)).

4.3.1. Simulations

Network sampling

To assess the reconstruction performance of dynoNEM, simulations were designed with networks extracted from KEGG signaling pathways as base graphs. To generate a test graph

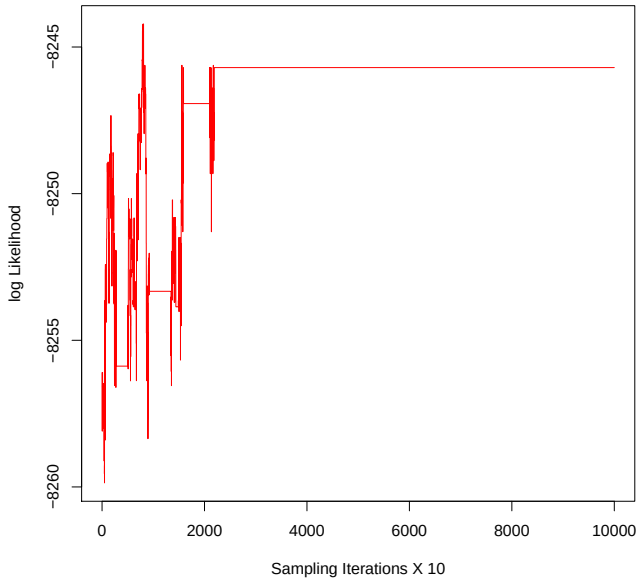

Figure 4.3.: Convergence plot for MCMC based dynoNEM showing the log likelihoods across sampling iterations.

first a base KEGG signaling pathway was selected randomly. From the selected pathway a core starting node was picked at random. From this starting node a random walk was performed in the graph, visiting each neighboring node with equal probability. The walk continued till the algorithm had been visiting a prescribed number of nodes. Revisiting same nodes was avoided. The sub-graph for the sampled node was extracted from the base KEGG graph. This sub-graph served as the S-gene network. Further, to attach E-genes to each S-gene a uniform random model was used. The time lags (time taken by signal to reach from gene S_1 to S_2) onto the S-gene networks were had been visiting a prescribed number of nodes from a geometric distribution with parameter p. While doing this, we ensured that the time lag for indirect edge between nodes S_1 and S_2 was smaller than the sum of all other paths between S_1 and S_2, thus avoiding edge redundancy. Furthermore the graph sampling ensured that at least one of the sampled network was cyclic. This way of sampling allows to test the algorithm performance on topologically biological graphs.

Data sampling

Following the graph sampled above corresponding perturbation data has to be generated. This was achieved by simulated knockdown of each of the S-genes in the sampled networks. The knock down simulation for the data was generated with the consideration of the time lags. To illustrate, when a gene say S_1 is perturbed then the gene S_2 downstream of S_1 was assumed to be perturbed after the time lags involved between these genes i.e. when the signal reaches S_2 from S_1. This created the status of S-genes (perturbed/unperturbed) at every time point. Depending on this status the P-values for corresponding E-genes were simulated. The P-values

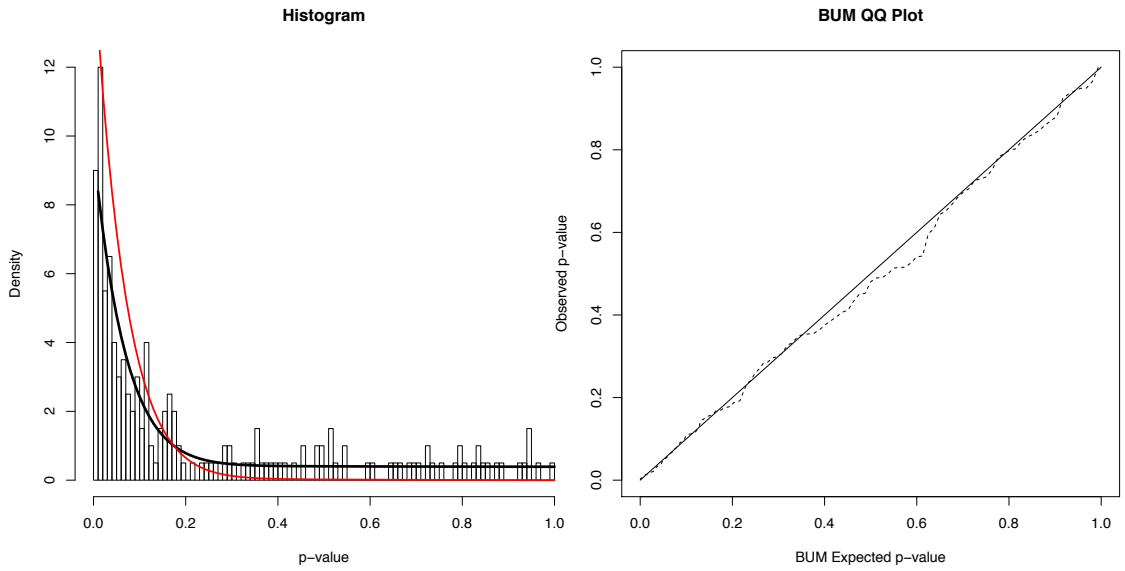

Figure 4.4.: BUM model fit for generated data (m=5, n=200). The plot on the left shows the histogram and on right is the corresponding QQ plot

followed a defined distribution, called f_1 if their corresponding S-genes were perturbed at the given time, otherwise, followed a uniform distribution (with a single control variable). Please see section 3.2.1 for details.

The parameters for the f_1 distribution in the BUM model were sampled randomly using a 3 component mixture of a uniform, and two $Beta$ distributions with $Beta(1, \beta_i)$ & $Beta(\alpha_i, 1)$ as proposed by Fröhlich *et al.* (Fröhlich et al., 2009) (Equation 4.6).

$$\mathcal{D}_{i,k}(t) \sim \gamma + (1-\gamma) f_1(\mathcal{D}_{i,k}(t)), \gamma \in (0,1) \tag{4.13}$$

This helped to model the p-values for E-genes in a realistic fashion in unlike the in Markowetz (2005), where the association are considered to be constrained in binary terms. After wards the local effect likelihoods were estimated using again the f_1 distribution, but this time with slightly different parameters in order to simulate the estimation error of distribution parameters occurring in practice

The data generation was repeated 100 times for each network and parameter configuration.

Network Reconstruction

The algorithms were used to reconstruct the sampled networks from the data generated and their performance was compared against the simpleDNEM as described in section 4.1. The performance were analyzed in terms of sensitivity (tpr), specificity ($1 - fpr$) and balanced accuracy (BAC).

$$tpr = \frac{100 \times TP}{TP + FN} \tag{4.14}$$

$$fpr = \frac{100 \times FP}{FP + TN} \tag{4.15}$$

$$BAC = \frac{tpr + (1 - fpr)}{2} \tag{4.16}$$

The reconstructed networks were then analyzed for various dependencies as follows.

A. Dependency on number of S-genes

First factor under consideration for the analysis was the effect of number of S-genes on the reconstruction performance. The sampled networks varied in terms of number of nodes ($m = \{3, 5, 10, 15$ and $20\}$). The other parameters (number of E-genes, time points and parameters) were unvaried for this analysis (T=5, p=0.8 and E=200). The reconstructed networks via GHC and MCMC algorithms (called dynoNEM.HC and dynoNEM.MCMC) were compared against simpleDNEM approach. It revealed a significant improvement in terms of sensitivity at equally high specificity compared to simpleDNEM (Figure 4.5).

Furthermore, increasing the number of S-genes increased the margin of improvement brought about by dynoNEMs (GHC and MCMC). However this increase came at the cost of reduced sensitivity. To illustrate; for smaller networks ($m = 3$) the performance margin between GHC and MCMC was marginal. This margin got bigger improved with the increase in network size but their absolute performance reduced. Nevertheless, both methods kept on surpassing the simpleDNEM across all sizes. In terms of specificity the algorithms (including simpleDNEM) performed well (close to 90%). A Wilcoxon pairwise test was performed to observe the significance of improvement. The MCMC method was found to be significantly better than the simpleDNEM (p-value=0.003) but GHC was not significant (p-value=0.09) at the sensitivity level. There was no significant difference in terms of specificity (see Appendix B).

B. Dependency on number of E-genes

In the second simulation the effect of number of E-genes was investigated. The performance for a varying number of E-genes ($|E| =$ {25, 100, 200 and 500}) for networks with 5, 10 and 15 S-genes. The data was simulated for a time series of length T = 10 with time lags sampled from a geometric distribution (p = 0.8) were simulated. In all cases, the dynoNEM (MCMC and GHC) led to a consistently better sensitivity than the simple DNEMs and high specificity. The observed gain in sensitivity was up to 15-20% for smaller networks (n = 5) with even with low number of E-genes (E = 25). However, increasing the number of E-genes reduced the difference between various algorithms under comparison. Larger networks (n = (10,

15)) continued to be difficult to learn, but using higher number of E-genes showed improvement in reconstruction compared to the use of fewer E-genes. The simulation demonstrated that even with a low number of E-genes, dynoNEMs reach a rather good reconstruction sensitivity (Figure 4.5). For instance, for n = 5 it was ovserved that just 25 E-genes are sufficient to get a sensitivity >90%.

In terms of specificity for smaller networks (n = 5) it was constantly $> 95\%$ in close run with the simple DNEMs. However, with increasing network size, i.e. larger n, the sensitivity drops slightly to 80% (n = 10, 100 E-genes) and 70% (n = 15, 300 E-genes) ; lower than simple DNEMs. The specificity nevertheless improves with increase in number of E-genes. The balanced accuracy was also found to be better than the simple DNEMs. In all the terms the MCMC approach outperformed the GHC approach. A pairwise Wilcoxon rank test with FDR correction revealed the significance of performance values of different methods (see Appendix B).

C. Dependency on length of time series

The length of time series in the experimental data can be instrumental in inferring graphs especially in case of cycles or indirect edges. The affect of number of length of time series was investigated in the third simulation for networks with n = 5 S-genes, m = 200 E-genes and p fixed to 0.5 (Figure 4.6). As expected, an increase of the number of time points lead to a strong improved sensitivity for both, dynoNEMs and static NEMs. The average difference in sensitivity between both dynoNEM methods and simpleDNEMs remained between 15 to 20%. The dynoNEMs achieved a very high specificity close to 100%, even with a low number of time points (especially for n=5 and 10), whereas static NEMs showed a slight decrease of specificity with an increasing number of time points. The difference in sensitivities is widened for larger number of S-genes (Figure 4.6).

A longer time series thus is needed for simple DNEMs to have better sensitivity, though it could not overrun dynoNEMs for larger networks. Thus the dynoNEMs were much more accurate in network reconstruction (see BAC plots in figure 4.6). The significance of improvement in accuracy is available in Appendix B

D. Dependency on time lag distribution

The time lag between interaction by dynoNEM is modeled as a geometric distribution controlled by parameter p (See Section 4.3.1). A high value for p results in sharper decline of probability values over time lags and vice versa. To examine the effect of these time lags the parameter p was varied across with values (0.2, 0.5 and 0.8). From the simulation it could be observed that the sensitivity of dynoNEMs as well as simple DNEM improves with increasing p, though performance of dynoNEMs remained superlative. The specificity however was more comparable i.e. in the same range.

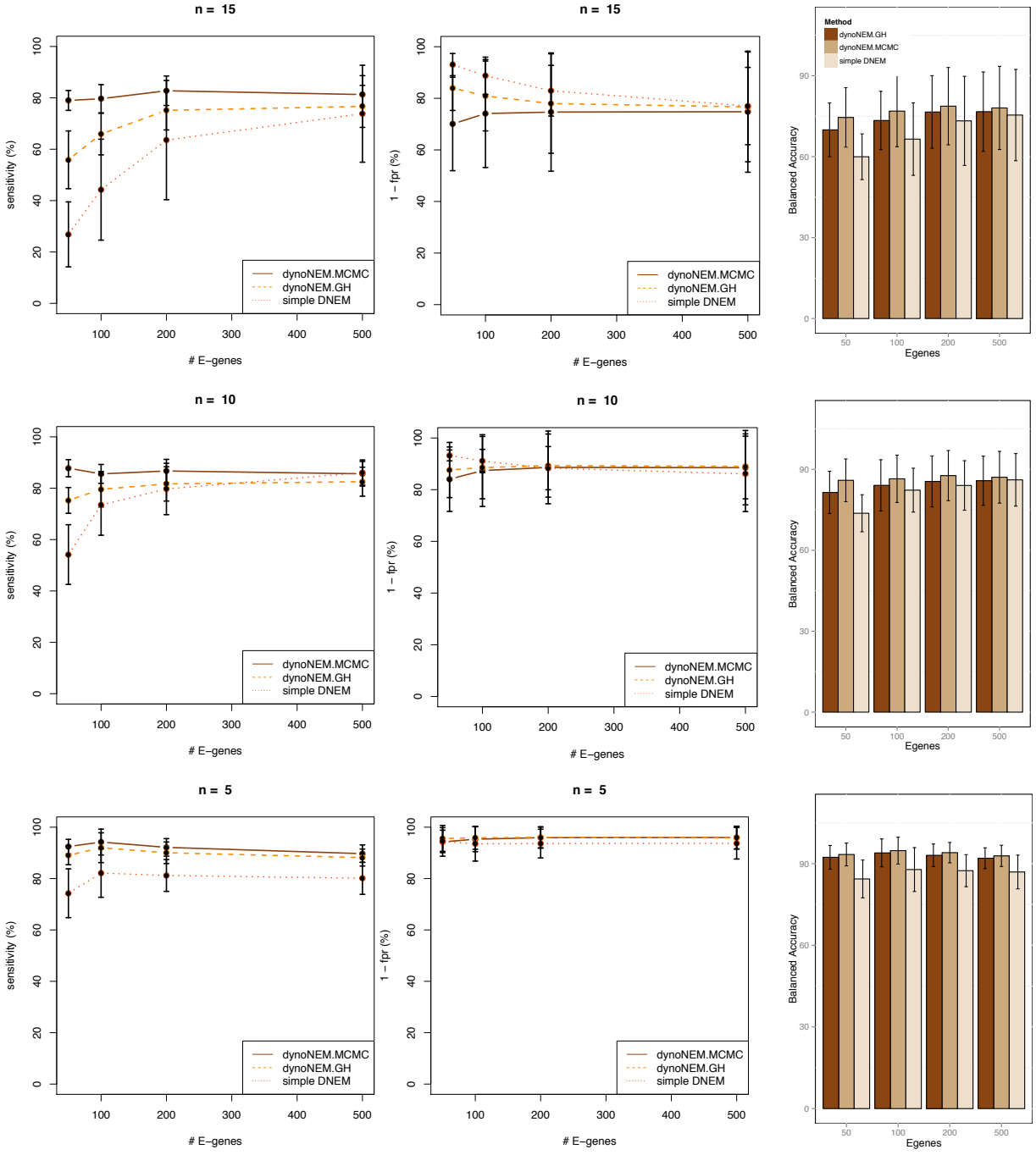

Figure 4.5.: Dependencies of network reconstruction via dynoNEMs on the number of E-genes. The plot shown here is for different number of S-genes (n=5, 10 and 15; from bottom to top). On the extreme left is the Balanced accuracy plot (BAC).

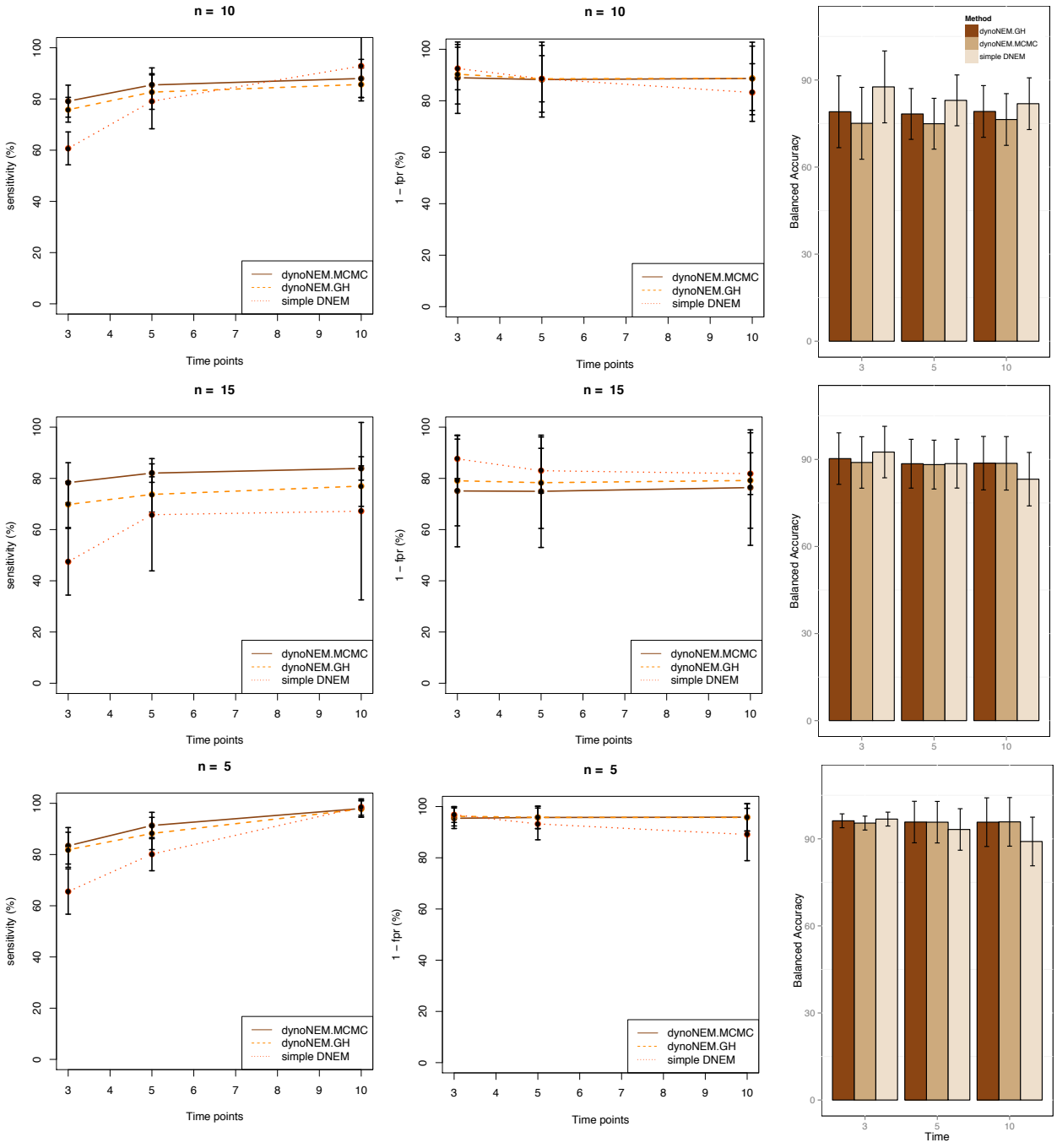

Figure 4.6.: Dependencies of network reconstruction via dynoNEMs on the length of time series in terms on number of time points. The plot shown here is for different number of S-genes (n=5, 10 and 15; from bottom to top). On the extreme left is the Balanced accuracy plot (BAC).

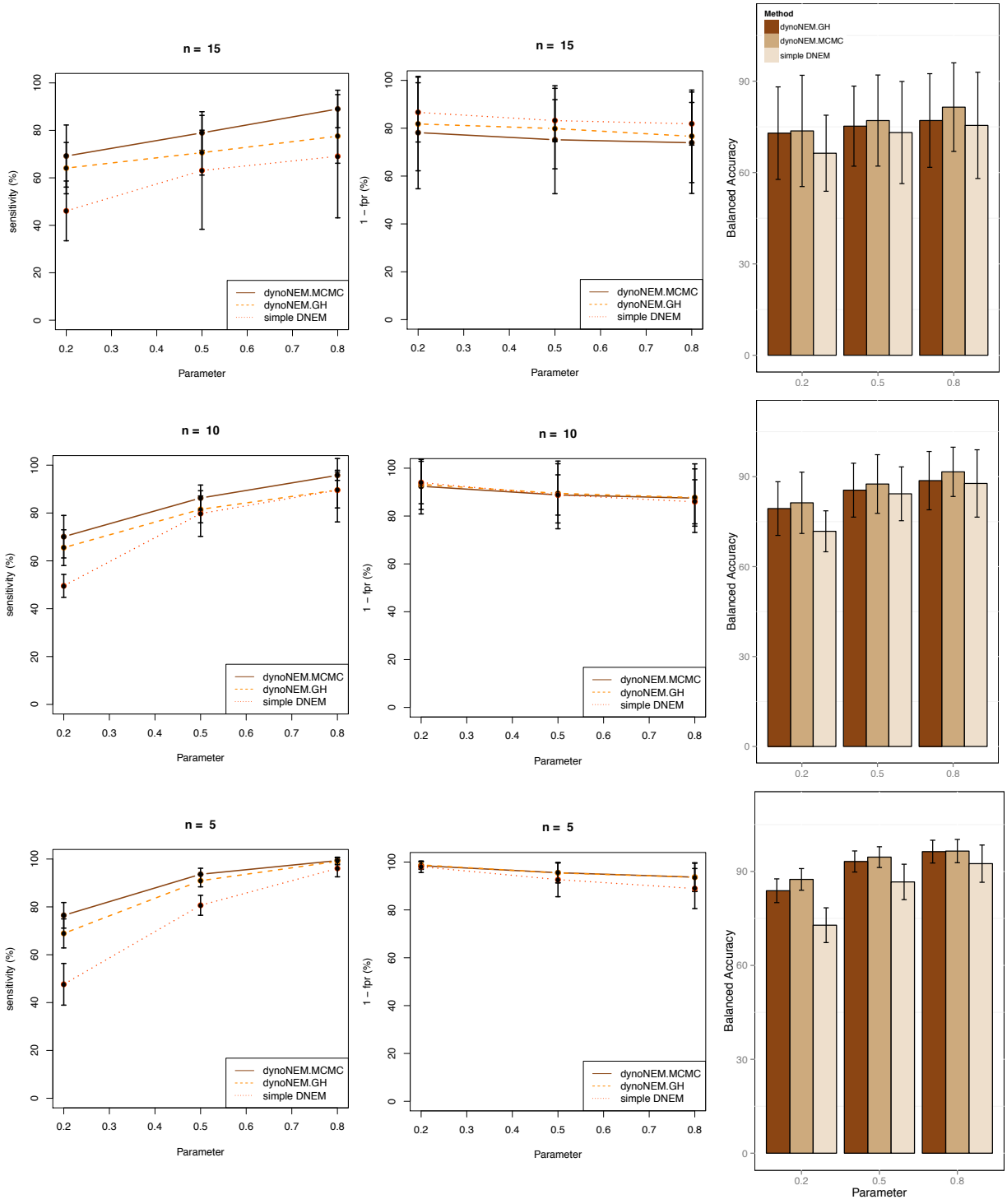

Figure 4.7.: Dependencies of network reconstruction via dynoNEMs on parameter for geometric distribution of time lags in simulated data. The plot shown here is for different number of S-genes (n=5, 10 and 15; from bottom to top). On the extreme left is the balanced accuracy plot (BAC).

Since in this simulation T was (intentionally) much larger than the maximum time lag of 4, the result showing an almost constant behavior of sensitivity, specificity over the whole parameter range was somewhat expected. It demonstrated that provided we have long enough time series the reconstruction quality is independent of the shape of the time lag distribution. Besides that, it is worth mentioning that our dynoNEMs achieved a sensitivity $>$90% with a specificity of almost 100%, whereas simpleDNEMs had much lower average sensitivity of $\sim$80% and specificity of $\sim$90%. Variation of the number of network nodes did not reveal a large impact on this general finding.

E. Dependency on network architecture

The next question to be answered is how does the nature of network topology affect the reconstruction performance ? What happens if the graph is heavily cycled or is having a number of indirect edge? For this we looked at individual sampled networks. For every simulation the sampled graphs were looked for the presence of cycles (please note that at least one of the 10 sampled graphs in every simulation were cyclic) and their corresponding reconstruction accuracies were mapped. For a fair comparison we used the simulation where the variables like- length of time series (=5), number of E-genes (=100) and time lags (p=0.5) were same. The figure presents 4 selected networks with 15 S-genes. Here the networks N2 and N3 was the hardest to learn compared to N1 and N4(Figure 4.8(a and b)). It was observed that on an average, dynoNEMs were better in reconstruction of these networks as compared to simple DNEMs. However, reconstruction performance for these networks was slightly worse than the average as anticipated. This is because getting probabilistic dependency structure in case of cycles in a network is indeterminable.

With an aim to investigate if there were systematic differences in the reconstruction accuracy of networks with indirect edges is affected by the time series length, we designed a simulation where we extracted networks topology (n=10) in terms of indirect edges and simulated data for it with 100 E-genes, time lags sampled with P = 0.5 and varying parameter T the results revealed that for longer time series the reconstruction slightly improved whereas for T=3 the results were highly restricted by the architecture (Figure 4.8(c and d)). Nonetheless, a detailed investigation of the frequency with which individual edges were inferred clearly revealed that dynoNEMs were in general able to identify feed-forward and feed-back loops correctly with high accuracy, even from short time series (see section 4.3.2).

4.3.2. Reconstructing network motifs

Finally we also evaluated the ability of dynoNEM to reconstruct some fundamental network motifs (Figure 4.9 (A-D)) The data for these reconstructions was simulated in the way as described in section 4.3.1. The time points were kept to 5 and parameter p to 0.5 and 100 E-genes were used for simulations. The investigations revealed the improvement in reconstructing these motifs specially sensitivity with comparable specificity (Figure 4.9 (E-F)). This is obvious as time differentiated data can map the dependencies and hence the cyclicity in network .

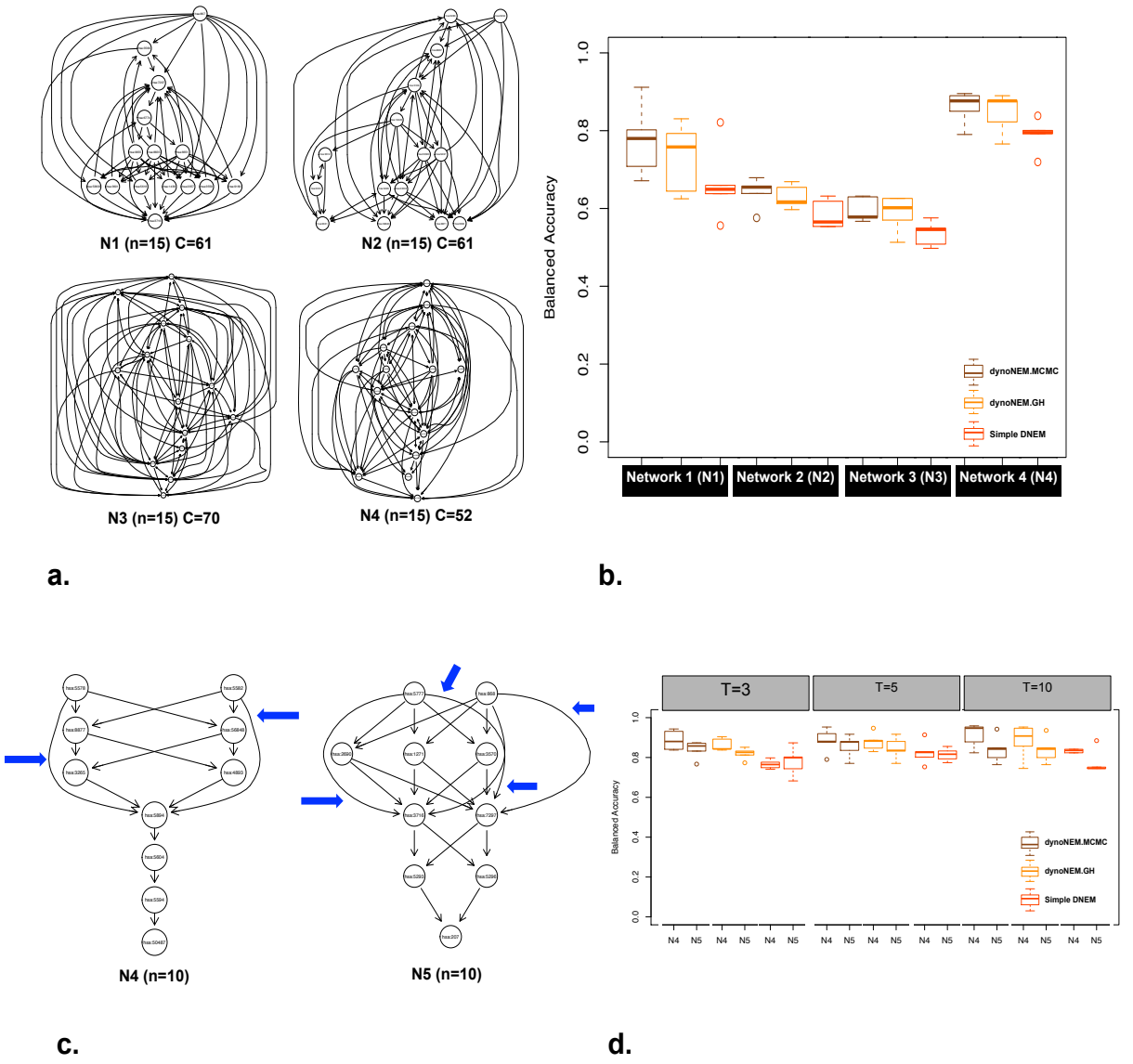

Figure 4.8.: Reconstruction accuracy of networks based on the architecture (topology) of the networks. On the top we have (b) 4 example cyclic networks (number of 1st order cycles (C) given) and (b) the balanced accuracy of reconstruction with three different methods. On the bottom is the (c) two example networks with direct and (indicated with blue arrows) indirect edges and (d) their corresponding balanced accuracy with the three approaches.

4.3.3. Convergence of the Markov chain

A critical issue for users of Markov Chain Monte Carlo (MCMC) methods in applications is how to determine when it is safe to stop sampling and use the samples to estimate characteristics of the distribution of interest. The phenomenon by virtue of which an MCMC sampler reaches the target distribution is termed as the convergence of the sampler. The convergence

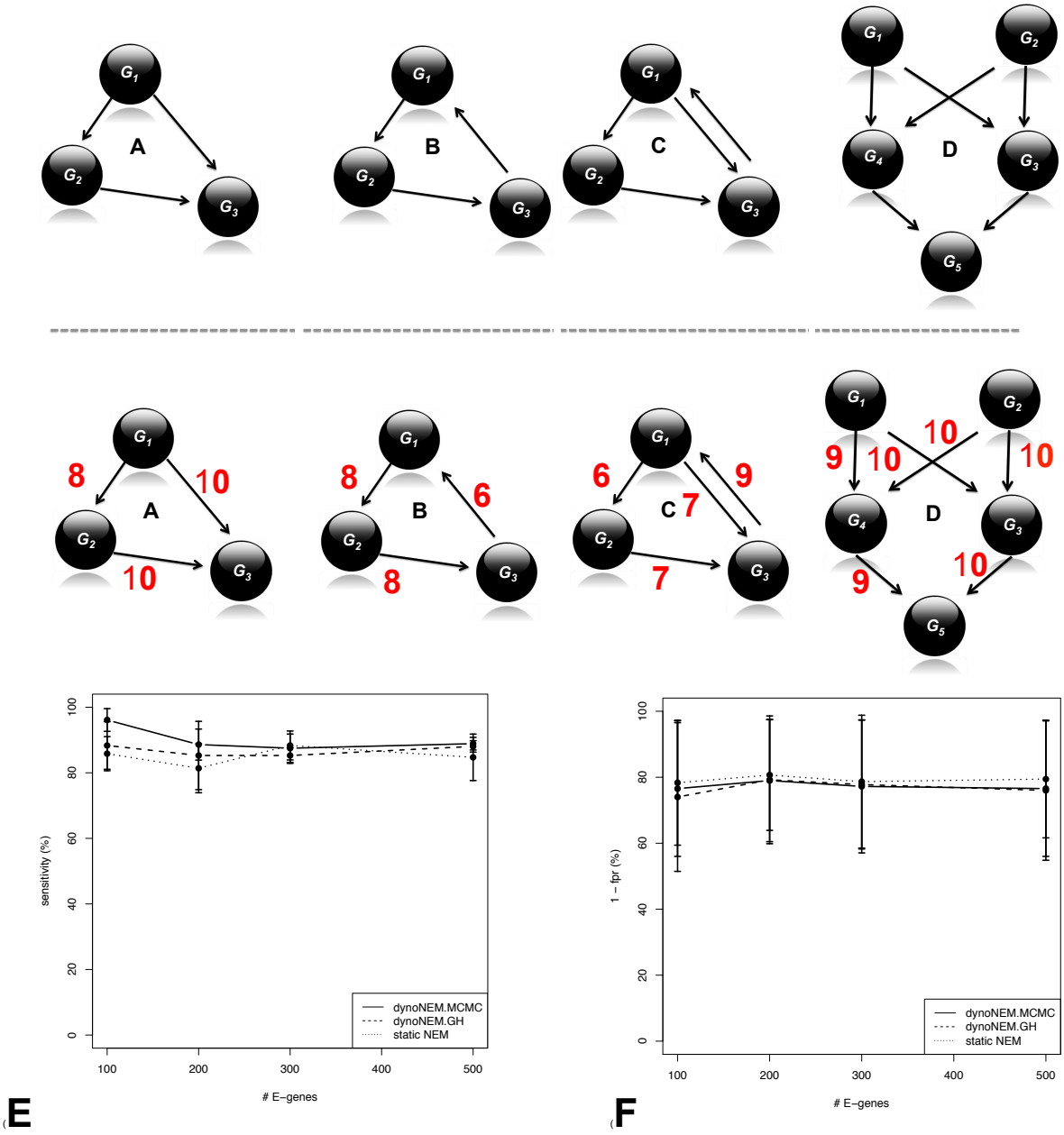

Figure 4.9.: Reconstructing four network motifs (A) Feed forward (B) Feed back (C)Double feed back and (D) Perceptron. The performance for reconstruction from simulated data has been provided in (E) Sensitivity (F) Specificity

of the MCMC sampler is critical with regard to sampling from right distribution. The seeding of MCMC sampling with a greedy hill climber makes a faster convergence possible in case of dynoNEMs. The results were analyzed in terms of log likelihoods against the iteration index (Figure 4.3). This proved a fast convergence of the MCMC algorithm in all cases.

4.3.4. Application: Network inference from "murine stem cell development" data

dynoNEMs were applied onto a data-set investigating molecular mechanisms of self-renewal in murine embryonic stem cells (Ivanova et al., 2006).

The data

The developed methods were applied on RNAi microarray data from mouse embryonic stem cells by Ivanova *et al.* (Ivanova et al., 2006). Ivanova and coworkers used RNAi loss-of-function techniques to perturb a set of genes, followed by a global gene expression measurement. The data set consists of time series microarray measurements for RNAi knockdowns of each of the six genes namely, Nanog, Oct4, Sox2, Esrrb, Tbx3 and Tcl1. The cells were grown in the presence of leukemia inhibitory factor (LIF), retaining their undifferentiated self renewing state acting as positive controls. Differentiation associated changes in gene expression were measured by replacing LIF with retinoic acid (RA), thus inducing differentiation of stem cells (negative controls). RNAi silencing of the six afore mentioned genes was done in (LIF+, RA-) cell cultures to investigate their potential for induced cell differentiation. Microarray expression measurements at 6 to 7 time points in 1-day intervals were taken for the two controls (positive and negative) and the RNAi assays.

The data-set consists of time series microarray measurements for RNAi knockdowns of each of the six genes namely, Nanog, Oct4, Sox2, Esrrb, Tbx3 and Tcl1. The platform used was Affymetrix MOE430A and MOE430B chips. The Affymetrix MOE430A chip was used in context with the dynoNEM application. The six perturbed genes (Nanog, Oct4, Sox2, Esrrb, Tbx3 and Tcl1) served as the S-genes for dynoNEM. The genes showing significant fold changes ($\geq$2 fold up/down-regulation across all time points) in response LIF depletion were selected as E-genes (122 E-genes)(Anchang et al., 2009). The continuous expression data used was a transformed data as binary values as described in Anchang et al. (2009). The E-gene values were set to be 1 if expression value is close to negative control else 0 The heat map for the processed data along the time course is shown in figure 4.10.

Reverse engineering: Greedy Hill Climber

The dynoNEM model with GHC algorithm was applied on the data with a non-parametric bootstrap procedure, where the E-genes are sampled from the original data with replacement and network is reconstructed with these E-genes. We record how often each edge appeared in 1000 (number of bootstrap) inferred networks (one network per bootstrap sample). We then computed exact binomial distribution confidence intervals (95%) for the appearance probability of each edge via R-package *binom* (Dorai-Raj, 2014). The validity of bootstrap probabilities for edges was further checked in a simulation (see Supplementary Material). Only edges with lower confidence bound >50% were regarded as reliable and shown in figure 4.11. The median time lags for all edges were 1, i.e. had the same speed.

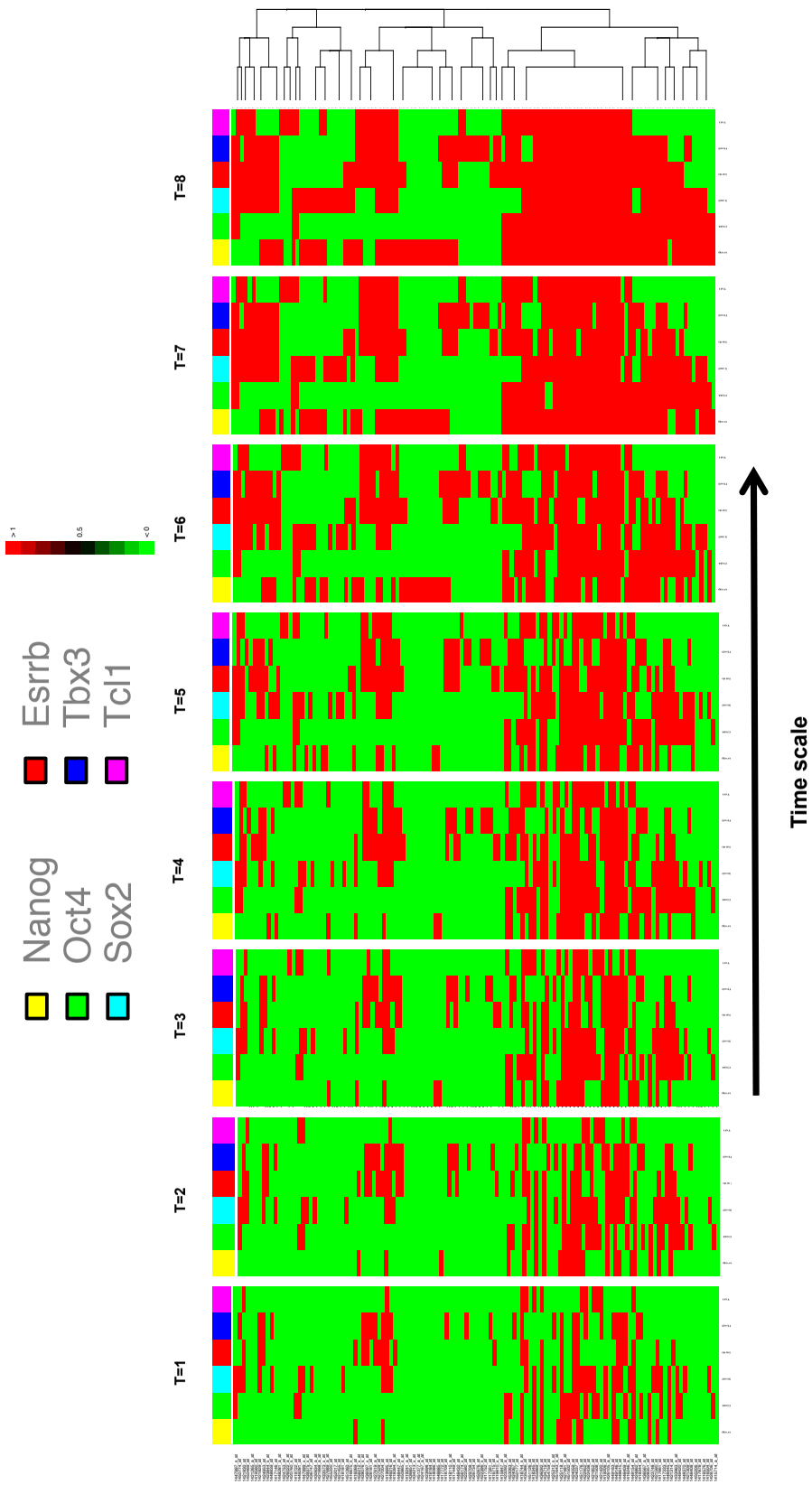

Figure 4.10.: Heatmap for the murine stem cell development data along the time course, showing the propagation of perturbation effect along the time scale from $T = 1$ to 8. The X-axis shows the binarized fold changes (fold change indicated in red an no change in green) of for every perturbation and the Y-axis depicts the E-genes.

The result were in general agreed with the D-NEM approach of Anachang *et al.* (Anchang et al., 2009). The cascades like (1) $Tbx \rightarrow Esrrb \rightarrow Oct4 \rightarrow Tcl1$, (2) $Nanog \rightarrow Oct4 \rightarrow Tcl1$ and (3) $Sox2 \rightarrow Oct4 \rightarrow Tcl1$; were mapped by the dynoNEM. Unlike D-NEM, dynoNEM mapped cascade (2) and (3) to be a Y-structure rather that a part of a coherent linear path with $Nanog$ uptream of $Sox2$. Although, the D-NEM maps several indirect/direct interaction pairs, dynoNEM maps only two of these (1) $Tbx \rightarrow Oct4 \Leftrightarrow Tbx \rightarrow Esrrb \rightarrow Oct4$ (2) $Esrrb \rightarrow Tcl1 \Leftrightarrow Esrrb \rightarrow Oct4 \rightarrow Tcl1$.

Reverse engineering: Markov Chail Monte Carlo

Furthermore, the MCMC algorithm was applied on the same data set. The MCMC uses 50000 burnin and 50000 sampling iterations. The Bayes factor compared to the greedy hill climber algorithm, obtained for the network on log scale was 19.356 indicating the high reliability of the network. The resulting network was not very different from the GHC algorithm results but contained different time lags (Figure 4.12). The network mapped showed two different kinds of edges based on time lags. We refer the edge with time lags = 4 as slower and those = 1 as fast interactions (red and green edges in figure

The reconstructed network could map the three cascades (1) $Tbx \rightarrow Esrrb \rightarrow Oct4 \rightarrow Tcl1$, (2) $Nanog \rightarrow Oct4 \rightarrow Tcl1$ and (3) $Sox2 \rightarrow Oct4 \rightarrow Tcl1$ similar to the GHC algorithm. The shortcut observed in this network was the pair $Sox2 \rightarrow Oct4 \rightarrow Tcl1 \Leftrightarrow Sox2 \rightarrow Tcl1$. This can also be observed in D-NEM and dynoNEM GHC network. Though other such pairs were not observed as in case of GHC algorithm. The most striking difference from GHC based network is the cycle $Oct4 \rightarrow Tcl1 \rightarrow Oct4$. This cycle has edges with different time lags 1 and 4. This marks that after certain time lags Tcl acts back on Oct4. 4.12).

Biological Implications

The murine stem cell network created here is transcriptional network with 6 regulators (the S-genes here). Nanog and Sox2 according to a biological model explained by Ivanova *et al* is considered as a global regulator (Ivanova et al., 2006) critical for pluripotency in stem cells (Silva et al., 2009). It has been proposed that Nanog harnesses its partner Oct4 and Sox2 to create ground state pluripotency (Silva et al., 2009). Oct4 on the other hand controls the transcription factor Tcl1 (Matoba et al., 2006). These biological findings can be comfortably explained by the networks inferred by both dynoNEM models (GHC and MCMC). Furthermore, biologically it is perceived that modestly raised levels of Nanog compensate for the loss of Esrrb, Tbx and Tcl1, implying cross-talk between the two pathways (Rao and Orkin, 2006; Ivanova et al., 2006) which is also explained by dynoNEM models. Thus the role of various involved gene is well explained by the models.

4.3.5. Comparing GHC and MCMC

The network hypothesis for both the methods were found to be highly significant against null hypothesis ($p - value = 5.089 \times 10^{-4}$ for GHC as well as MCMC) via a permutation test with 1000 permutations. This was done by sampling N random permutations of the node labels keeping node degree distribution same as in the given network. An exact p-value is computed by counting, how often the likelihood of the permuted network is bigger than that

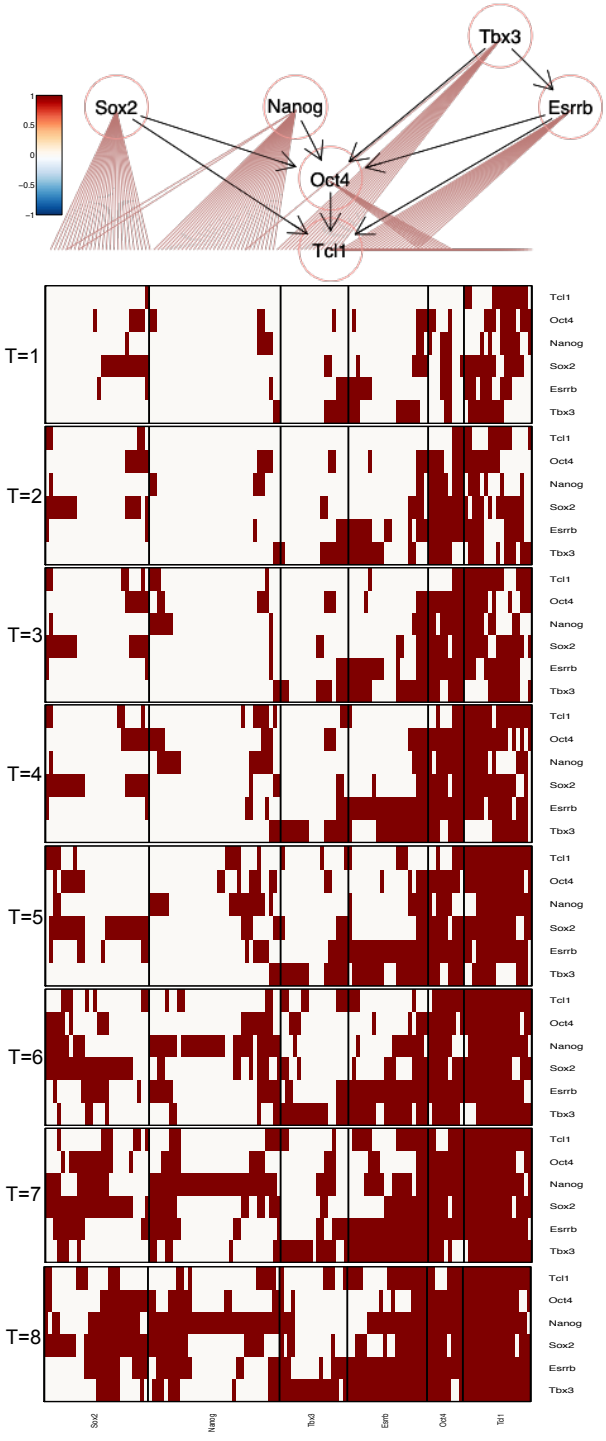

Figure 4.11.: Network for murine stem cell development data reverse engineered via Greedy Hill Climber algorithm (top), Heatmaps (bottom) depict estimated perturbation effects along the timescale 1 to 8. In the heat map the X-axis indicates the set of E-genes for each S-gene shown via lines from node to the heatmap columns. The Y-axis indicates the perturbation effect

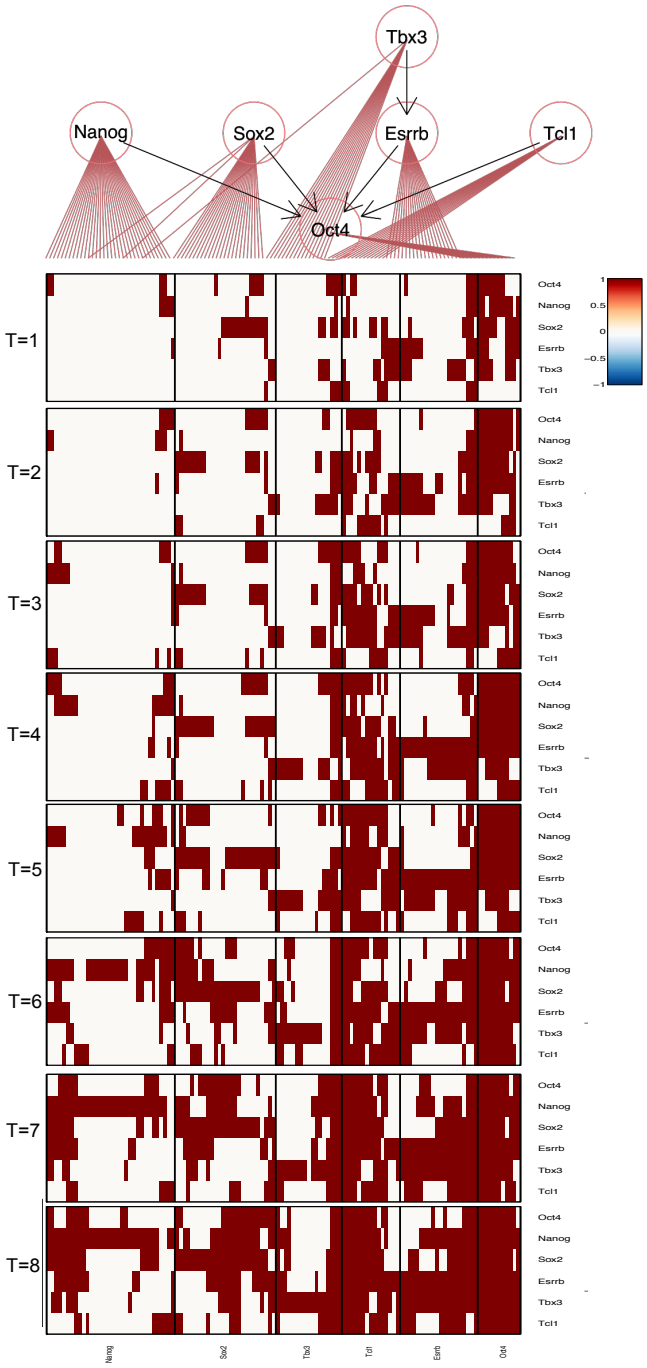

Figure 4.12.: Network for murine stem cell development data reverse engineered via MCMC algorithm with edge labels indicating the time lag (top), Heatmaps (bottom) depict estimated perturbation effects along the timescale 1 to 8. In the heat map the X-axis indicates the set of E-genes for each S-gene shown via lines from node to the heatmap columns. The Y-axis indicates the perturbation effect. The edge labels depict the time delays in the signaling.

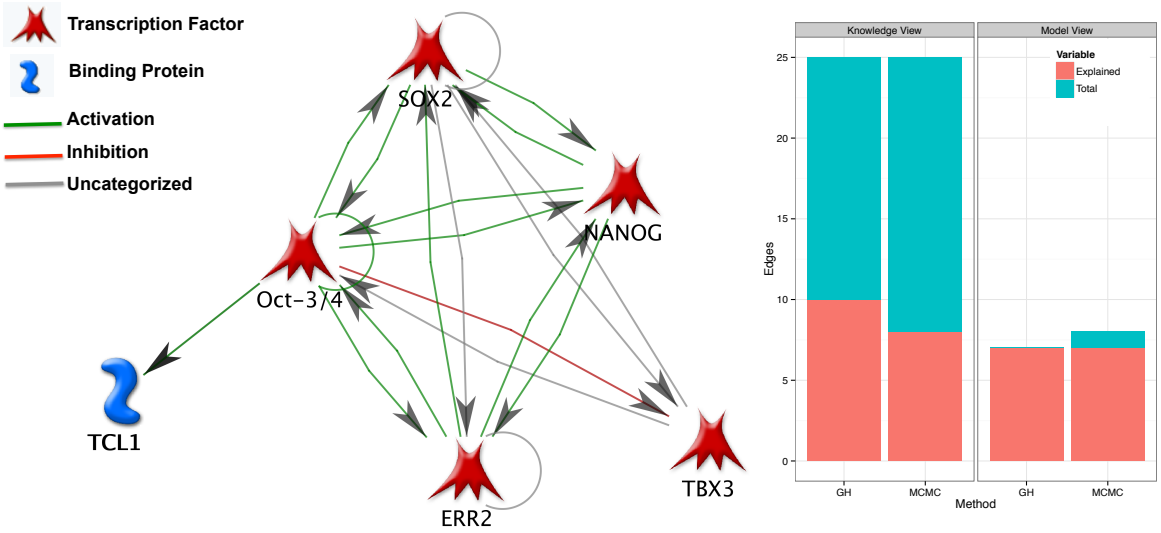

Figure 4.13.: Literature network from Metacore for Ivanova data (left) and its comparison with inferred networks via GHC and MCMC based dynoNEM. The plot (right) represents two views: knowledge view represents the total number of edges (among the S-genes) found in the literature and the number of these edges that could be explained (Legend: Explained) based on dynoNEM inferred network. The model view presents all the edges in the inferred network and how many of these edges were explained (Legend: Explained) by the literature (Metacore^{TM}) network. The corresponding list can be found in appendix C

of the given network. The network modification operations used are edge weight increase, decrease and swap.

The network was compared with a literature curated network, created via MetaCore^{TM}tool [1]. It can be observed that both the networks can be mostly explained by the literature (90% to 100% edges). MCMC method could discover one new edge (See appendix B). This was accompanied with cyclicity in pathway between Oct4 and Tcl1. On mapping the network onto literature network both algorithms could explain exactly same literature pathways (Appendix B). The main difference between GHC and MCMC based network is the cyclicity in terms of pathway $Oct4 \rightarrow Tcl1 \rightarrow Oct4$. Comparing the E-gene S-gene association with the heatmaps in figure 4.11 and 4.12 the difference is observed for the gene involved in this cyclic pathway.

4.3.6. dynoNEM vs D-NEM

dynoNEM are a computationally feasible model to reconstruct networks from time course perturbation data. The proposed method distinguishes itself in two paradigms:

1. It estimates the time lag between perturbation and its downstream effect and not the

[1] http://thomsonreuters.com/products_services/science/science_products/a-z/metacore/ ; Accessed: March 2013

actual signaling time. Nevertheless, it is unclear whether true signaling times on protein level can be estimated from gene expression data

2. The structure learning method used in dynoNEM (GHC as well as MCMC) are fast an efficient thanks to the unrolling of network along time which allows faster likelihood computation.

Avoiding the Gibbs sampling strategy employed by Anchang *et al* has been a critical element in making dynoNEM efficient and hence practical in terms of computational resources. To add MCMC algorithm can actually define the time lag for the interaction thus providing a biologist the capability to distinguish the direct and indirect interactions.

4.4 Summary

dynoNEM is an attractive extension of NEMs framework introduced by Markowetz *et al.* (Markowetz, 2005) from the static to the dynamic case by unrolling the network structure over time. It enables the analysis of perturbation time series data and network inference from it. They allow distinguishing between direct and indirect signaling and to resolve feedback loops. It can serve as a useful tool to generate data-driven hypotheses about signaling and/or transcriptional networks based on high-dimensional time series perturbation effects. Applying the dynoNEMs to infer the network between six proteins (five transcription factors) playing a key role in murine stem cell development, we achieved a good agreement with results published by DNEM (Anchang et al., 2009) and with the biological literature.

This chapter presented dynoNEM a Nested Effects Models extension to reverse perturbation based networks from time series data. The next chapter will introduce the dynoNEM based network inference based on a new kind of data 'microscopic image/video' features, going beyond gene expression data

Chapter 5
MovieNEM: dynoNEM on phenotype data

Last chapter brought about a methodology to reverse engineer biological network using time series gene-expression data. This chapter introduces a further extension of dynoNEM in the direction to use of image features from microscopic movies to infer biological networks, thus widening the scope of network inference to phenotypic data from the cell. The work has been published in a peer review journal and presented at the German Conference in Bioinformatics (GCB 2013) as a highlight paper (Appendix M).

5.1 Motivation

Learning gene network from RNAi perturbation data via NEMs and dynoNEMs has been explained in previous chapters. The methods explained so far use gene expression data. Recent times have seen gene-expression (microarray) data as the most conventional input for network inference and has been established as the state of art (Lee and Tzou, 2009). The rationale behind the choice is the "Central Dogma of Molecular Biology" (Figure 5.1). According to it, the genome itself is not modified when the cell expresses its phenotypes, rather the phenotypes like cellular differentiation, division or tumorigenesis are the outcomes of differential expression of the genome or a part of it, talking to each other (Repsilber et al., 2002).

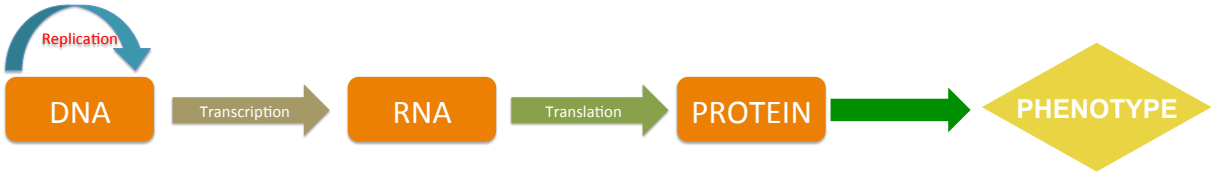

Figure 5.1.: 'The Central Dogma' of molecular biology extended upto phenotype.

Gene expression data provide means of measuring this differential expression via experimental techniques such as RNA-seq and DNA microarrays. However, the cellular phenotype is ultimately controlled and manipulated by genetic information via a network genes. Therefore, these phenotypic features can act as an analogy for gene expression data at a different

level. This concept directs towards the potential use of phenotypic data for the reverse engineering of such networks. Perceiving phenotype signatures of cell is emerging as another high-throughput data source that can serve as measurements to infer biological networks (Nir et al., 2010).

Morphological cellular signatures and observable phenotypic features are emerging as another set of data source, potentially exploitable to infer cellular networks. To illustrate, let us consider the case of tissue morphogenesis or cell division. The details of spatio-temporal patterns of molecules in a cell is reflected in the dynamicity of the morphology of the cell or tissues. Observing the cell/tissue and its component along time and space gives the picture of events within the biological system at a holistic level, shedding light on how interactions among the biomolecules is orchestrated to generate complex structures and functions (Zerial and Kalaidzidis, 2011).

Microscopic images can serve as a promising source of such morphological or phenotypic features. Cellular imaging via microscopy offers a wealth of data on how cells respond to stimuli. To exemplify; RNAi based perturbation of genes creates disturbances in the pathway and this is evident in the physical visualization in terms of microscopic images. Although harnessing such image based data to study biological systems is challenging (Evals et al., 2013). The advent of image-based automated technologies and acquisition of high-throughput quantitative imaging data (Ohya et al., 2005; Carpenter et al., 2006) has paved the way to a great extent in this direction. Methods have recently been developed that attempt to use these technologies to quantify shape (Bakal et al., 2007), DNA morphology (Moffat et al., 2006), and subcellular localization of organelles or proteins (Perlman et al., 2004; Glory and Murphy, 2007) on a single-cell level. Initial analysis was commonly performed by averaging single-cell results to derive mean scores or by clustering such results (Neumann et al., 2006; Bakal et al., 2007).

To principally illustrate the use of microscopic images let us consider a hypothetical situation where a gene (G_4) controls the process of cytokinesis. Cytokinesis is the physical process of cell division, where the cytoplasm of a parental cell divides into two daughter cells. It happens concurrently with mitosis and meiosis. During this process protein filaments; forms a contractile ring around the equator of the cell. The ring then gradually shrinks, drawing the plasma membrane inward forming a cleavage furrow. Ultimately, the ring shrinks to the point of formation of two separate cells each bound by its own plasma membrane [1]. A microscopic imagery can efficiently capture this event through time dimension. Thus, if the gene G_4 is perturbed or artificially switched off, the mitosis/meiosis will occur, but not the cytokinesis. Similarly, the targeted perturbation of all the parental regulators (say G_1, G_2, G_3 etc.) of gene G_4 will result in a similar morphology and similar features in the images. Based on the imagery data an insight into the signaling network (Figure 5.2) can be achieved. This hypothetical example was proved by Evals *et al.* in their work on Drosophilla (Evals et al., 2013) with similar observations in many organisms (Fededa and Gerlich, 2012).

Current chapter aims to exploit such image data from microscopic video to learn biological networks from time lapse cell imaging in RNAi Knock-Downs. The dynoNEMs forms the main

[1] http://www.nature.com/scitable/definition/cytokinesis-100; Accessed: August 2013

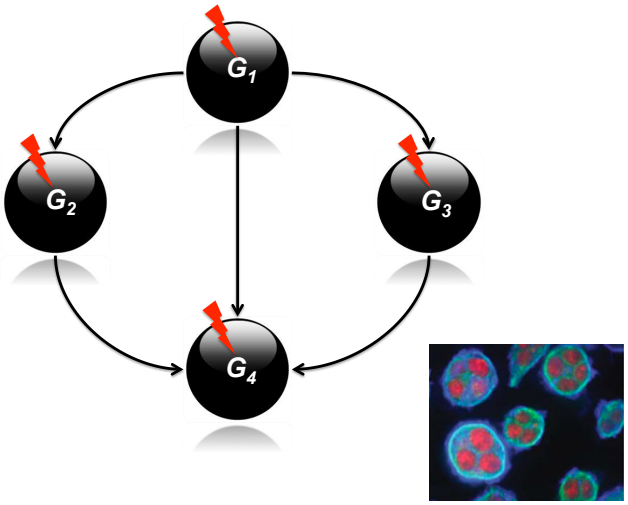

Figure 5.2.: The hypothetical signaling network involving four genes (perturbation indicated in red) for the cytokinesis pathway. The microscopic image of the cell shown at bottom right corner, showing the non-separation of cells. Image based on and redrawn from Evals *et al.* (2013).

framework of the methodology presented here. Henceforth, we refer to this methodology as "movieNEM"

5.2 MovieNEM

"MovieNEM" is an application extension of dynoNEM introduced in the last chapter, towards the use of image features from time lapse microscopic movies in order to infer networks. The entire approach of MovieNEMs (Figure 5.3) consists of image feature extraction, estimation of perturbation effects and network estimation via dynoNEMs and has been discussed under the following heads.

5.2.1. Movie to features

The time-lapse microscope movies used in the work were downloaded from the Mitocheck database [2] (Neumann et al., 2010). The database has about 22,000 knocked-down human genes and are screened for cell cycle defects via time-lapse microscopy. Furthermore, for illumination distinction, the nuclei of HeLa cells were fluorescently labeled by tagging the core histone 2B with GFP (Green Fluorescent Protein). Images of these cells were taken in 30min intervals over a period of 48 hours.

For the entire image analysis step, an open source software CellProfiler (Kamentsky et al., 2011) was used. The software enables biologists without training in computer vision or programming to quantitatively measure phenotypes from thousands of images automatically.

[2] http://www.mitocheck.org ; Accessed: August 2013

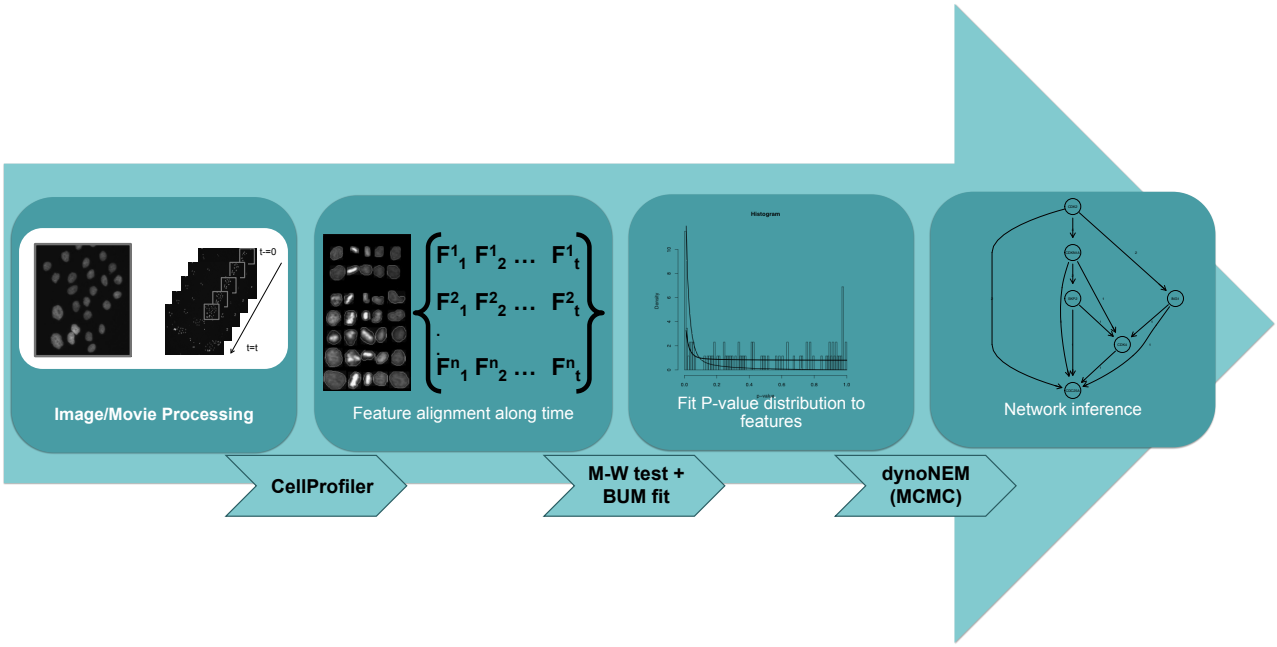

Figure 5.3.: Overview of the MovieNEM approach: Individual movies are first fed into an image processing pipeline consisting of four steps: (i) cell nuclei detection in the individual movie frames; (ii) tracking of the nuclei over time; (iii) calculation of morphological features; and (iv) calculation of cell cycle time. After image processing features are grouped according to the binned cell cycle time. This allows for estimating time-wise perturbation effects. Several movies, each showing one perturbation, are processed in this way and the perturbation likelihoods collected along the binned cell cycle time axis. This allows for applying Dynamic NEMs to infer the network between perturbed genes via MCMC

The software contains already developed methods for many cell types and assays and is also an open-source, flexible platform for the sharing, testing, and development of new methods by image analysis experts. It contains: advanced algorithms for image analysis which, can accurately identify crowded cells and non-mammalian cell types. The software is a modular with a flexible design that allows analysis of new assays and phenotypes.

In the movies being used here, many cells were affected by a common gene perturbation. Each movie was then processed with the following steps.

Cell nuclei detection in the individual frames of the movies

Detecting the cell nuclei is important to track cell division event. For this Otsu thresholding (Otsu, 1979) was used. It is a clustering based image thresholding for image binarization. It selects the threshold by minimizing the within-class variance of the two groups of pixels separated by the thresholding operator. In addition, the watershed algorithm was used to separate clustered nuclei (Malpica et al., 1997). As we realized that the watershed algorithm often over-segmented nuclei we implemented a segmentation correction scheme in CellPro-

filer based on Chen et al. (Chen et al., 2006). Standard CellProfiler features were used for the morphological description of the nuclei.

Tracking of the nuclei over time

CellProfiler's nearest neighbor tracker was used to track the nuclei. Images in CellProfiler are processed sequentially by frame. The method exploited the fact that the cells did not move quickly between frames. In order to track objects (in our case nuclei), it associated the nuclei in the frames before and after. This allowed the study nuclei in terms of lineages and the timing characteristics in movies.

Calculation of morphological features

Once we have the images processed, it is important to measure features or descriptors. CellProfiler can measure features that include area, shape, intensity, texture and Zernike shape features (Appendix E). For the purpose of our work we used 85 features of different classes that include AreaShape, Intensity, Texture, Radial and Zernike moments (Boland et al., 1998) (Table 5.1). These feature will play the role of measurements analogous to E-genes and form the basis of network inference using dynoNEM.

Table 5.1.: Number of features used from different classes

SNo.	Feature Class	Number of Features
1.	Area Shape	13
2.	Intensity	16
3.	Texture	14
4.	Radial Distribution	12
5.	Zernike	30
	Total	85

Calculation of cell cycle time

The movie was recorded along an absolute time. This has to be converted into a relative time reference frame, which we call "cell cycle time". This frame defines time-points within an idealized cell cycle. Mean cell cycle time T was computed as the quotient of the length of all trajectories (in minutes), divided by the total number of division events observed in the movie. This was done for every movie. With this we define the following scenarios in the cell cycle time under specific conditions

1. For cells that were observed at time t between two division events at times $t1 \geq t2$, we define the (relative) cell cycle time r as the quotient

$$r = \frac{(t - t_1)}{(t_2 - t_1)} \tag{5.1}$$

2. For cells that were observed before the first division event of their trajectory the cell cycle time is defined as

$$r = \frac{1-t}{max(t_1;T)} \tag{5.2}$$

3. If cells in a trajectory of length t_ω were observed after the last cell division, we define

$$r = \frac{(t-t_n)}{max(t_\omega - t_n;T)} \tag{5.3}$$

4. Cells that never divided during the observation period were assigned the relative cell cycle time as

$$r = \frac{t}{t_\omega} \tag{5.4}$$

Thus, we reach the state where data set in form of features measured along the cell cycle time scale starting from raw movies.

5.2.2. Estimating perturbation effects

The perturbations effects in a cell are manifested in terms of the phenotype measured by the set of features compared to a wild type. For illustration, let us consider the measurement for a single feature from two images- one from a perturbed cell and other from a wild type cell. The difference in these measurements is the phenotypic outcome of the perturbation. In order to do so, we split the cell cycle time into 10 equally sized intervals and computed for each trajectory and each feature the median value per interval. It should be noted that at this point that there is one trajectory per individual cell. We typically obtained more than 50 median feature values (corresponding to 50 cells) per interval from one movie. Manual inspection of the data revealed that we could not assume feature values to follow any known parametric distribution. Thus, treated and control conditions were compared via a Mann-Whitney U-test (Hettmansperger, 2011).

We then fitted the distribution of P-values obtained for all 85 features at a given time step via a Beta-uniform mixture (BUM) model, as described in chapter 4. The idea is that the distribution of P-values can be decomposed into a uniform part (the null distribution) and a second part (the alternative distribution), which itself can be modeled via two Beta distributions. It has to be mentioned here that at each time point, the number of compared cells is different. Therefore, the null distribution of P-values (the uniform part of the BUM model) also differs between time points. However, we are here only interested in the alternative distribution (describing the likelihood of an effect).

After having learned the BUM model, each P-value can be converted into a probability of being generated by the null and the alternative distribution, respectively. Thus, for each perturbation, feature and interval, we calculate the probability of being phenotypically different from the wild type.

In order to account for possibly irrelevant features, we used a trick described earlier in (Tresch and Markowetz, 2008). A dummy S-gene "$null$" was added to the signaling graph Ψ which was always unconnected to all other S-genes, i.e. not predicting any downstream effects. Features assigned to "null" are hence irrelevant and do not show significant effects in any of the perturbation experiments.

5.3 Applying dynoNEM

After getting the perturbation data mentioned above along the time scale the dynoNEM algorithm described in chapter 4 was applied onto the data set. For this purpose we used the Markov Chain Monte Carlo implementation of the algorithm. We present the application study in terms of simulated data and real experimental data.

5.3.1. Simulations

The network sampling followed the same protocol as in chapter 4. Besides, 85 features (corresponding to image features) were attached randomly to the S-genes (with uniform probability). A set of 10 simulation on networks with $n \in \{5, 10 \text{ and } 15\}$ were generated. The time lags for edges between connected S-genes were sampled from the set $\{1, 2, 3 \text{ and } 4\}$ via a geometric distribution $P(X = k) = (1 - p)^k p$ with parameter p. It was ensured that at least one of the sampled networks was cyclic as in our previous simulations.

To generate feature data for perturbation of each node in our sampled graph were knocked-down in the way similar to that in chapter 4. This was achieved by following the dynoNEM simulation approach. We sampled p-values for all features attached to perturbed S-gene from the alternative part of the Beta-uniform mixture model $f1$, which is a mixture of $Beta(1, \beta)$ and $Beta(\alpha, 1)$. The α parameter was sampled uniform randomly from $[0.1, 0.2, \ldots 0.9]$, whereas the β parameter was drawn uniformly from $[5, 6, \ldots, 15]$. The coefficient for the mixture of both Beta distributions was sampled uniformly from $[0.01, 0.02, \ldots, 0.49]$. P-values for unaffected features were sampled from a uniform distribution. The length of simulated perturbation time series was kept to 10 time steps in compliance with the experimental data (in the next section). The sampling process was done independently for each time step and perturbation to simulate different distribution characteristics. After p-value sampling, Beta-uniform mixture models (BUM) (Equation 5.5) were fitted to each time point and perturbation (Details in chapter 3), allowing for computing local effect likelihoods.

$$p(D_{ik}(t)|S(t) = s(t), \Theta_{si} = 1) \tag{5.5}$$

The whole data generation process was repeated 10 times for each network.

The dynoNEM algorithm with 50,000 burn-in and 100,000 sampling iterations was executed. The edges with a mean time lag less than 2 SDs away from 0 were discarded in order to filter out edges with very large variance.

The simulations revealed a high reconstruction accuracy in terms of sensitivity and specificity overall, and the performance patterns were same as observed in chapter 4.

In terms of the number of S-genes, time lag distribution the dynoNEM pattern matched our observation in the last chapter for the dynoNEM algorithm (Figure 5.4 and 5.5). For the simulation where we varied the number of $S-genes$, smaller network were reconstructed more accurately while there was a small drop in the performance for larger networks. Nevertheless, all the accuracies were more than 0.8 in terms of sensitivity and specificity. Similarly, for the variation of time lag distribution in terms of exponential distribution parameter (p), the trend was same as corresponding dynoNEM simulations. There was higher accuracy for more non uniform distribution with increase in sensitivity and small decrease in specificity. Please recall a smaller p results in more uniform time-lag distribution.

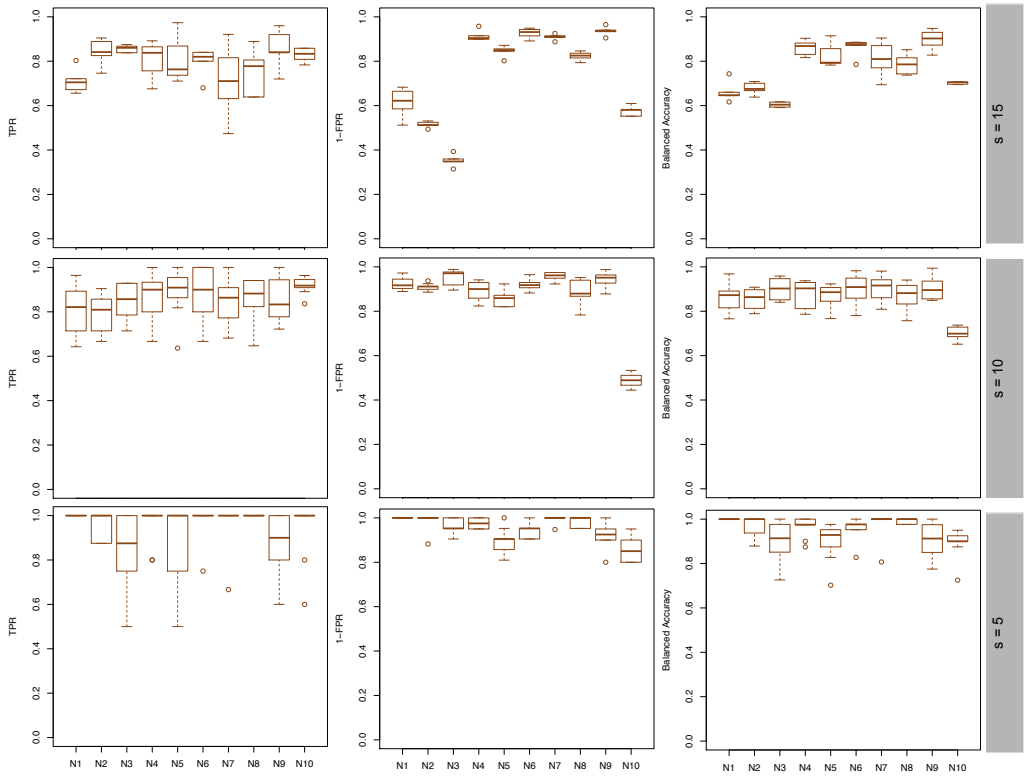

Figure 5.4.: Simulation results for n = (5, 10 and 15) nodes with 10 networks (N1:N10) for every size, p = 0.5 and 85 features. The box plots are for the sensitivity, specificity and balanced accuracy.

Besides these simulations, we ran a new simulation to understand the effect of non-informative features from image to understand the effect of noise in the data.

Dependency on Uninformative Features

Here we looked, in how far our results were influenced by unspecific / noisy image features. For this purpose, we fixed n = 10; p = 0.5 and removed a varying number (5, 10, 40) of informative features from our dynoNEM model. Subsequently we added the same number (5, 10, 40) of unspecific features, for which p- values were drawn from a uniform (0, 1) distribution.

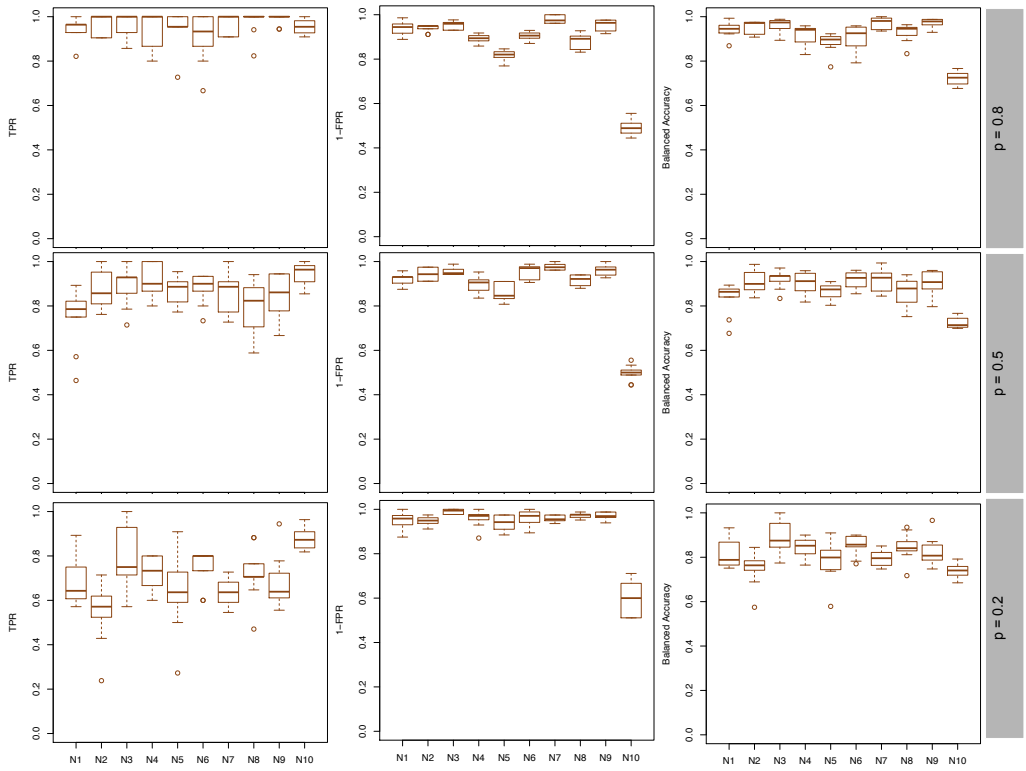

Figure 5.5.: Simulation results for n = 10 nodes with 10 networks (N1:N10). The parameter for time lag distribution was varies as 0.2, 0.5 and 0.8 for 85 features.

Therefore, these noise features had no relation to the network to be estimated. As it can be observed, our method performed robustly against noise features with respect to reconstruction of the network topology (Figure 5.6). Even with 40 uninformative features the average sensitivity and specificity was almost identical to the situation with 85 informative features, which underlines the effectiveness of our feature selection approach described above.

5.3.2. Application

So far we have applied dynoNEM to microarray data to infer networks. The simulations above suggest that it works on phenotype data as well (images/movie). Attempts have been made to reverse engineer such networks from image data for instance, the work of Bakal et al. (Bakal et al., 2007), based on hierarchical clustering of static images and the work of Kaderali et al. (Kaderali et al., 2009), using probabilistic graphical model for only one binary phenotypic variable in static images. A method for the inference of networks from time lapse microscopy based on large numbers of statistical image features does not exist so far to our knowledge. This section presents one such application to reverse engineer a cellular network from movie/video data.

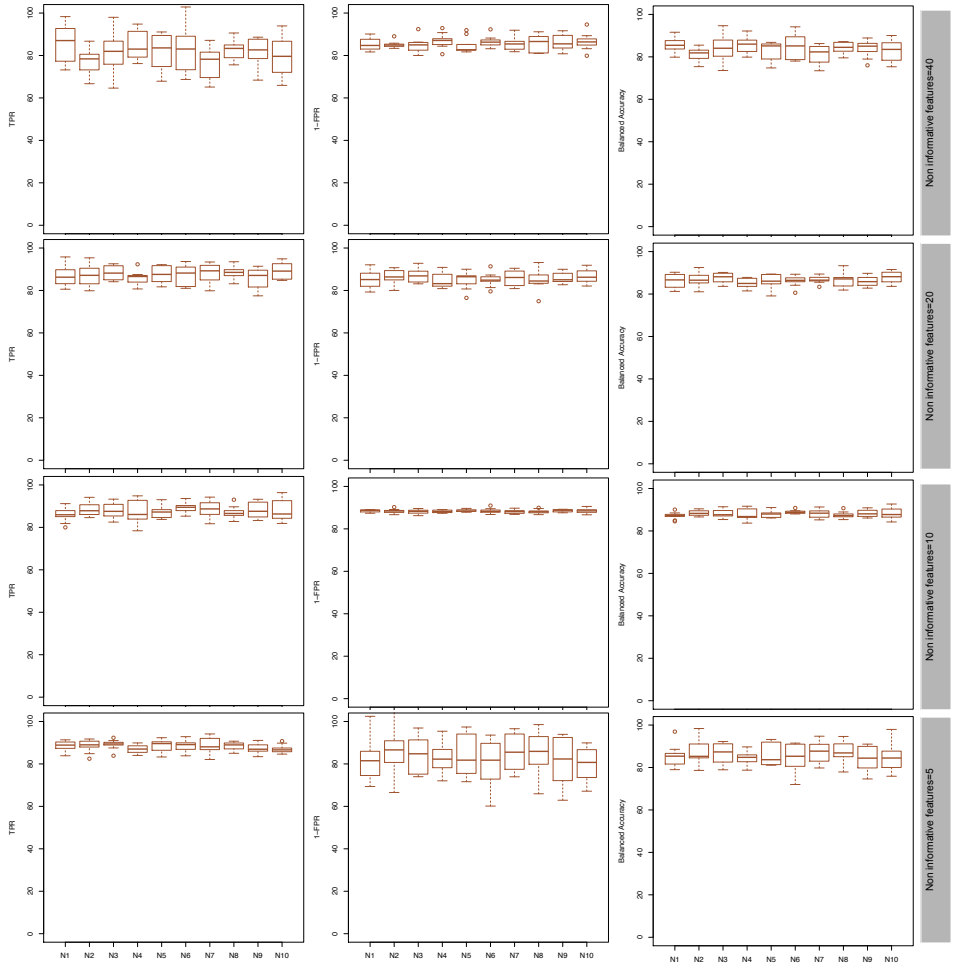

Figure 5.6.: Simulation results for networks with n = 10 nodes, p = 0.5 and 5 to 40 non-informative features. The plots are for the sensitivity, specificity and balanced accuracy. The results shown here represent the distribution observed for 10 networks.

The Data

The time-lapse movies were downloaded from the Mitocheck database (Neumann et al., 2010). The Mitocheck database [3] about 20,000 human genes were knocked-down and screened for cell cycle defects via time-lapse microscopy. The nuclei of HeLa cells were fluorescent labeled by tagging the core histone 2B with GFP. Images of these cells were taken in 30min intervals over a period of 48 hours. The high-content data set from the database can be used for an in-depth analysis of cell division phenotypes, making it suitable for movieNEM.

The movies from the database were processed as explained in the approach section of the current chapter and the features were computed for perturbations and control(Figure 5.7).

[3] http://www.mitocheck.org/cgi-bin/mtc ; Accessed: January 2014

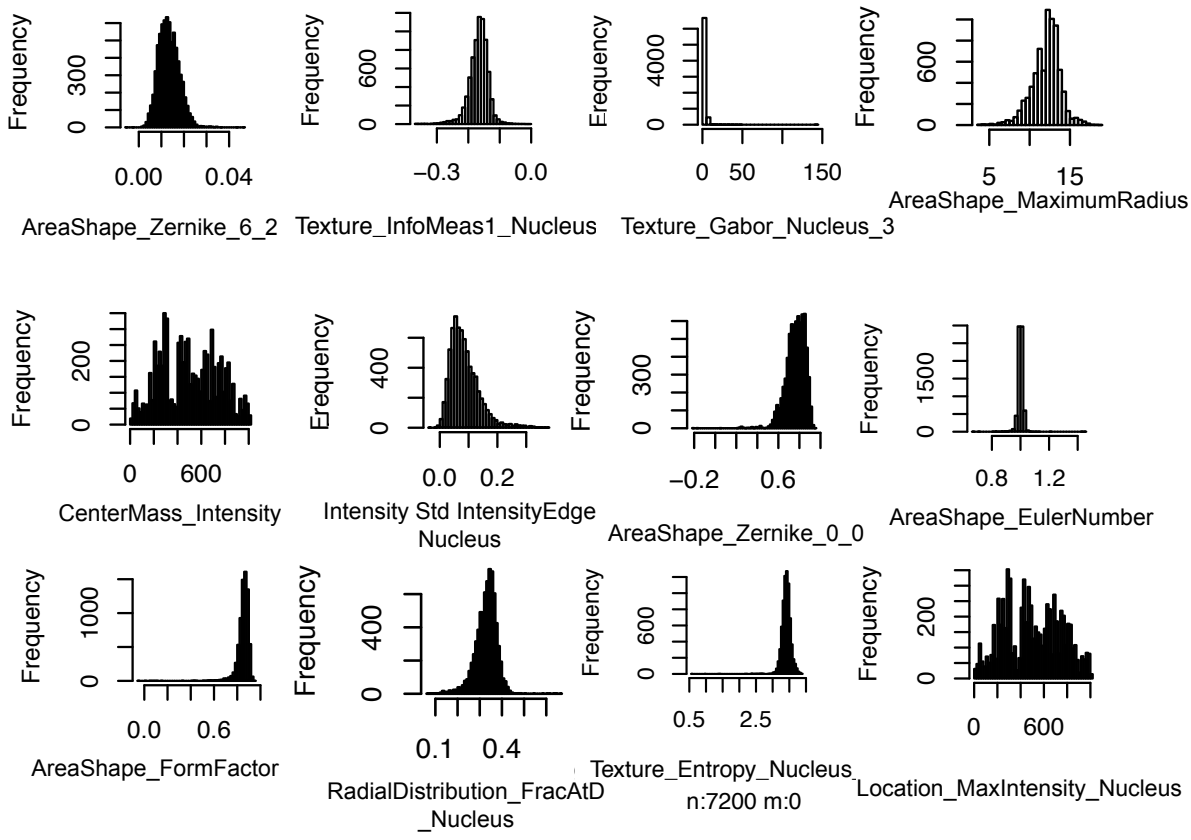

Figure 5.7.: Histograms for some randomly selected images features from the movie data for control sample showing differences in the distribution pattern.

A set of 22 genes were selected from the data set. The selection was based on the fact that these 22 genes demonstrate significant phenotypic perturbation effects in our movies (Figure 5.8). According to a literature based network reconstruction these 22 genes are relatively close to each other and are mainly involved into cell cycle, transcriptional regulation and cell differentiation.

Subnetwork of 6 Genes

The first application we present is on perturbation movies for 6 selected genes out of the 22 defined above. The associated network was inferred by MovieNEM for these selected 6 genes. The purpose of the example is to visualize the results produced by MovieNEM. The number of MCMC iterations in the burn-in and sample phase were both set to 100,000. Figure 5.9 presents the edge-wise posterior expectation network together with the observed and predicted phenotypic effects and corresponding typical cell morphologies (phenotypes or image features) at different time points.

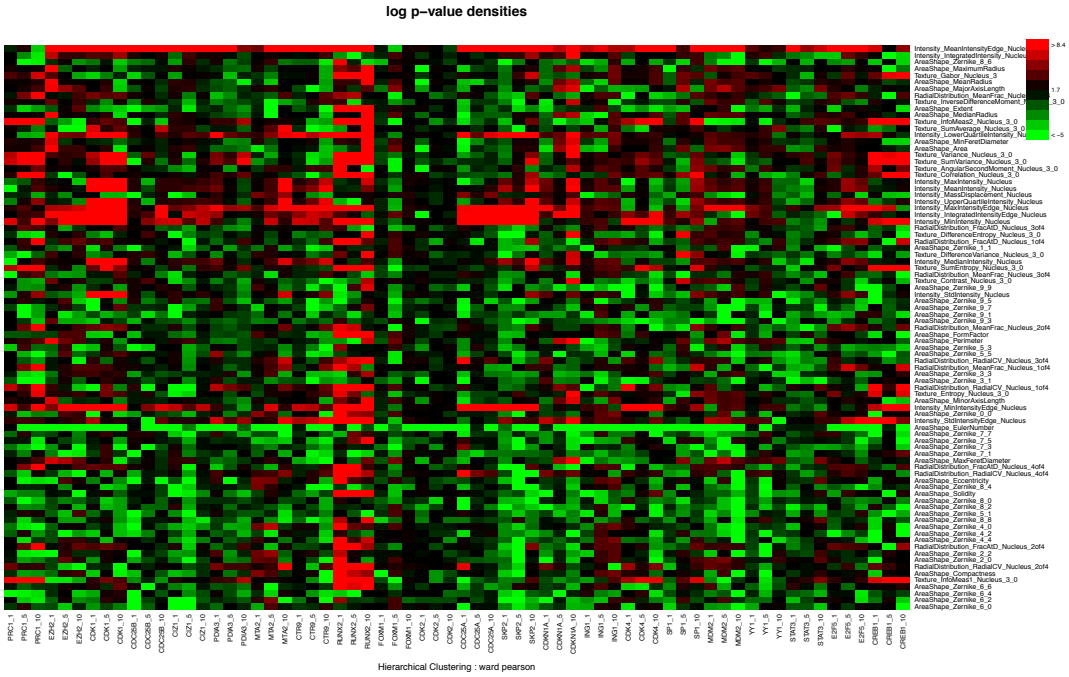

Figure 5.8.: Heatmap of perturbation effects (image features) at time point 1, 5 and 10 for all 22 genes. The heatmap depicts log p-value densities. The more red the stronger the effect

Knocking down CREB1 exhibits perturbation effects at the first time point. STAT3 knockdown affected fewer features than the CREB1 perturbation. This brought CREB1 at the upstream of all the other genes. The STAT3 gene is placed downstream of CREB1 in the network graph as the effect bis seen much later for its knock down. Similar observations can be made, for CREB1 and YY1, E2F5 and STAT3, YY1 and STAT3 and MDM2 and STAT3. The E2F5 showed less interference with CREB1 putting them in a forked relation with YY1. The degree of phenotypic effects increases over time and this causes a development of subset relations in the perturbation effects of different genes. We can observe that at time point $t = 2$ there is such a relation between CREB1 and MDM2. At time point 4 yet more effects and corresponding subset relations are observable, e.g. between E2F5 and YY1 as well as SP1 and MDM2. At the final time point ($t = 10$) there is high degree of noisy subset relation among most of the genes. The $null$ associated features remain unaffected or show unchanged behavior during the time course.

The dynoNEM model presented allows for making predictions of knock down effects on image features sets. Thus, according to the model inferred here, features assigned to STAT3 will exhibit significant perturbation effects throughout the time course because of its downstream location. This is evident from the heatmaps (Figure 5.9). On the other hand, the gene at the upstream will show perturbation effects at the first time point and will maintain its status throughout the later time points as there is no other gene affecting it along the time. CREB1 is an example for this case. The intermediate nodes will evolve their perturbation effect along the time line. To illustrate; MDM2 features at time point 1 should react only to MDM2 and

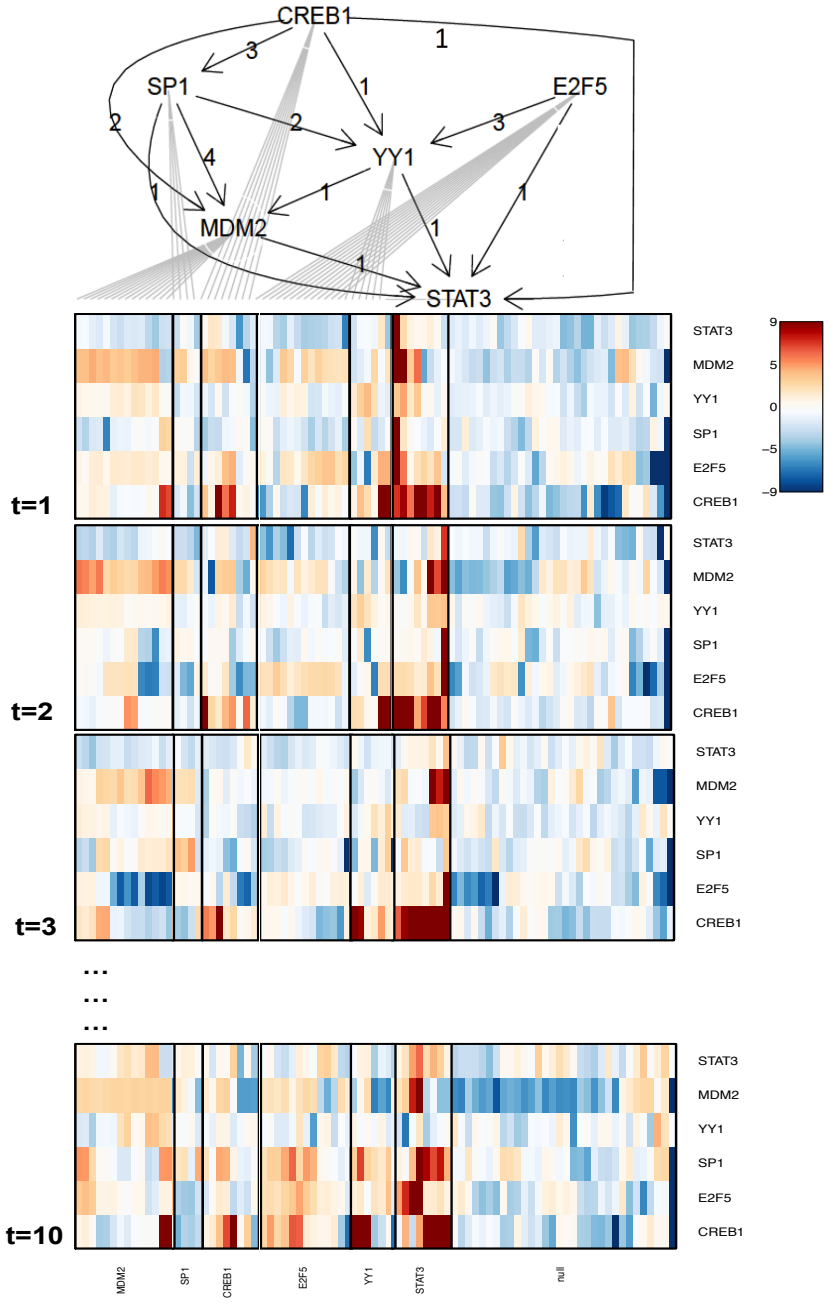

Figure 5.9.: Inferred MovieNEM model for 6 genes. Heatmaps represent estimated perturbation effects at different time points. The ordering of phenotypic features in the heatmaps is due to the MovieNEM model, and gray lines indicate the maximum likelihood of features to perturbed genes. *null* indicates a dummy S-gene, to which features with unspecific response could be assigned during the MCMC procedure. The edge labels on the graph indicate the time lags for corresponding edges.

YY1 knock-down, which is again in agreement with the data. However, at time point 2 MDM2 assigned features are predicted to show effects also under CREB1 and at time point 4 also under SP1 and E2F5 perturbation. All these predictions for effect on features can be observed in the heatmap and verified.

Furthermore, a permutation test was conducted on our network to check if the inferred network could have been expected by chance. For this purpose we permuted the node labels in each network 1000 times and calculated for each permuted network its likelihood (Equation 3.6). All of these likelihoods were inferior to that of our inferred network. The p-value for the network was found to be $\sim 5.89 \times 10^{-4}$. The presented application demonstrates that dynoNEMs are able to learn network structures, which are in agreement with the observed phenotypic data.

The network was also validated with literature based networks from Metacore ^{TM}. The results showed the graph to completely (100%) explainable by the literature (Figure 5.11).

Complete Network

Finally we applied MovieNEM in the same way as described above to infer the complete network among the selected 22 genes with significant phenotype (Figure 5.10). As mentioned above, these 22 genes are mainly involved into cell cycle, transcriptional regulation and cell differentiation. Again our permutation test for edge-wise posterior expectation network scored better than 1000 random S-gene permutations.

Again for the literature based validation, we investigated, in how far our edge-wise posterior expectation network contained edges, which were explainable by paths contained in the literature network (knowledge view). This way all 122 edges could be mapped to known literature pathways from Metacore ^{TM}(Figure 5.11 b). Moreover, out of these paths more than 80% had a length shorter than 3 (Figure 5.11 a).

On the other hand, not very surprisingly the literature network contained additional interactions, which could not be observed in our estimated networks. In consequence 60% of all paths between the 22 genes within the literature network corresponded to paths within our estimated network (Model view). This can have two reasons: Either the additional literature known interactions exist in reality, but MovieNEM could not infer them or they do not exist in HeLa cells and are not inferred. An important factor for false negatives is that we can only infer interactions between genes that show a clear phenotypic knock-down effect. If, for example, proteins A and B can independently activate C, then the knockdown effect of A may be partially compensated by B. In this case we would only see a weak or no knock-down phenotype of A and consequently the edge A $\rightarrow$ C could be missed in our MovieNEM model.

5.3.3. Biological implications

Our MovieNEM inferred CDK1, CDC25A, CDKN1A and E2F5 as central hub nodes. All these proteins play an important role into G1/S and (except for E2F5) G2/M phase transitions according to their GO annotation. All these genes were found to be associated with mitosis

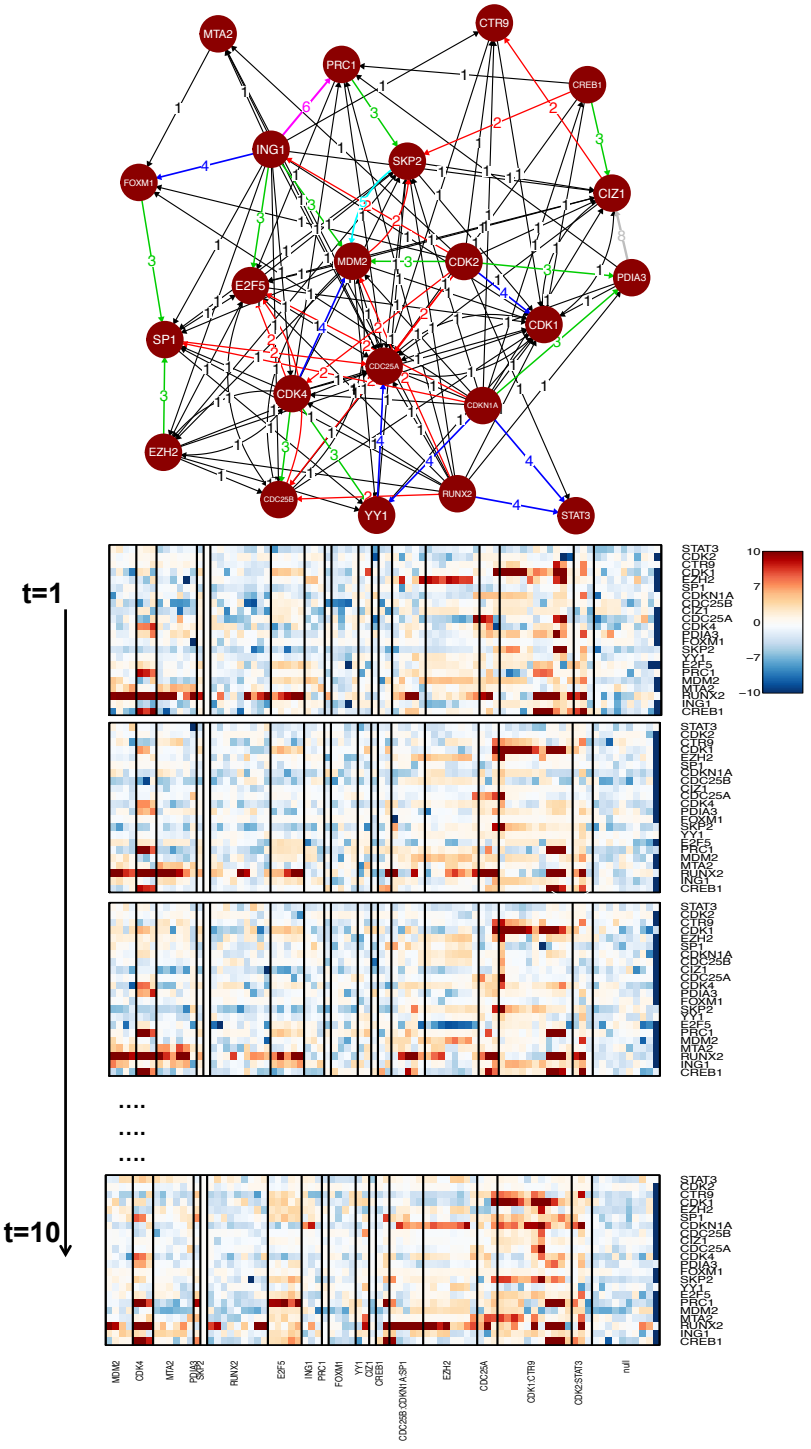

Figure 5.10.: NEM results for perturbation data for 22 genes. (Top) Inferred network for all 22 perturbed genes in the data without transitive closure. The edge color and edge labels indicate the time lag. (Bottom) Estimated perturbation effects at different time points for the 22 genes for the inferred network. *null* indicates a dummy S-gene, to which features with unspecific response could be assigned during the MCMC procedure.

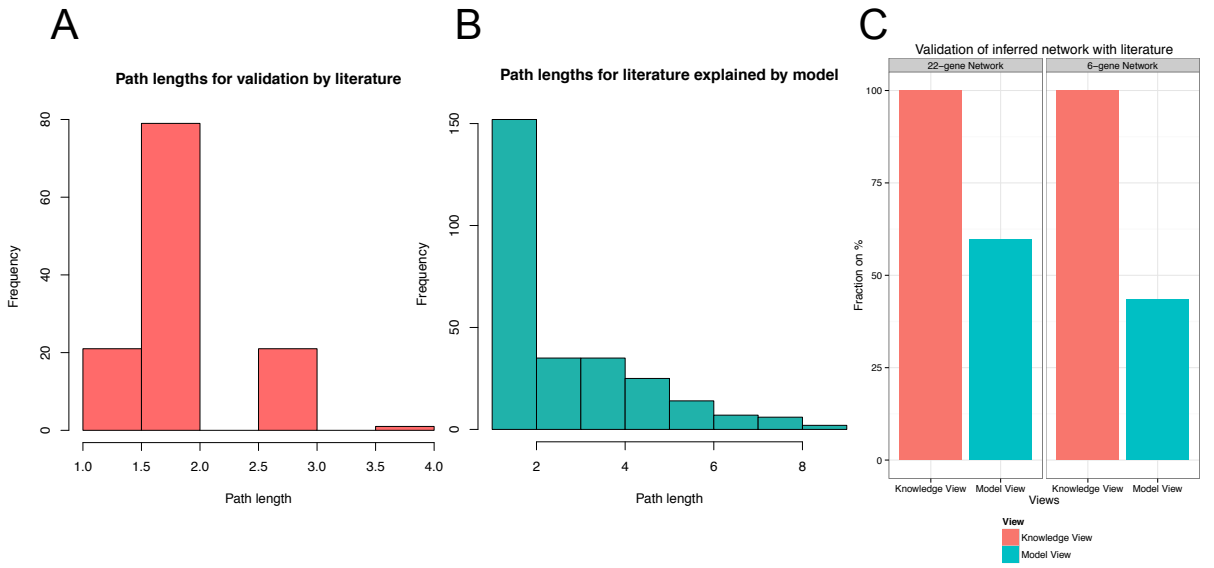

Figure 5.11.: Validation of inferred neworks (22 gene and 6 genes) against literature based networks. (A) Histogram for the lengths of the path in the literature network that explained the edges inferred by MovieNEM (in the 22 gene network). (B) Histogram for the lengths of the path in the inferred network (22 gene network) that explained the edges in the literature based network. (C) Fraction of edges(in %) validated for 22 gene and 6 gene networks. The Y-axis represents the fraction of edges in inferred network that were explained by the literature (Knowledge view) and the fraction of edges in literature network that were explained by the inferred network (Model view)

related GO terms. CDC25A, E2F5 and CDK1 have an extraordinary high in-degree, whereas CDKN1A has a high number of outgoing interactions. CDKN1A and CDK1 are both involved into Ras signaling. It is known that CDKN1A binds to and inhibits CDK1 (Dulic et al., 1998), which is in agreement with the inferred edge CDKN1A $\rightarrow$ CDK1 in our model. E2F5 is a transcription factor playing a role into cell cycle control (GO annotation) and is inhibited by CDKN1A (Dimri et al., 1996), which is reflected by the edge CDKN1A $\rightarrow$ E2F5 in our inferred network.

5.4 Summary

The work opened up a new paradigm of learning cellular pathway structures from phenotypic perturbation effects recorded in time-lapse movies. dynoNEM forms the core of the movieNEM approach (Fröhlich et al., 2011). Extending dynoNEMs to movieNEM required dealing with image data which were aligned along the time scale of the cell cycle. The features from these images then underwent statistical testing and scoring via a Mann-Whitney test (Hettmansperger, 2011). For this the distribution of each feature across different cells were compared for control against treatment. The large number of samples (cells) made the entire process robust and sensitive for detection of effects compared to the use of. One might also

imagine comparing each feature in a single treated cell against the population of control cells as an alternative. This would probably be less sensitive, but could also be valuable, if cell by cell differences are of relevance.

The simulations demonstrated the high sensitivity and specificity of the approach. The application of the approach to movies of 22 siRNA knock-downs from the Mitocheck database, where all estimated interactions were explainable by literature known pathways, proved the potential of MovieNEMs.

The MovieNEM detects and exploits phenotypic differences developed as a response to the targeted perturbation of the cell. In our application, all phenotypic differences are derived from merely one fluorescence staining of the cell (GFP-histone tagging). A significant increase in power of our method can be expected, if multiple staining was used, and the individual fluorescence labels were indicative of the biological process under investigation.

A basic assumption of our present MovieNEM approach is that phenotypic perturbation effects should occur at some point during the cell cycle and then remain until the end of the cell cycle. This assumption may not be fulfilled in all cases. Even more, some perturbations may not yield a phenotypic effect at all. In such cases MovieNEM is not applicable. Finally, it is worth emphasizing that MovieNEM up to now is certainly not an approach for large scale network reconstruction. This is due to the huge network space (increasing exponentially with the number of network nodes), which makes sampling based methods (as well as heuristic approaches) computationally slow.

The reconstruction of cellular networks affords a better insight into the organization of a cell. Targeted RNAi screens have revealed their great potential for this purpose during the last years. Given the fact that the combination with omics-data is not always possible or affordable, image based techniques can offer a promising alternative. Our MovieNEM approach is therefore a step towards better exploiting this information rich data to uncover biological mechanisms.

The chapter proposed a method MovieNEM for learning a cellular network from image (movie) data using dynoNEM. The MovieNEM was applied onto real biological data. The approach outlined here proved to be high accuracy in terms of network reconstruction from phenotype data. Next chapter will propose the use of existing biological knowledge together with data to further improve the accuracy of network reconstruction via probabilistic methods

Chapter 6
The Network Prior

The last chapter augmented the network inference approach for perturbation data by extending the NEMs, via introduction of temporal information in the data. This chapter will focus on use of a prior in network learning from data. The chapter will introduce newly developed method to integrate information from multiple sources as a structure prior in the inference process. The experiment and findings entailed here were presented at the European Conference on Computational Biology (ECCB 2012) and have been also published in a peer review journal (Appendix M). The work also resulted in a new R-package software (available on request).

6.1 Introduction

6.1.1. Motivation

So far, approaches to infer network from experimental data (static and time-course) has been put forward in this thesis. Probabilistic graphical models methods, such as Bayesian Networks or NEMs have been highly successful to infer cellular networks from high throughput experiments. These models allow inferring features of complex cellular networks directly from data. A challenge in that context is the typical low signal to noise ratio in experimental data. To illustrate, high throughput data like microarray is high dimensional, but a limitation is the number of replicates, and the unavoidable measurement noise. A network inferred from such data can bear edges that are not very likely or not possible in reality. The edge just represents an artifact in data. Reverse engineering of networks is thus challenging and often fails to reach the reliable level of accuracy.

Network inference approaches typically attempt to find a network topology that maximizes a score for the given data. However, one of the greatest challenges involved in network inference is search space explosion (Aijö and Lähdesmäki, 2009). This implies that even with a moderate number of variables (vertices), the graph space can be astronomic. However, every possible graph in principle from this space is not plausible practically (Mukherjee and Speed, 2008). Nevertheless, while learning the network this search space can be constrained to certain graphs (Heckerman et al., 1995c). This is achieved by the use of a structure prior that biases the search towards the plausible networks in the search space. This not only makes the search process easier, but also contributes to scientific rationality of the inferred network.

Incorporating prior knowledge into the learning process has thus been identified as a way to address this problem, and principle a mechanism for doing so has been devised, e.g. by Mukherjee and Speed (Mukherjee and Speed, 2008), and Fröhlich *et al.*, (Fröhlich et al., 2007a). The prior needed for such a purpose comes from the domain or expert knowledge. In biology, such knowledge is distributed in the form of databases, knowledge-bases and even biological literature and annotations, like- GO terms. Although there is a lot of knowledge and information, their integration into one quantitative prior gets difficult due to the heterogeneous nature of the different information sources (e.g. GO, KEGG, HPRD, etc.). The current chapter precisely answers the call.

6.1.2. State of art

In the past most authors have concentrated on integrating *one* particular information resource into the learning process (Imoto et al., 2002; Tamada et al., 2003; Nariai et al., 2004; Tamada et al., 2005; Imoto et al., 2006; James et al., 2009): E.g. gene regulatory networks were inferred from a combination of gene expression data with transcription factor binding motifs in promoter sequences (Tamada et al., 2003), protein-protein interactions (Nariai et al., 2004), evolutionary information (Tamada et al., 2005), KEGG pathways (Imoto et al., 2006) and GO annotation (James et al., 2009).

On the technical side several approaches for integrating prior knowledge into the inference of probabilistic graphical models have been published: In (Larsen et al., 2007) and (Eyad Almasri et al., 2008) the authors only generate candidate structures with significance above a certain threshold according to prior knowledge. Another idea is to introduce a probabilistic Bayesian prior over network structures. Fröhlich *et al.* introduced a prior for individual edges based on the a-priori assumed degree of belief (Fröhlich et al., 2007b). Mukherjee *et al.* describes a more general set of priors, which can also capture global network properties, such as scale-free behavior (Mukherjee and Speed, 2008). Using a similar form of prior as Fröhlich *et al.*, but additionally combining multiple information sources via a linear weighting scheme was proposed later proposed (Werhli and Husmeier, 2007). The weights are sampled together with rest of the parameters and the network structure in a specifically designed Markov Chain Monte Carlo algorithm for Bayesian Network inference.

In contrast, (Gao and Wang, 2011) treat different information sources as statistically independent. Consequently, the overall prior is just the product over the priors for the individual information sources. The advantage of the approach is that it is independent from a particular class of probabilistic network models (e.g. Bayesian Networks). The limitation is its strong assumption of statistical independence of information source, which in reality is unlikely since biological knowledge in different databases is not orthogonal to each other.

Most of the methods described above allow a limited number of sources or are not flexible to include new sources of knowledge. In addition, some assumption used during the integration (independence of biological knowledge (Gao and Wang, 2011)) are implausible. The aim of this chapter to address these issues and analyze the effects of various factors (data size, network size, knowledge sources, etc.) on network reconstruction.

6.1.3. Challenges

Though necessary, defining such prior has two major challenges. First, defining the sources of knowledge and second condensing them to one prior. The first challenges can be met with several knowledge sources available in biology and is discussed in section 6.2. Considering the heterogeneity and complexity of these data, abstracting them into one robust prior gets challenging. Hence, a framework is needed to combine these heterogeneous sources of information into a quantitative Bayesian prior. Besides, each knowledge source on network structure is an incomplete recapitulation of the underlying biology, and may include false positive and false negative knowledge. Therefore, one has to consider the issue while constructing a prior. Thus, incorporating structure priors network inference pipeline poses several interesting scientific challenges. The upcoming sections will entail on how to meet these challenges.

6.2 Knowledge sources

The approaches presented here primarily use the GO annotation, pathways databases, protein domain annotation (InterPro - (Mulder et al., 2002)) and protein domain interactions (DOMINE (Raghavachari et al., 2008)) as the sources of biological knowledge. However, new sources can be added if available. The details of these sources and the corresponding measurement is described in the following sections.

GO similarity

The Gene Ontology (GO) has been developed to offer controlled vocabulary for aiding in the annotation of molecular attributes for different model organisms. Predicting the map of potential physical interactions between proteins by fully exploring the knowledge buried in two GO annotations has been explored. Interacting proteins often function in the same biological process, which assumes that two proteins acting in the same biological process are more likely to interact than two proteins involved in different processes.

Briefly, for GO annotation we used the default similarity measure for gene products implemented in the R-package GOSim (Fröhlich et al., 2007a), which resembles the functional similarity proposed by (Schlicker et al., 2006) on the basis of the information theoretic GO term proximity measure by (Lin, 1998). Protein domain annotation was compared on the basis of a binary vector representation via the cosine similarity.

$$Sim(t_1, t_2) = \frac{2IC_{ms}(t_1, t_2)}{IC(t_1) + IC(t_2)} \tag{6.1}$$

where, IC is the information content of term t defined as:

$$IC(t) = -logP(t) \tag{6.2}$$

and IC_{ms} is the minimum subset of two GO terms t_1 and t_2 given as

$$IC_{ms}(t_1, t_2) = \max_{t' \in Pa(t_1, t_2)} \left(ICt'\right) \tag{6.3}$$

Protein-Protein Interactions

Protein-protein interaction (PPI) data can be instrumental to understand biology at a system-wide level. They present the known knowledge about the pairs of proteins that interact in living system and hence can be an important source of knowledge for our approach. PPIs have traditionally been measured using a variety of assays, such as immunoprecipitation and yeast two-hybrid (Y2-H). Such knowledge resides in various databases, like IntAct, HPRD etc. The entire set of known interaction referred as interactome can be structured as a graph. Here we use the interaction data from the PathwaysCommons database (Cerami et al., 2011). To compute a confidence value for each interaction between a pair of genes/proteins we can work at the level of this graph in different ways: One way is to look at the shortest path distance between the two entities. Another way is to employ diffusion kernels (Kondor and Lafferty, 2002). To calculate the shortest path distance between two nodes the function $sp.between$ function based on Dijkastra's algorithm is used from R-package RBGL. The edge confidence is then computed as the inverse shortest path distance.

$$d_{A,B} = \frac{1}{\min\limits_{1:n} (path(A, B))} \tag{6.4}$$

Protein Domain Annotation

Hahne *et al.* (Hahne et al., 2008) and Fröhlich *et al.* (Fröhlich et al., 2008a) found that proteins in distinct KEGG pathways are enriched for certain protein domains, i.e. proteins with similar domains are more likely to act in similar biological pathways. The confidence of interaction between two proteins can be seen as a function of the similarity of the domain annotations of proteins. Protein domain annotation can be retrieved from the InterPro database (Mulder et al., 2002). For each protein we constructed a binary vector, where each component represents one InterPro domain. A "1" in a component indicates that the protein is annotated with the corresponding domain. Otherwise, a "0" is filled in. The similarity between two binary vectors u, v (domain signatures) is presented in terms of the cosine similarity.

$$S_{domain} = \frac{\langle u, v \rangle}{\|u\| \|v\|} \tag{6.5}$$

Domain-Domain Interactions

Two proteins are more likely to interact if they contain domains, which can potentially interact. The DOMINE database collates known and predicted domain-domain interactions (Raghavachari et al., 2008). Calculation for edge confidence (I_{AB}) based on the domine database is done as follows:

$$I_{AB} = \frac{H}{D_A.D_B} \tag{6.6}$$

where H is the number of hit pairs found in the DOMINE database and D_A and D_B are the number of domains in proteins A and B, respectively.

6.3 Prior

Let $\mathcal{D}$ denote our experimental data and Φ the network graph (represented by an $m \times m$ adjacency matrix), which we would like to infer from this data. According to Bayes' rule the probability of network Φ given data $\mathcal{D}$ is given as

$$P(\Phi|\mathcal{D}) = \frac{P(\mathcal{D}|\Phi)P(\Phi)}{P(\mathcal{D})} \tag{6.7}$$

where $P(\Phi)$ is the prior. We assume that $P(\phi)$ can be decomposed into

$$P(\Phi) = \prod_{i,j}^{m} P(\Phi_{ij}) \tag{6.8}$$

e.g.

$$p(\Phi_{ij}) = \frac{1}{\nu} \exp\left(-\frac{1}{\nu}|\Phi_{ij} - \hat{\Phi}_{ij}|\right) \tag{6.9}$$

where $\hat{\Phi}$ is a matrix of prior edge confidences Fröhlich et al. (2007b). A value of $\hat{\Phi}_{ij}$ close to 1, indicates a high prior degree of belief in the existence of the edge $i \rightarrow j$. Our purpose is to compile $\hat{\Phi}$ with consistency from n available information sources. We suppose that each of these sources allows for obtaining an edge confidence matrix by itself, i.e. altogether with n information sources we have n edge confidence matrices $X^{(1)}, X^{(2)}, \ldots, X^{(n)}$.

6.4 LFM: Latent Factor Model

6.4.1. Mathematical formalism

The Latent Factor Model is based on the idea that the prior information encoded in matrices $X^{(1)}$, $X^{(2)}, \ldots, X^{(n)}$ all originate from the true, but unknown network Φ (Figure 6.1). This specifically implies that direct correlations between edge confidences across matrices can be explained by this hidden dependency. In other words Φ is a latent factor explaining correlations between the $X^{(k)}$ ($k = 1, ..., n$). We use this notion to conduct joint Bayesian inference on Φ as well as additional parameters θ given $X^{(1)}, X^{(2)}, \ldots, X^{(n)}$:

$$P(\Phi, \theta|X^{(1)}, X^{(2)}, \ldots, X^{(n)}) = \frac{\prod_{k=1}^{n} P(X^{(k)}|\Phi, \theta)P(\Phi)P(\theta)}{P(X^{(1)}, X^{(2)}, \ldots, X^{(n)})} \tag{6.10}$$

The idea behind this equation is that we can identify $\hat{\Phi}$ with the posterior $P(\Phi, \theta|X^{(1:n)})$. In other words, the prior edge confidences $\hat{\Phi}$ are identical to the posterior edge probabilities learned from our n information sources $X^{(1)}, \ldots, X^{(n)}$.

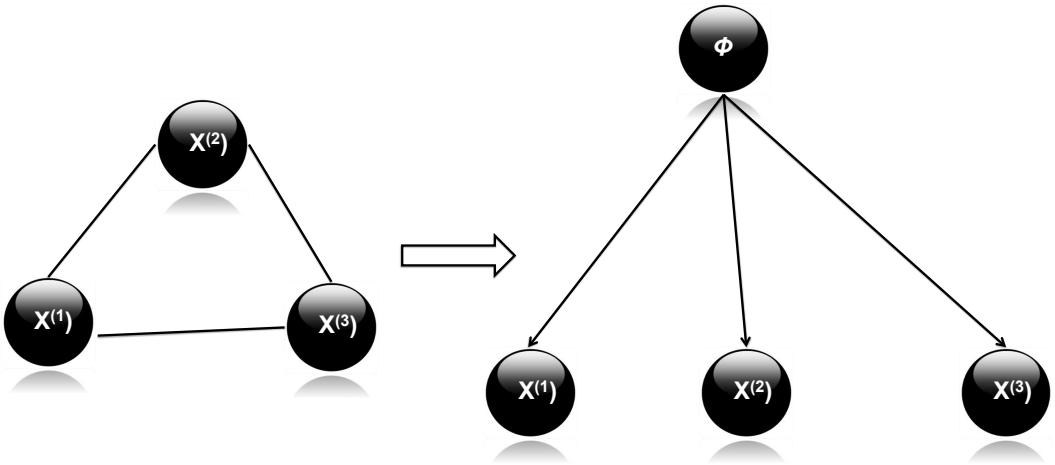

Figure 6.1.: A general Latent Factor Model (LFM). The random variables x^1, x^2 and x^3 are highly related variables (left) and an assumption that these related random variables originate from a common, true but unknown variable ϕ results a Bayesian network (right) in case of networks ϕ is the true but unknown network.

The entries of each matrix $X^{(k)}$ can be assumed to follow beta distributions. More specifically we have:

$$P(X_{ij}^{(k)}|\alpha, \Phi_{ij} = 1) = Be(X_{ij}^{(k)}, \alpha^{(k)}, 1) \tag{6.11}$$

$$P(X_{ij}^{(k)}|\beta, \Phi_{ij} = 0) = Be(X_{ij}^{(k)}, 1, \beta^{(k)}) \tag{6.12}$$

and $P(X^{(k)} \mid \theta, \Phi) = \prod_{i,j}^{m} P(X_{ij}^{(k)} \mid \theta,\ \Phi_{ij})$

Please note that α and β are vectors. That means $\alpha^{(k)}$ and $\beta^{(k)}$ are specific for source k. If the values in matrix $X^{(k)}$ are all either very high (close to 1) or low (close to 0) parameters $\alpha^{(k)}$ and $\beta^{(k)}$ will have a large magnitude. Consequently, $P(X_{ij}^{(k)}|\theta,\ \Phi_{ij})$ will be large, i.e. source k has a large impact. On the other hand, if values in $X^{(k)}$ are rather uniformly distributed, parameters $\alpha^{(k)}$ and $\beta^{(k)}$ will be close to 1, which implies $P(X_{ij}^{(k)}|\theta,\ \Phi_{ij})$ to be close to 0. Thus, such an information source has only small impact. By introducing source specific beta distribution parameters we are therefore, able to weight these source individually.

We employ an adaptive Markov Chain Monte Carlo (MCMC) strategy (Robert and Casella, 2004) to learn the latent variable Φ together with parameters $\theta = (\alpha, \beta)$. For this purpose we define MCMC moves in network space as well as in parameter space. More specifically, in network space MCMC moves are; edge insertion, deletion and reversal. In parameter space α and β are adapted on log-scale using a multivariate Gaussian transition kernel. This is done every 10th iteration. The covariance matrix of the transition kernel is initialized to the identity matrix and every 100th iteration updated to the empirical covariance matrix. The number of burnin steps used is 100000 and number of sampling iterations is 400000 for our MCMC

algorithm here. The convergence of the sampling process has been shown in figure 6.2

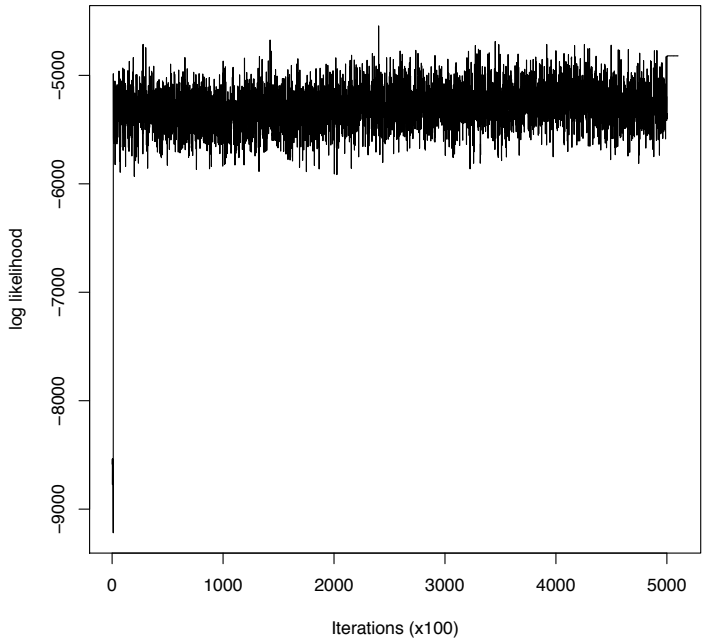

Figure 6.2.: The plot showing convergence of the adaptive MCMC sampling. Along the x-axis are the burnin and sampling iterations (every 100^{th})and the y-axis show the log likelihood for the samples (see equation 6.10)

6.5 NOM: Noisy-OR Model

6.5.1. Mathematical Formalism

The Noisy-OR represents a non-deterministic disjunctive relation between an effect and its possible causes and has been extensively used in artificial intelligence (Pearl, 1988). The Noisy-OR model assumes that the relation between the causes and the effect is non-deterministic, allowing the presence of the effect in absence of any of the modeled causes. The Noisy-OR principle is governed by two hallmarks: First, each cause has a probability to produce the effect and second, the probability of each cause being sufficient to produce the effect is independent of the presence of other causes (Figure 6.3).

In our case $X_{ij}^{(1)}, X_{ij}^{(2)}, \ldots, X_{ij}^{(n)}$ are interpreted as causes and $\hat{\Phi}_{ij}$ as effect. The link between both is given by

$$\hat{\Phi}_{ij} = 1 - \prod_{k}(1 - X_{ij}^{(k)}) \tag{6.13}$$

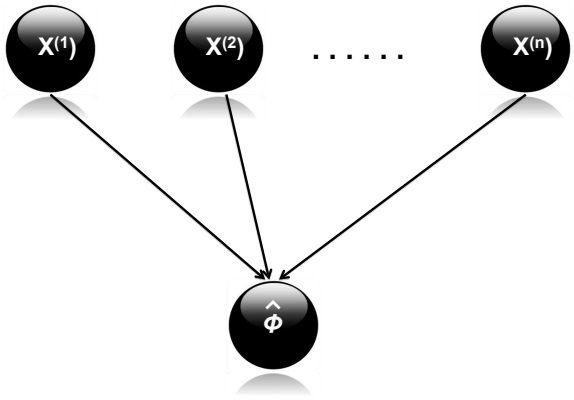

Figure 6.3.: A generalized view of a Noisy-OR model showing the relation between causes $X^{1:n}$ and effect ϕ through a Noisy-OR function

In consequence $\hat{\Phi}_{ij}$ becomes close to 1, if the edge $i \rightarrow j$ has a high confidence in at-least one information source because then the product gets close to 0. Hence, in the Noisy-OR model high edge confidences in one information source can overrule low confidences in other information sources. This is in contrast to the LFM model, where a high-level of agreement between information sources is required in order to achieve high values in $\hat{\Phi}$.

In addition to the above described Noisy-OR model, which integrates edge confidences directly into the consensus prior, we also experimented with a variant based on relative ranks, which is in the spirit of (Marbach et al., 2012): Within each matrix $X^{(k)}$ we first assigned each edge confidence $X_{ij}^{(k)}$ to its rank $R_{ij}^{(k)}$ in descending order. Then, we converted these absolute ranks into relative ranks by dividing each rank value by the maximum rank:

$$R_{ij}^{(k)} \leftarrow \frac{R_{ij}^{(k)}}{\max_{ij} R_{ij}^{(k)}} \tag{6.14}$$

Matrices $R_{ij}^{(1)}$, $R_{ij}^{(2)}$, ..., $R_{ij}^{(n)}$ consisting of relative ranks were then considered in Eq. (6.13) rather than the original matrices $X_{ij}^{(1)}$, $X_{ij}^{(2)}$, ..., $X_{ij}^{(n)}$. We call this method NOM.RNK in the following.

6.6 Simulation results

6.6.1. Reconstruction from simulated sources

In a first series of validation experiments, we looked in how far the true network could be recovered purely from the inferred prior edge confidence matrix $\hat{\Phi}$ after applying a certain threshold. This was solely based on the knowledge without any network inference algorithm and data involved. This allowed us to investigate if our proposed models could really infer the

correct consensus knowledge based of information from many sources. For this reconstruction we simulated artificial knowledge for a KEGG subgraph under study.

Graph sampling

To start with our validation we needed reference networks to be tested. For this purpose we generated 10 networks with 10, 20, 40 and 60 nodes each. These networks later on served as our ground truth. To obtain our ground truth networks we parsed XML files of all KEGG signaling pathways and converted them into graphs via the R-package KEGGgraph (Zhang and Wiemann, 2009). Then, we randomly picked one of these graphs and performed a random walk starting from a randomly selected core node. The random walk was stopped once a predefined number of distinct nodes had been visited, and the corresponding sub-network was returned as a ground truth network.

To evaluate the performance of a prior edge confidence matrix relative to the ground truth network we looked at sensitivity and specificity at different probability cutoffs. In addition we also computed the balanced accuracy (= average of sensitivity and specificity) at each cutoff. We then defined the *optimal Balanced Accuracy* (oBAC) to be the maximum balanced accuracy over all cutoffs.

Simulating knowledge sources

We first simulated matrices $X^{(1)}$, $X^{(2)}, \ldots, X^{(n)}$ by sampling from $Be(1, \beta)$ and $Be(\alpha, 1)$ distributions according Eq. 6.12. The whole simulation was repeated 10 times for different parameter combinations, network sizes m and number of information sources. We compared our LFM, NOM and NOM.RNK approaches against a set of other proposed priors namely.

1. An independent prior (IP), which just takes the product of all matrices X(k) (mimicking the method by Gao et al. (Gao and Wang, 2011))
2. A variant of IP working on relative ranks (IP.RNK) in the same way as described for the NOM method
3. An unweighted average prior (MP), which takes the arithmetic mean of all matrices X(k)
4. A variant of MP, which works on relative ranks (MP.RNK) and is thus identical with the approach proposed by Marbach et al.(Marbach et al., 2012)

Evaluation results

A. Dependency on network size

The objective of this study is to observe the effect of number of nodes (network size) on our proposed priors as well as the dependence of overall reconstruction accuracy of network.

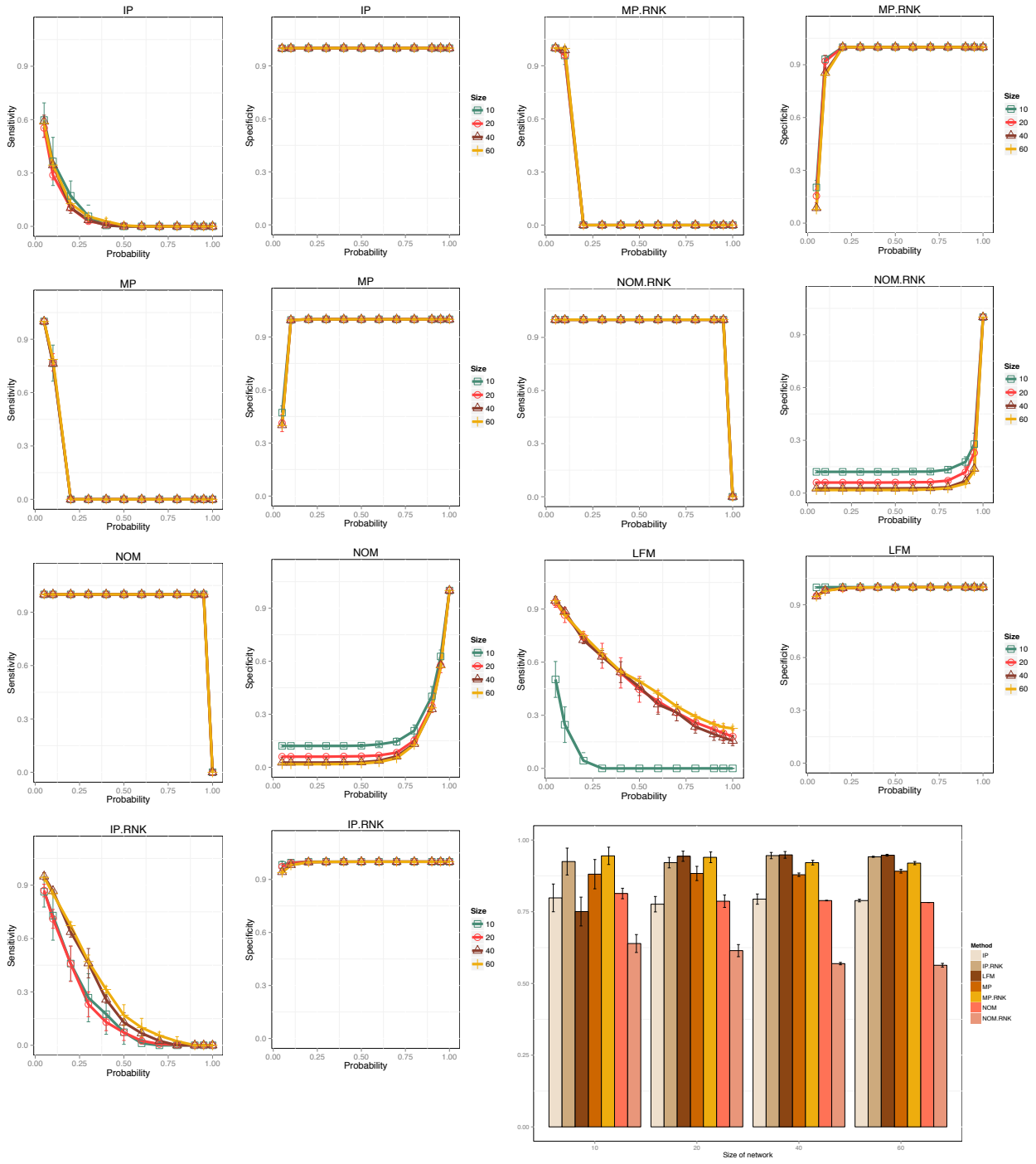

Figure 6.4.: The reconstruction performance during simulation for networks of different size. On the bottom right corner is the corresponding $oBAC$ (optimal Balanced Accuray). The shape parameters for the simulations were kept to $\alpha = 2, \beta = 2$ and number of information sources = 6.

It can be observed that the LFM approach performs better with larger networks as compared to the smaller network. This is evident in terms of better sensitivity and specificity. Decreasing the number of network nodes from $m = 20$ to $m = 10$ yielded a drastic performance loss of LFM (Figure 6.4). This may be explained be the fact that the LFM method learns from the entries in the matrices $X^{(k)}$. The larger these matrices, the more independent observations LFM has to learn from. In contrast, increasing the number of network nodes from $m = 20$ to 40 and $m = 60$ for $n = 6$ sources and $\alpha = 2, \beta = 2$ did not influence the previously observed good performance of LFM significantly. The difference across the size of networks was not prominently observed for other priors as in the case of LFM models.

B. Dependency on number of sources

In order to look into the dependency of number of knowledge sources that can be used to learn a consensus prior, we designed a simulation where the number of these sources was varied and it's reconstruction accuracy was observed.

We simulated the network reconstruction using our approaches for different number of sources viz. $1, 2, 3, 4, 5$ and 6 sources, for networks of size $m = 20$ and $\alpha = 2, \beta = 2$. In this situation we could observe that increasing the number of sources helped to improve the accuracy for most methods (Figure 6.5). The oBAC of our methods were similar to those of the other approaches for a low number of sources ($1, 2$ and 3 sources). However, with an increasing number of sources ($4, 5$ and 6) the performance of LFM increased constantly. For NOM an optimum was reached for $n = 4$ sources, after which the performance declined again, suggesting an increasing loss of specificity.

C. Dependency on shape parameters

To understand the affect of the shape parameters in our information sources we used data generated with different shape parameter pairs and learn the overall network from it. This could give us the information about the affect on learning process with the strength or value range of information beside parameter dependence. To understand the dependency on α and β we first varied both parameters in the range $2, 3, 4$ and fixed $n = 6$ for networks with $m = 20$ nodes.

Our results (Figure 6.6) indicate a dependency of the priors, on the beta distribution shape parameters. Under most parameter settings the methods using relative ranks performed better than their counterparts using raw edge confidences. This was not true for NOM versus NOM.RNK, however, were the opposite behavior was observed: NOM.RNK compared to NOM lacks specificity. Almost all the models performed better for highly correlated sources (i.e. higher α and β values see Figure 6.7). However, the LFM model performed well even with an overall low correlation among sources, which can be interpreted by the ability of the approach to down-weight uninformative/weakly correlated sources. The same held true for MP.RNK. IP was comparable to the other methods for only two parameter combinations

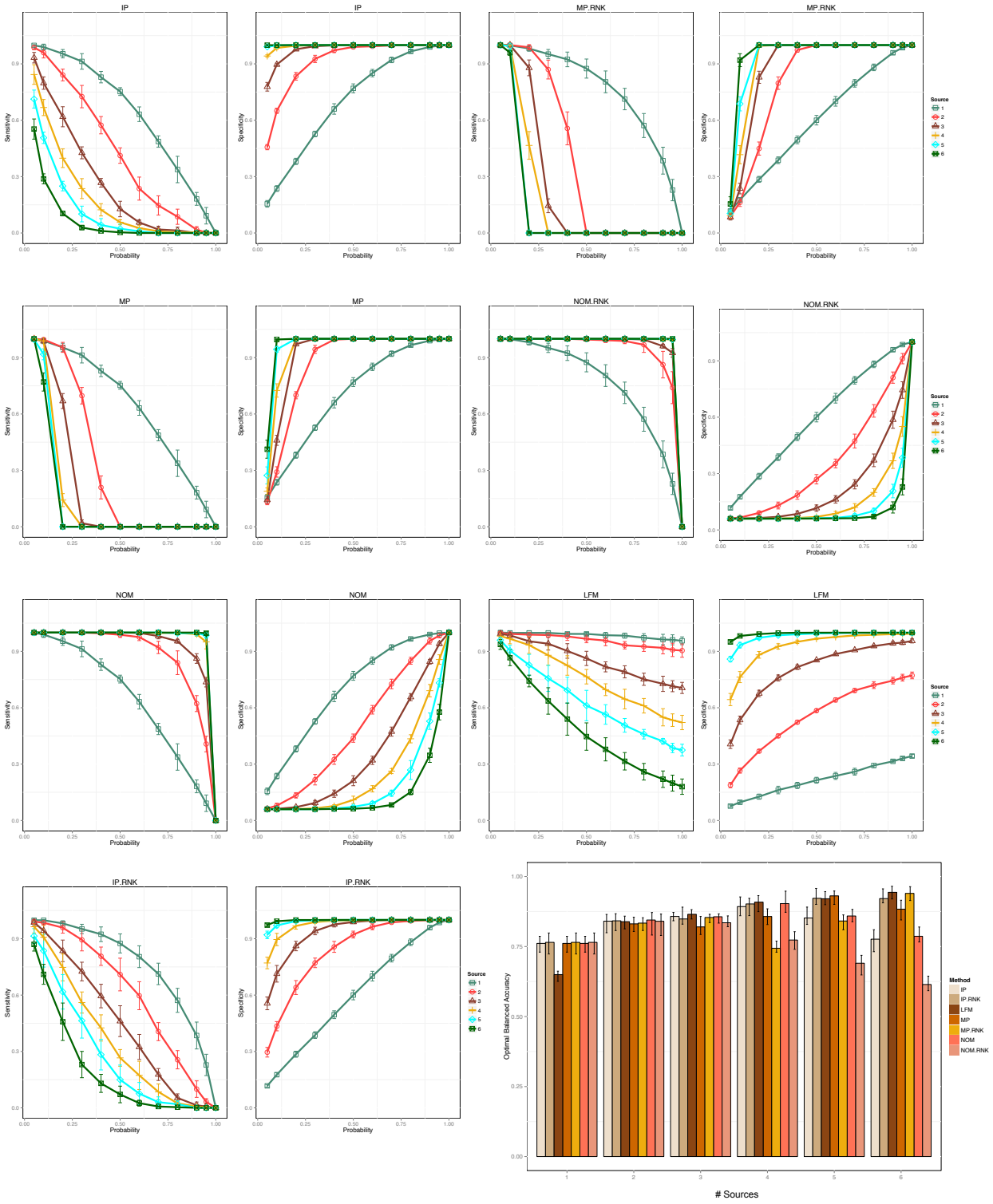

Figure 6.5.: The reconstruction performance during simulation for different number of sources from $m = 1$ to $m = 6$. On the bottom right is the corresponding Optimal Balanced Accuracy. The shape parameters for the simulations were kept to $\alpha = 2, \beta = 2$ and number of nodes m=20.

($\alpha = 4, \beta = 2$) and ($\alpha = 4, \beta = 4$). In both cases numerically the beta distribution yields relatively high values in the $X^{(k)}$ matrices, hence the product does not as quickly tend to 0 as with lower values. NOM could beat LFM only for $\alpha = 2, \beta = 4$. In this case confidence values for non-existing edges are relatively concentrated around 0, and LFM lacks sensitivity. On the other hand LFM performed significantly better than NOM for $\alpha = 2, \beta = 2$ and $\alpha = 3, \beta = 2$ and $\alpha = 3, \beta = 2$. In these cases confidence values for existing edges are relatively high, and NOM lacks specificity. In general it was observable that LFM, MP, MP.RNK, IP and IP.RNK are extremely specific methods, whereas NOM is highly sensitive. Consequently LFM gives the best results in terms of balanced accuracy at low edge probability cut-offs whereas, the NOM does the same at higher cut-offs. The correlation of entries in matrices $X^{(k)}$ were dependent on the beta distribution parameters (Figure 6.71). For example $\alpha = 4, \beta = 4$ yielded high correlations (median 0.7), whereas $\alpha = 2, \beta = 2$ lead to much weaker ones (median 0.2)

The sources with high value for α and low β results the higher mean value (α=4, β=2) for the source (behavior of β distribution). This can be observed in the figure 6.7. It can be observed in figure 6.6 that for these cases the IP performed well otherwise had lower accuracy achieved.

For the quantitative priors from our simulation results we defined the presence (adjacency = 1) or absence (adjacency = 0) of an edge in the network at various probability cut-offs (0 to 0.95). It was observed that the LFM gives the best results in terms of sensitivity and specificity at lower probability cut-offs whereas, the NOM does the same at higher cut-offs. In terms of balanced accuracy the optimal balanced accuracy for LFM is obtained at a cut-off of 0.10 but for the NOM this goes to 0.95. The scene however changes for real data as the confidence from real information sources is not as high as the simulated one and the missing information. This leads to decrease in the probability cut-offs for the NOM model in this case. Compared to this the independent prior method fails to perform because of low of missing information.

D. Weighting of Information Sources

We tested, in how far the automatic weighting of sources provided by the LFM method was able to filter out irrelevant/noisy information. For this purpose we added an artificial source, which contained values sampled uniform randomly between 0 and 1. Figure 6.8 depicts the posterior expectations for and parameters, which were retrieved for individual information sources for 10 sampled networks with nodes. The picture clearly reveals that the posterior expectation of parameters for the noise source was always close to 1, which indicates an influence close to 0 in the likelihood function (Equation 6.12). Hence, the noise source was filtered out effectively.

6.6.2. Inferring KEGG pathway

In a second round of experiments we reconstructed networks sampled from KEGG based on existing biological knowledge encoded in GO, PathwayCommons, InterPro and DOMINE. We ran the whole simulation for networks of different sizes ($m = 10, 20, 40$ and 60).

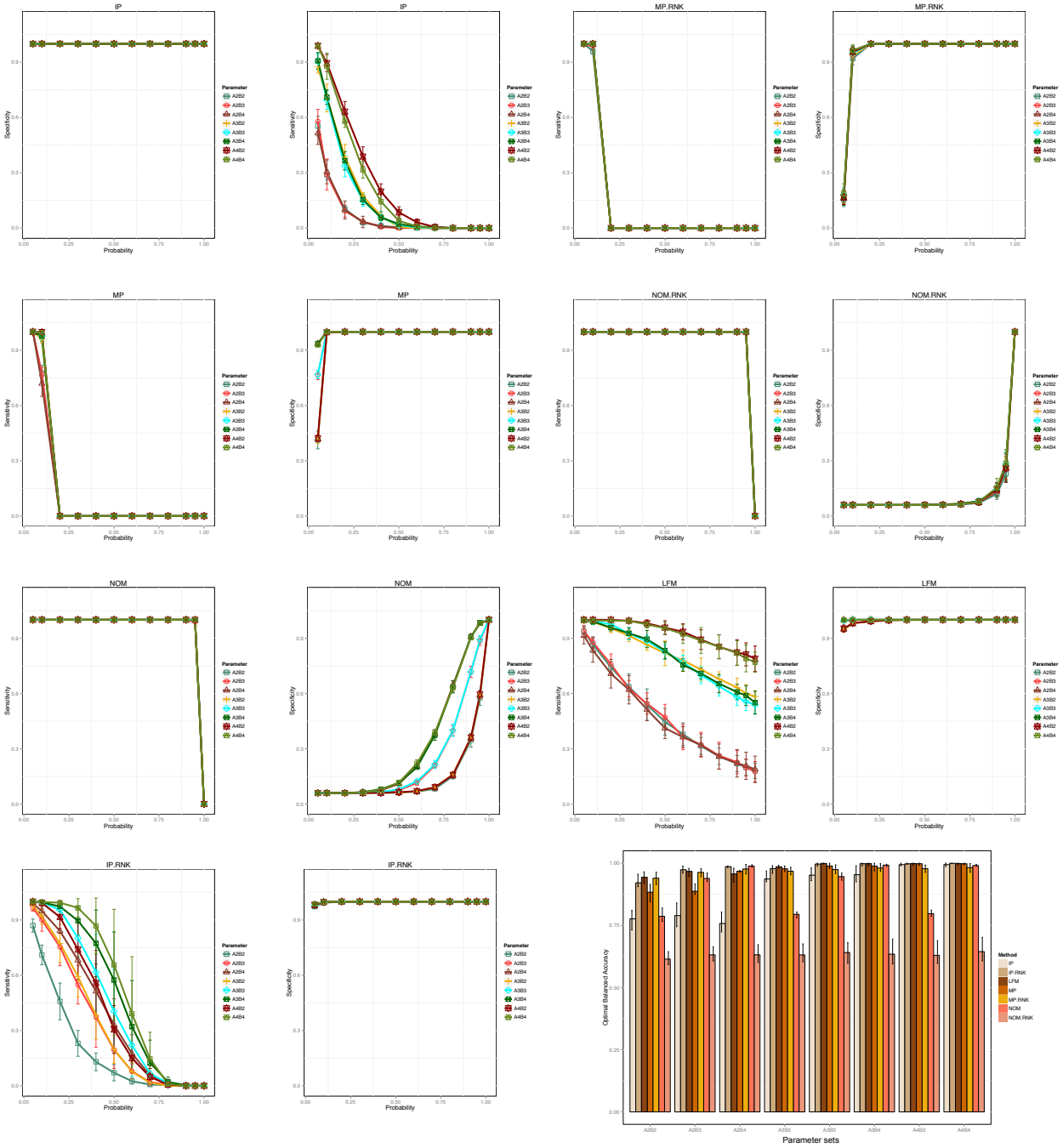

Figure 6.6.: Performance of reconstruction with sources sampled from varying α (A) and β (B). The corresponding oBAC in the bottom right corner. The network size was kep constant at m=20 and number of sources = 6

Our studies revealed a significant improvement of our suggested methods (LFM, NOM, NOM.RNK) compared to the other models in all cases (Figure 6.9). These findings were underlined by a pairwise Wilcoxon signed rank test to assess the statistical significance of

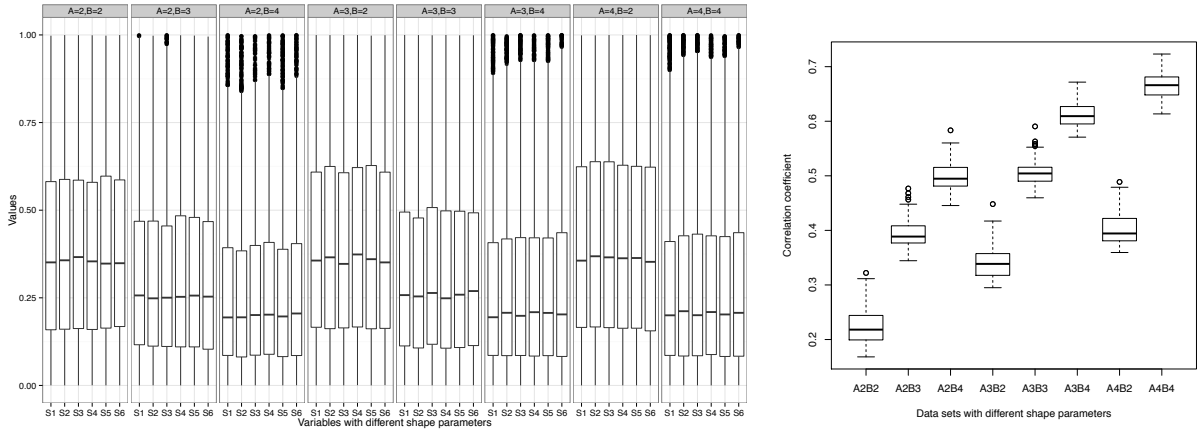

Figure 6.7.: Characteristics of artificially generated information sources with different shape parameters $(\alpha(A), \beta(B)$. (Left) Box-plot showing the distribution of confidence values in 6 artificially generated information sources $(S1 - S6)$. (Right) Boxplot showing the distribution of pairwise Spearman rank correlations across 6 artificially generated information sources. Rank correlations were computed for every pair of artificially generated sources

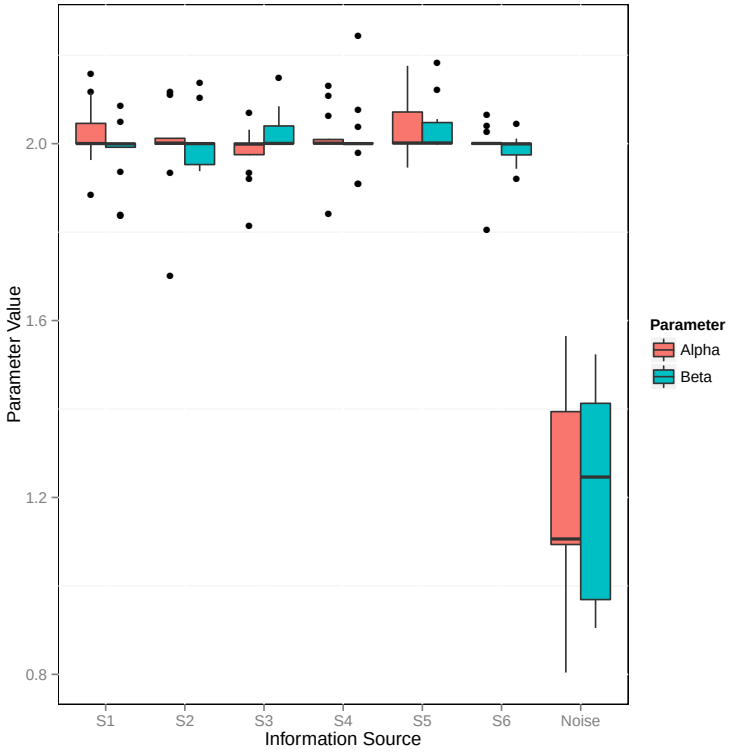

Figure 6.8.: Boxplot of posterior expectation parameters learned for individual information sources in 10 randomly sampled subgraphs of KEGG pathways of size $m = 20$.

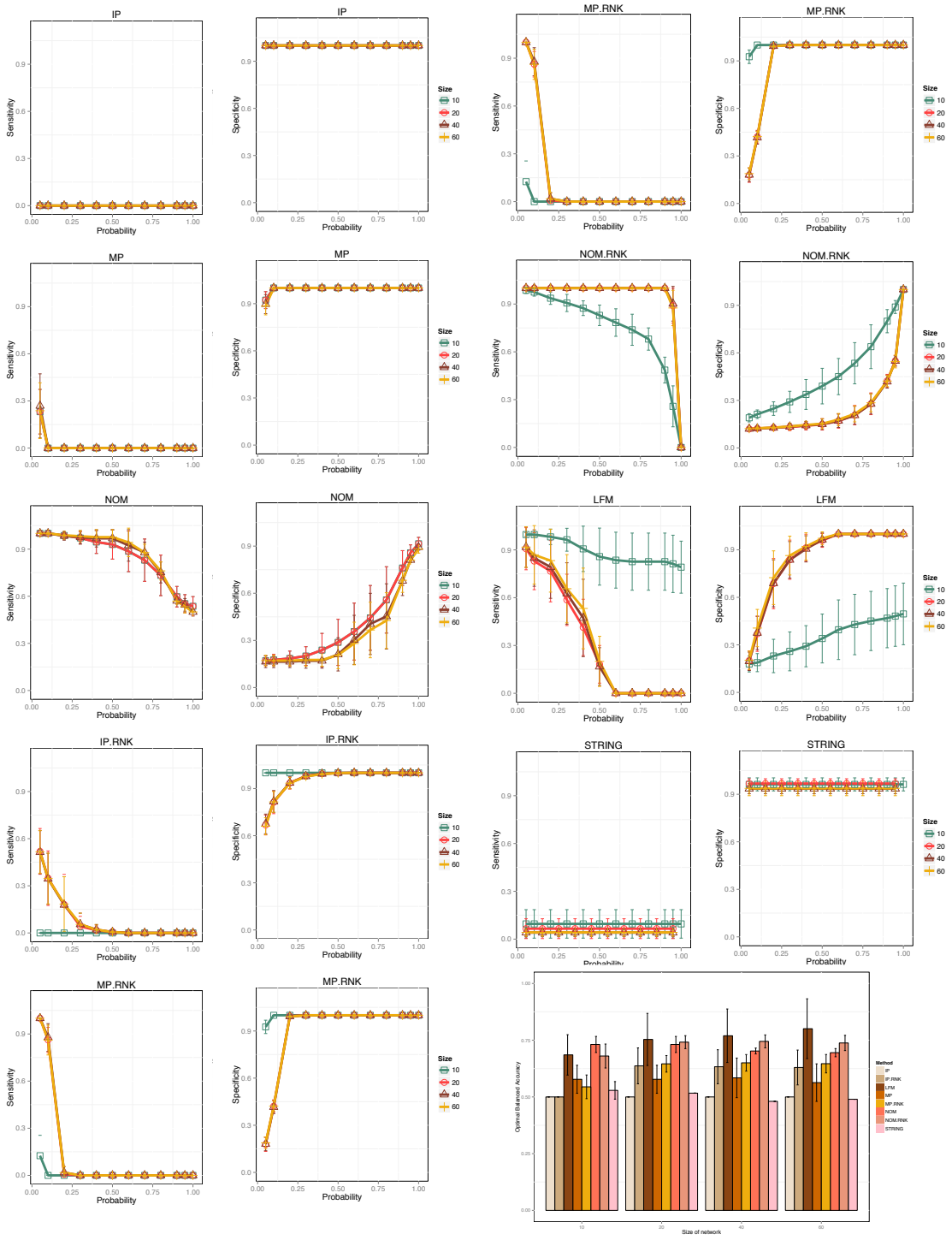

Figure 6.9.: Performance measurements for KEGG subgraph reconstruction based on real knowledge and comparison with STRING database. Plot showing the balanced accuracies of networks with varying number of nodes (20, 40 and 60) created just from different kinds of prior knowledge (bottom right)

the observed differences (Table 6.1). At the same time no statistically significant differences between NOM, LFM and NOM.RNK could be observed in terms of oBAC here. The IP prior revealed a oBAC which was almost constantly at 0.5. The reason for this behavior is that multiplicative nature of the IP method often yields numerically very small values, hence making IP close to a pure sparsity prior.

We also compared the reconstruction performance of our priors to a reconstruction with confidence scores from the STRING database (von Mering et al., 2003). The comparison showed a clear and significant advantage of our priors over the STRING in terms of higher oBAC (Figure 6.9, Table 6.1 and Tables D.1). Most methods showed a very low dependency on the network size, except for the LFM method, which improved the more nodes the network had. The reason for this behavior is that the LFM method essentially learns from the entries of the edge confidence matrices. Having larger matrices implies more information to learn from, hence the performance increases.

Table 6.1.: Pairwise Wilcoxon test for model performance comparison (false discovery rates) for $m = 60$. For $m = 10$, 20 and 40 see Appendix D.1.)

Methods	IP	IP.RNK	LFM	MP	MP.RNK	NOM	NOM.RNK
IP.RNK	0.0091	-	-	-	-	-	-
LFM	0.0036	0.0036	-	-	-	-	-
MP	0.2503	0.0249	0.0036	-	-	-	-
MP.RNK	0.0036	0.6953	0.0036	0.0036	-	-	-
NOM	0.0036	0.0433	0.0137	0.0036	0.0091	-	-
NOM.RNK	0.0036	0.0333	0.0182	0.0036	0.0137	0.3889	-
STRING	0.0036	0.0068	0.0036	0.1466	0.0036	0.0036	0.0036

6.7 Improving Network Reconstruction

6.7.1. Reconstruction via NEM

NEMs (see chapter 3 and 4) allow the use of a structure prior while network learning from perturbation data. Main idea of the inference framework by Markowetz et al.: A network hypothesis is a directed graph between S-genes. Attached to each S-gene are several E-genes. Knocking down a S-gene interrupts signal flow in the downstream pathway, and hence an effect of E-genes attached to itself and downstream genes is expected. The position of the E-genes is introduced as a model parameter $\Theta = \{\theta_i | \theta_j \in \{1, ..., n\}, i = 1, ..., m\}$, i.e $\theta_i = j$, if E-gene i is attached to S-gene j. The posterior probability of a network hypothesis ϕ is given as:

$$P(\phi|\mathcal{D}) \propto P(\mathcal{D}|\phi)P(\phi) \tag{6.15}$$

The equation above (Equation 6.15) allows one to specify a prior $P(\phi)$ on the network structure itself. This can be thought of as biasing the score of possible network hypotheses

towards prior knowledge.

Priors in NEM

NEMs (chapter 3, 4) allow specifying priors at two levels:

- S-gene prior: The $P(\phi)$ prior in the equation 6.15 in real sense of term refers to the S-gene prior. The S-gene prior describes the prior probability of connections among the S-genes as the name suggests. The S-gene prior represents the degree of belief can be very different for each edge in the network of dimensions $S \times S$ (3.7).

- E-gene prior: The E-gene prior provides a prior probability of association of the E-genes with corresponding S-genes. The principles of the prior are same as above while, the dimensions are $S \times E$ (giving a rectangular matrix). Each entry in the matrix is the probability of association of an E-gene to S-gene. This supports in selecting affected E-genes for each perturbation and hence mapping of E-genes to S-genes.

Model Selection

Using prior in network inference raises an important issue of an inferential dilemma, i.e. whether to believe prior assumptions/prior knowledge or the data. A pure data driven approach will not incorporate any biological background, whereas a complete prior or knowledge based assumptions do not give any new information and prevents learning. Practically this questions amounts to set the parameter ν in equation. 6.9 in the right manner. One way of dealing with it is to find the parameter ν that minimizes the Akaike information criterion (AIC) (Equation 6.16) reaches minima (Fröhlich et al., 2007a).

$$AIC(\nu, \phi_{opt}) = -2logP(\mathcal{D}|\phi_{opt}) + 2d(\nu, \phi_{opt}) \tag{6.16}$$

Here $d(\nu, \phi_{opt})$ denotes the number of free parameters (i.e. the number of unknown edges) in the network structure ϕ_{opt} optimizing. This approach is relevant for static NEMs, for the dynoNEMs the parameter ν in integrated into the MCMC strategy.

One can either use one of the priors (S-gene or E-gene) for NEMs or both together. Our simulation studies show the effect of these combinations of priors (compared to different methods for prior) on the network reconstruction accuracy for NEMs.

Simulations

To examine the effect of using prior knowledge in NEMs we first ran a set of simulations on static NEMs and dynoNEMs (MCMC). The aim was to evaluate the effect of different types of priors as discussed in the last chapter as well as to check the significance of improvement of results while using S-gene or E-gene priors (or both). Please note that the parameter ν is sampled itself from an exponential distribution in case of dynoNEM (MCMC), i.e. there is no parameter optimization.

Network Sampling

For these simulations the graph sampling was done in a biologically as most as possible realistic way. First, a S-gene graph was sampled using the method similar as in chapter 5. The E-gene attachment to S-genes was done here by using a biological graph. For this purpose a KEGG graph was constructed with 3776 nodes and 29878 edges by merging ∼80 KEGG graphs (See Appendix H). The S-genes were mapped onto this larger graph. These S-genes on the graph were considered as the seed nodes and then the genes attached downstream were extracted as E-genes via a random walk (downstream). For each network size to be tested via simulations we sampled 10 networks with this approach. A diffusion kernel (Kondor and Lafferty, 2002) was then computed for the S-gene $\times$ E-gene matrix using the pathClass R package (Johannes et al., 2011). The diffusion kernel matrix actually gives the amount of diffused information from one node to the other in a graph. Thus, it can consider all alternative paths between two nodes. The matrix value is interpreted as the probability for attaching E-genes to S-genes after normalizing the row sum to 1 (Please remember that in chapter 4 a uniform probability was used to sample the E-gene attachment).

Data simulation

The effects on E-genes (sampled from ~ 3800 genes as explained above) were simulated via a BUM (Beta Uniform Mixture) model (see chapter 5). The BUM model was generated as per the effects observed on E-genes attached from real biological pathway.

Reconstruction

For simulations following conditions were considered.

- NEM without prior (NP)
- NEM with different priors e.g. sparsity prior (Sparse), IP, MP, NOM based S-gene prior (NOM), LFM, IP.RNK, MP.RNK and NOM.RNK based S-gene prior.

Furthermore we also looked at the effect of adding the E-gene prior to the best results we obtain in the simulations discussed above. This includes the following sets:

- NEM only with S-gene prior
- NEM only with E-gene prior
- NEM with both S and E-gene priors

Sparsity prior

Each single gene is controlled by a limited number of other genes, which is small compared to the total gene content (Bailly-Bechet et al., 2010). This is the governing principle for sparsity priors. It tends the edges to have a minimum prior probability to bias the networks towards sparse (very few edges) topology. In the simulations performed here the sparsity prior is a matrix of 0's

Each of these E-gene priors were generated using the NOM model as they are sufficiently fast to compute for large data sets and at the same time provide sufficiently good results as observed in last chapter. However, for S-genes all the sets of priors discussed in last chapter (IP, IP.RNK, MP, MP.RNK, NOM, NOM.RNK and LFM) were used. We compared our results against a completely data driven approach i.e. without any prior.

Static NEMs

Static NEMs are used to learn network from static perturbation data (see chapter 3). First the effect of above mentioned priors were investigated on the NEMs. The simulation method used was similar as explained before. The parameters affecting the use of priors were analyzed as follows:

A. Dependency on number of S-genes

Larger networks are usually more difficult to learn from the data than smaller ones. It is expected that for larger networks (higher number of S-genes) priors can be very useful.

To undertand the effect of number of S-gene (i.e. the number of knockdowns) in NEM learning with prior, a set of networks with number of S-genes=(5, 10, 15 and 20) were generated. The set of 11 priors discussed in section above were compared against each other and the solely data driven approach (NP) (Figure 6.10). The results showed the smaller network (n=5) have better reconstruction accuracy even without prior (Sensitivity $\sim$ 80%, Specificity $\sim$ 90%). The sparsity prior and the IP proved to hamper the sensitivity but maintained good specificity. As the network became larger the sensitivity reduced for all the priors. The specificity for smaller network without prior was $\sim$90% (n=5) and it reduced to $\sim$65% (n=20) (all cases for 50 E-genes sampled by the method explained above i.e. random walk along the merged KEGG graph).

Observing the behavior of different priors was as per the expectation. IP and sparsity priors showed high specificity and low sensitivity. NOM based methods were better in terms of sensitivity but poor in specificity as seen before in our early simulations. Nevertheless, the LFM maintained good sensitivity and specificity. Ranked methods performed better than their non ranked counterparts. In general for all the sizes the priors seemed to improve the reconstruction accuracies.

B. Dependency on number of E-genes

During our simulation we also varied the number of E-genes for each of the S-gene network (n=$\{$5, 10,15 and 20$\}$). Availability of fewer genes is less informative for network inference via NEM, whereas more E-genes can lead to a more accurate network inference in the absence of a prior. A biological prior can be extremely helpful to learn a network.

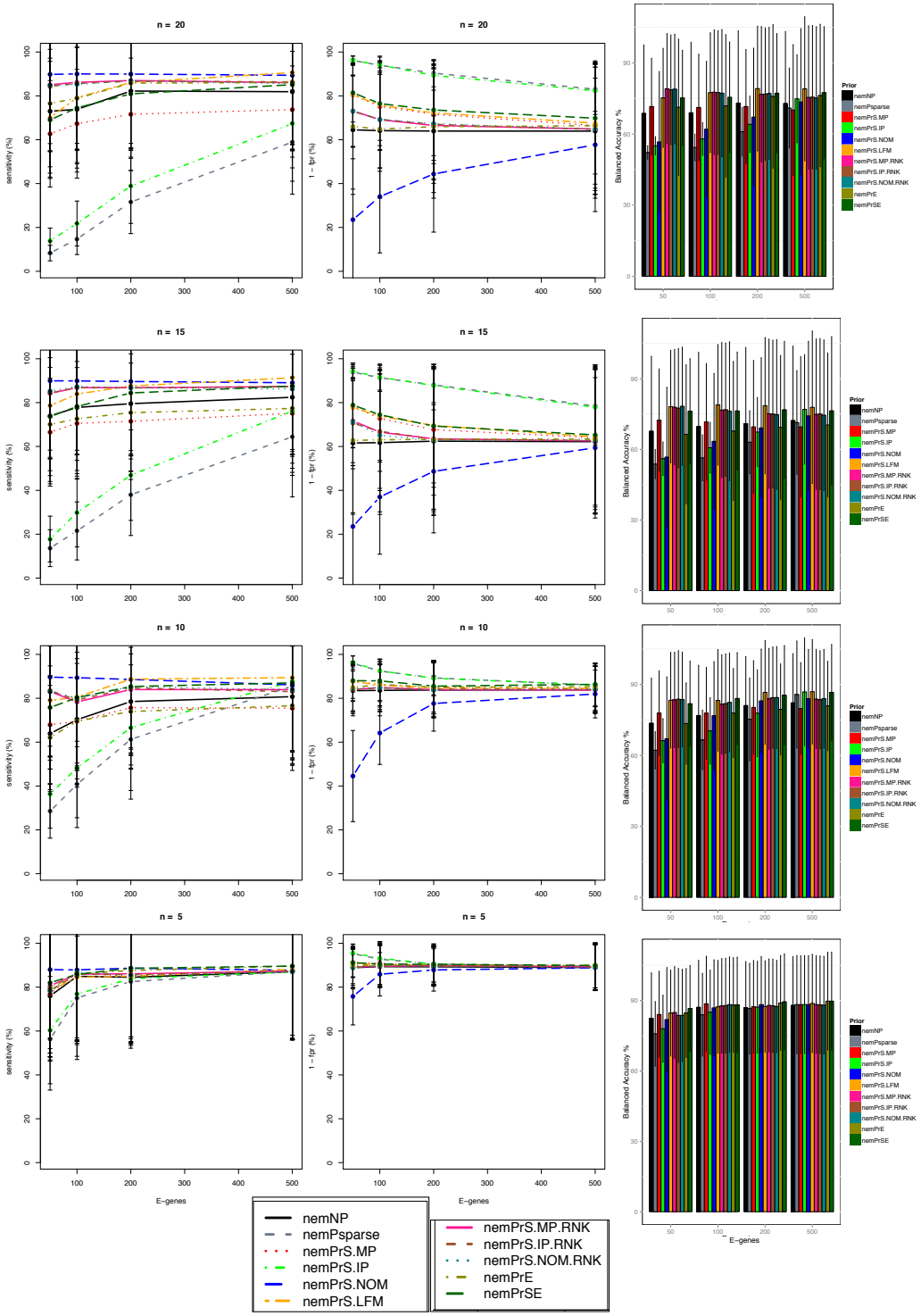

Figure 6.10.: Reconstruction of network via NEM; showing the effect of number of $S - genes$ and $E - genes$. The plots show sensitivity, specificity $(1 - fpr)$ and balanced accuracy in order from left to right

In our simulation we ran NEM with different set of E-genes (50, 100, 200 and 500) to analyze the above mentioned dependency. As anticipated with fewer E-genes the completely data driven reconstruction returned poor results. Use of prior (specially non-sparsity and non IP) priors improved the reconstruction even with smaller number of E-genes. For example the reconstruction of a network (n=10, E=100) without prior had a sensitivity of $\sim$ 70% and specificity of $\sim$ 85. Using an LFM prior improved the sensitivity to $\sim$80% and maintaining similar specificity at the same time. Thus the trend demonstrated an increase in sensitivity and decrease in specificity with increasing E-genes. The reconstruction without any prior needed more E-gene information for higher sensitivity, whereas, similar sensitivities could be attained with the use of prior even with very low number of E-genes (Figure 6.10). Overall, adding more E-genes increases the balanced accuracy however with sufficiently high number of E-genes, the positive effect of prior saturates.

C. Dependency on Prior sets

Our next objective was to sudy the effect of using S-gene priors and E-gene priors individually or together. For this we used the LFM based S-gene prior and NOM based E-gene prior. The reason was that in our current implementation the LFM method becomes prohibitively slow for larger numbers of genes.

The use of S-gene and E-gene priors at the sensitivity level is very much comparable to the reconstruction without prior, but a clear advantage of these priors in established in terms of specificities. The use of combined prior knowledge (S-gene and E-genes) increased the sensitivity by a difference of about 20% for a data with 50 E-genes. However, at individual level S-gene priors were found to be more effective than the standalone E-gene priors. This is explained by the nature of NEM where, S-gene prior actually maps the network while E-gene prior contributes to association probability of E-genes to S-genes. Nevertheless, the combined prior brought a small improvement (2-8%) compared to the standalone S-gene priors.

First of all a comparison was made among different possible S-gene priors for NEM as discussed above. Sparse priors tend to make the networks carry low number of edges. This is evident in the results as the networks reconstructed with sparse prior have low sensitivity and high specificity (Figure 6.10.

In the above described simulations different methods were used to generate prior (namely: IP, IP.RNK, MP, MP.RNK, NOM, NOM.RNK and LFM). The ranked priors plus the NOM and LFM outperformed the non ranked versions. The sparsity prior as expected showed very high specificity and low sensitivity and was matched only by the IP method. The reason for this behavior is that the IP prior with an increasing number of information sources tends to lead to values close to 0, as discussed above. The cases where we do not use a prior was although acceptable but was outflanked by most of the priors except for $sparsityprior$ and IP. The LFM showed the best specificity for larger networks as observed before. The other methods showed performance trends similar to each other (Figure 6.10).. In terms of the balanced accuracy the NOM matched the ranked methods that performed at par with the LFM method.

Dynamic NEMs (dynoNEM)

Dynamic NEMs (dynoNEM) are used to learn network from dynamic perturbation data (see chapter 4). The simulation and learning was followed in the same fashion as static NEMs. For the purpose of studies here the MCMC approach was used with burnin = 50000 and 100000 samples. As we have mentioned before the scale parameter ν trades the degree of influence of a prior against the likelihood of the model. A large ν reduces the influence of the prior, whereas a smaller ν increases it. The parameter ν is included in the sampling process for the MCMC based dynoNEM. Thus, it will exploit this mechanism to handle the prior while learning a network from data and prior.

The method used for generating prior here was only NOM for E-gene priors, being computationally faster as we did in case of our simulations with static NEMs. Nevertheless, for the S-gene priors all the defined methods were used The results were analyzed based on the following factors:

A. Dependency on number of S-genes

The use of different priors in simulations with dynoNEMs showed similar improvements in results as in case of the static NEMs. The inclusion of a prior based on the existing knowledge brought in better network reconstruction accuracy compared to the inference with any prior (NP) or the sparsity priors (Figure 6.11). It was observed that sparsity prior as seen before showed high specificity but low sensitivity. The ranked versions were found to outperform the non-ranked version of priors. The LFM and NOM was at par with the best methods. The ranked methods (MP.RNK, IP.RNK and NOM.RNK) also performed better than the NP and sparsity prior.

B. Dependency on number of E-genes

The performance trend observed for the E-genes was again similar to that in the static NEMs. Increasing the number of E-genes decreased the effect of any of the priors. This was more in case of sensitivity and less when we observed the specificity. In terms of specificity most of the priors performed at par with each other except NOM. We have seen such behavior of NOM based prior during our initial simulations. Overall, at the level of balanced accuracy the LFM seemed to provide minor advantages over NOM and ranked priors in case of large networks. However, this advantage was lost for smaller networks as they were easy to construct even without prior.

C. Dependency on number of time points

As shown before (chapter 4) dynoNEM algorithm shows better reconstruction with more time points (Figure 6.12). Prior from biological knowledge can be used to possibly compensate for lower number of time points. In order to explore this we ran a simulation of dynoNEMs with 15 nodes and three different time points 3, 5 and 10. The priors are expected to yield

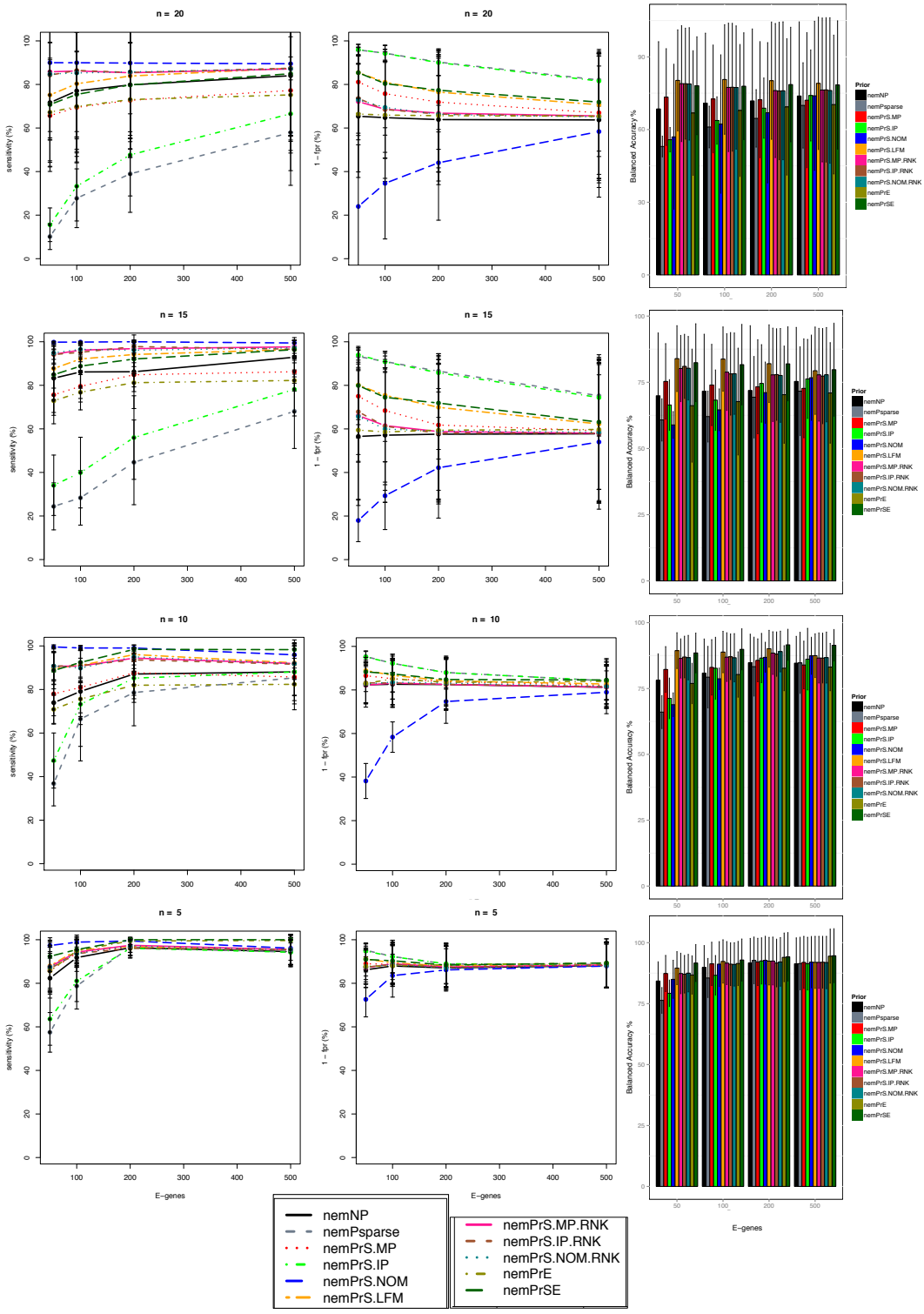

Figure 6.11.: Reconstruction of network via dynoNEM; showing the effect of number of $S-genes$ and $E-genes$. The plots show sensitivity, specificity $(1-fpr)$ and balanced accuracy in order from left to right

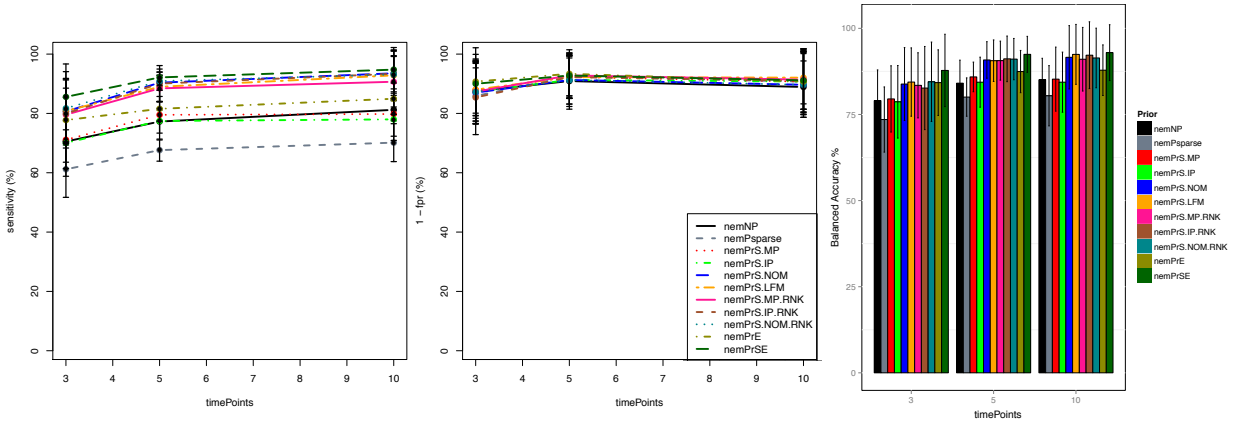

Figure 6.12.: Reconstruction of network via dynoNEM; showing the effect number of time point measurements in data. The plots show sensitivity, specificity $(1 - fpr)$ and balanced accuracy in order from left to right

better dynoNEM reconstruction accuracy as compared to the reconstruction with prior even for lower number of time points.

The results of the simulation showed that all the priors except sparsity prior and IP improved the sensitivity. The reconstruction without prior showed a sensitivity of ∼80% for 10 time points, but introducing a prior like LFM improved the sensitivity up to a range of ∼90%. Nevertheless, this improvement was found to depend on the prior used. The only exception to this was IP and the sparsity priors. This was again because of the nature of these priors, i.e. numerically very small edge confidences. The ranked priors as well as the LFM outperformed the other priors. The NOM prior also performed well and comparable to the ranked priors and LFM. But the results on non ranked MP and IP did not show a great advantage over other priors. Furthermore, the trend observed along the number of time points was same as observed in the dynoNEM chapter (chapter 4).

In terms of specificity the performance difference was not as clear as in case of sensitivity. However, the IP and sparse prior showed the best specificity, though their overall balanced accuracy was poor. Furthermore, all the other priors showed comparable balanced accuracy.

D. Dependency on time lag

In our simulation for dynoNEMs we have seen that for more uniformly distributed time lags the reconstruction shows lower sensitivity. The objective of this simulation was to check if the use of priors compensate for the time lag dependency. It is expected that the use of prior knowledge will improve the sensitivity of reconstruction even for low values of parameter and for higher values the gain caused by the use of prior will diminish. The reconstruction specificity of dynoNEM has already been shown to be high (chapter 4). Hence a big leap in specificity performance is not expected

In order to investigate this we ran a simulation for 10 networks of n = 15 (number of S-genes) with parameters 0.3, 0.5 and 0.8. The number of E-genes attached to the network were kept 200. The results did not demonstrate to overcome the time lag dependency. There appears to be an increasing sensitivity when using priors, but this was found to be true for all time lags. The sensitivity at the parameter value 0.3 already showed a substantial improvement for ranked priors and LFM (Figure 6.13).

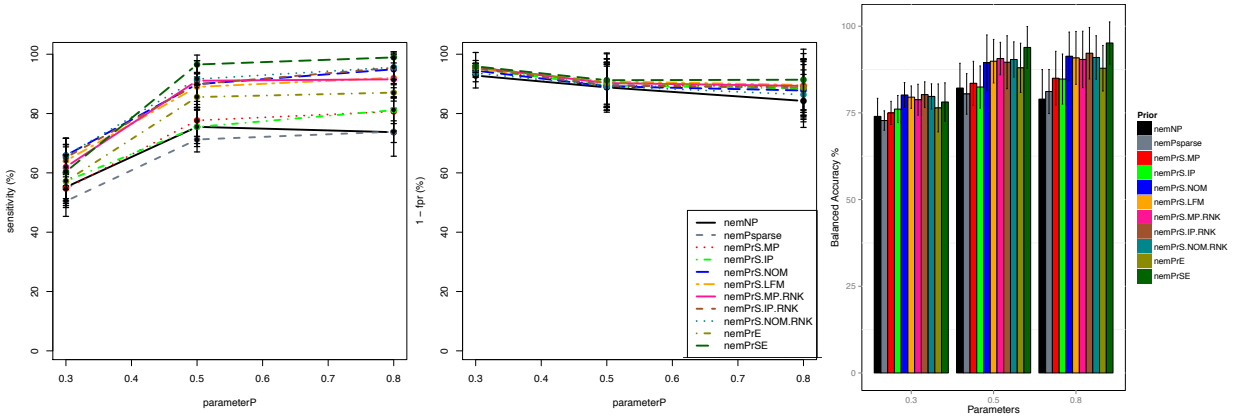

Figure 6.13.: Reconstruction of network via dynoNEM; showing the effect of exponential distribution parameter defining the time lag. The plots show sensitivity, specificity $(1 - fpr)$ and balanced accuracy in order from left to right.

6.8 Application

6.8.1. dynoNEM Application: Murine Stem Cell Network

We defined the dynoNEM model in chapter 4 with an application on time course data from murine stem cells to reconstruct network for 6 genes (Please see chapter 4 to know more about the data). To check the effect prior in real world with NEM models we extracted the prior knowledge for these 6 genes and ran the dynoNEM method with this prior. The results showed to have brought improvement over our sparse prior method that we used in chapter 4. For the purpose we used the Noisy-OR prior (S-gene prior) as during our simulations it showed to have very good results. Moreover the murine stem cell network is a small network of 6 S-genes, and as we have seen LFM is not the most obvious choice for small networks. Furthermore, the MCMC based dynoNEM was used with the MCMC chain starting with the greedy hill climber output for faster convergence. The number of burnins was set to 50000 and samples to 100000.

The reconstructed network (Figure 6.14 a) showed better results compared to the results without the prior. The inferred network had higher number of edges and most of it could be explained using the literature network obtained from MetaCore^{TM}(Figure 6.14 (b)). From the

other side the model could explain the literature network better that the one inferred without the use of a prior. Looking at the literature network and the inferred network presented the fact that most of the paths described by the model are explained by the literature (Figure 6.14 (d) and (e)).

The results shows the inference with prior actually retrieved more edges that the one without it (10 compared to 7). Out of these 10 edges 9 could be explained by the literature compared to 6 out of 7 in the inference without using a prior. From the point of view of the literature based network (Figure 6.14 (d)), only one extra edge was explained by the literature. This is not surprising as the literature network contained additional interactions, which could not be observed in our estimated networks.

As more comprehensive analysis we tried to reconstruct the network with different set of priors. It was observed that reconstruction with non ranked priors especially MP, IP was as good as the one with a $sparsityprior$. NOM and NOM.RNK was found to perform well and had a slight edge over the other methods. This was the reason we finally selected our network with NOM prior. The LFM prior based network was acceptable and better than the non ranked priors, however for such small network we have observed the drawback with LFM at theoretical as well as practical level.

6.9 Summary

We developed and proposed two new methods to integrate different sources of biological information as quantitative prior knowledge for network learning in biological network reconstruction via NEM. Our approach is based on the assumption of relatedness in biological data, which is biologically realistic. Furthermore, our approach takes into consideration diverse sources like database information, ontological information, literature as well as protein domain data. The current approach allows not only to include existing heterogeneous information sources but is flexible enough to include new ones. With the simulations and validation we reached a better performance. As the database for the biological information and annotation grows, larger number of correlated information can be compiled into prior knowledge ultimately leading to far more realistic models from experimental data in biology.

With the above set of experiments it was shown the use of biological prior improved the network reconstruction from data. Furthermore, the use of prior also facilitated better network inference even with smaller sample size. As databases for biological information and annotation grow, a larger amount of correlated information can be compiled into prior knowledge, which ultimately can be utilized to more realistic probabilistic model inference from experimental data. We showed the improvement achieved with static and dynoNEMs while using prior knowledge. The next chapter will present a novel example in Non Small Cell Lung Cancer cells for 20 genes and the associated prior knowledge via the use static NEMs and priors.

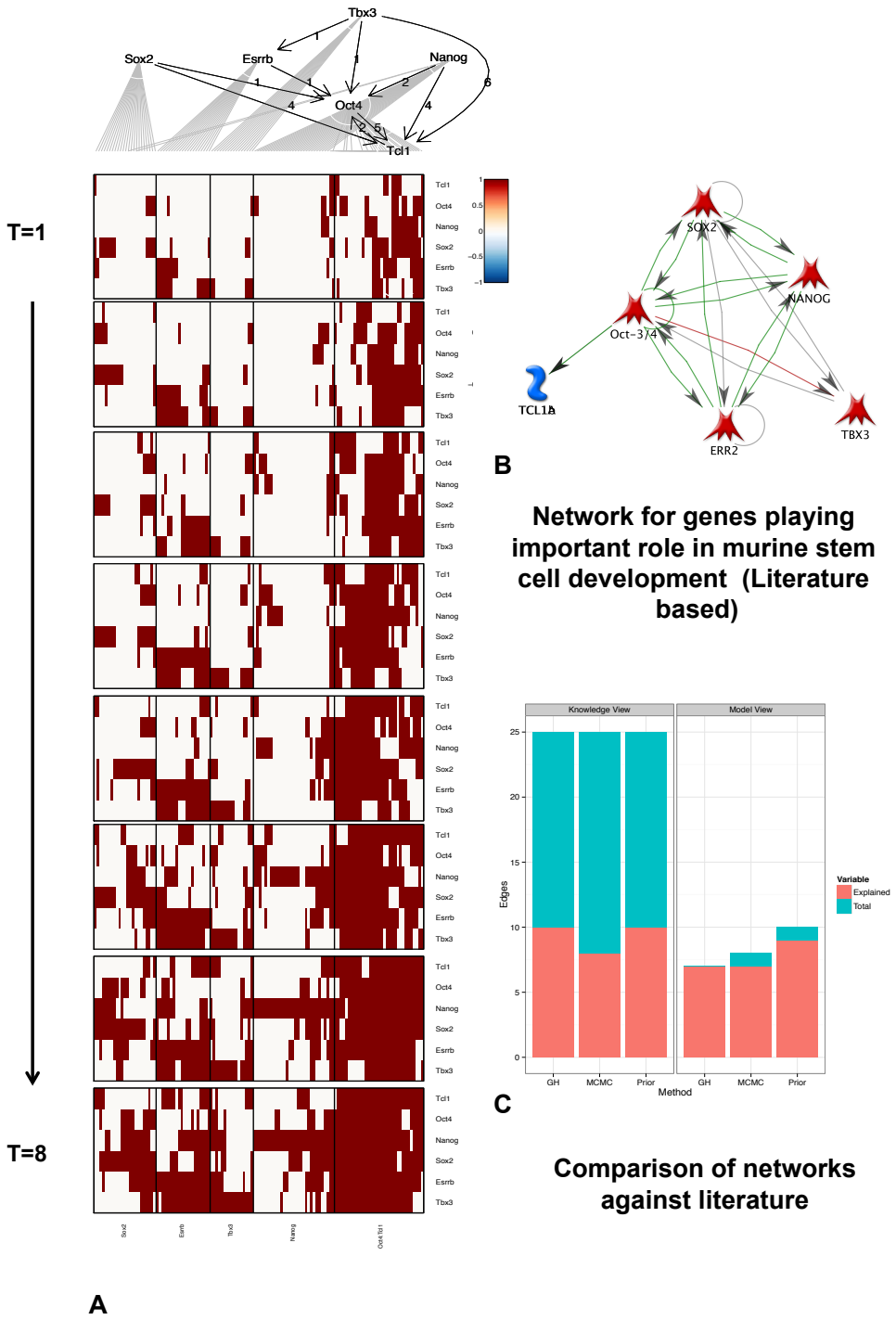

Figure 6.14.: Applying priors in Murine Stem Cell Data to reconstruct network via dynoNEM and prior knowledge. For the network reconstructed without using a prior (only data) see figure 4.12

Chapter 7
Reconstructing EGFR/AMPK Signaling in NSCLC

The thesis so far presented a detailed description of Nested Effects Model and its extensions. The last chapter presented few methods to combine different sources of knowledge as network prior. A simulation experiment was also presented in the last chapter depicting the use of prior in inferring networks using NEM followed by an application of dynoNEM to reconstruct parts of a network known to be involved into murine stem cell development. The current chapter will present an application of NEM on inferring a network of different proteins that have been experimentally found to play a role into drug sensitivity in Non Small Cell Lung Cancer (NSCLC). The major outcome of the analysis is, besides the network structure itself, a putative novel drug target, namely AMPK. Activation of this target in followup experiments leads to a significant down-regulation of a number of cell proliferation markers. The work has been submitted to a peer review journal for publication (Appendix M).

7.1 Lung Cancer

Lung cancer, also known as carcinoma of the lung, refers to the uncontrolled cell growth in tissues of the lung. Being the second prevalent cancer in both men and women, lung cancer accounts for 12% of all new cases of cancers reported worldwide (van Meerbeeck et al., 2011). The global scenario (Figure 7.1) has shown lung cancer as the leading cause of cancer related death worldwide. It caused about 1.5 million deaths globally in 2010 (Lancet, 2013). The symptoms of lung cancer include cough, chest discomfort, weight loss, and sometimes hemoptysis. However, many patients with metastatic disease do not exhibit any clinical symptoms (Robert S. Porter, 2009) [1]. About 85% of lung cancer cases are related to smoking cigarette [2].

It is classified into two major categories; (1) Small cell lung cancer (SCLC) and (2) Non-Small cell lung cancer (NSCLC)

[1] http://www.merckmanuals.com/professional/pulmonary_disorders/tumors_of_the_lungs/lung_carcinoma.html ; Accessed: January 2014

[2] http://www.cancer.gov/cancertopics/pdq/prevention/lung/HealthProfessional/page2 ; Accessed: January 2014

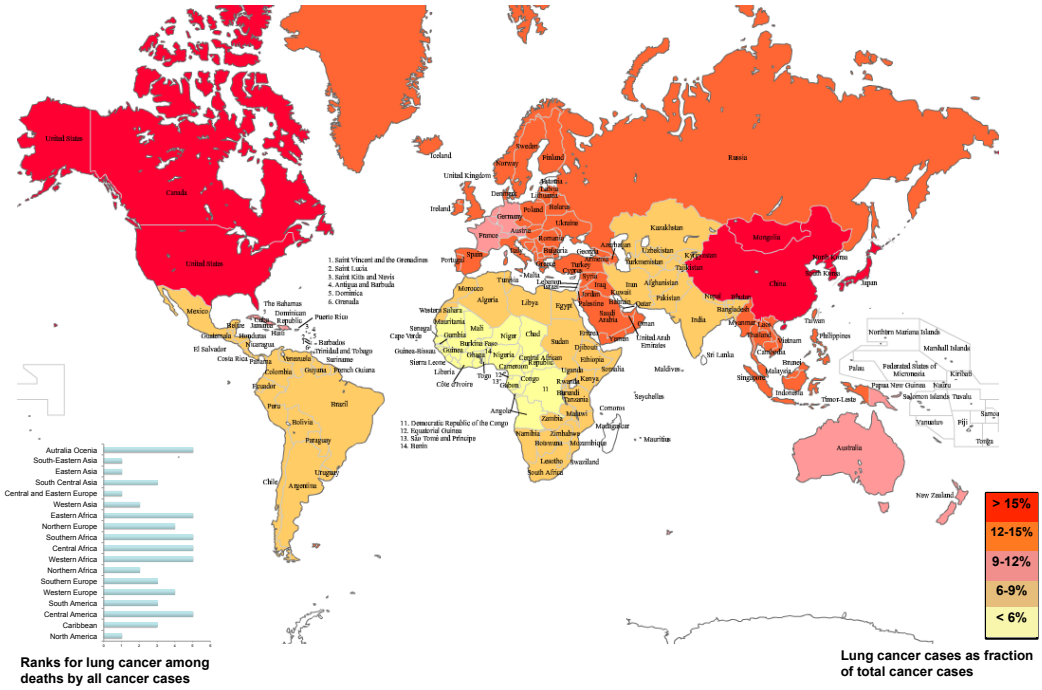

Figure 7.1.: Lung cancer cases around the world . Shows the geographical distribution as well as the rank of lung cancer among different types of cancers, in terms of fatality. The figure is based on data from: Cancer Research UK

7.1.1. Small Cell Lung Cancer

SCLC; also termed as oat cell cancer, is a malignant epithelial tumor consisting of small cells with scant cytoplasm. It typically involves only small cells (Junker and Petersen, 2009). It accounts for about 15% of the cases reported with lung cancer. It usually starts in the bronchi close to the center of the chest and tends to grow and spread quickly. The tumor can spread to distant parts of the body even the brain, before being diagnosed (D'Angelo and Pietanza, 2010).

7.1.2. Non-Small Cell Lung Cancer

NSCLC is a disease in which malignant (cancer) cells are formed in the lung tissues. About 85% of lung cancers are NSCLC (Reck et al., 2013). The clinical behavior of NSCLC is more diverse depending on histologic type. However, $\sim$ 40% of patients have metastatic disease outside the chest at the time of diagnosis (Robert S. Porter, 2009; Goldstraw et al., 2011). Driver mutations have been identified primarily in adenocarcinoma, although attempts are being made to identify similar mutations in squamous cell carcinoma (RS et al., 2012).

NSCLC can be categorized into following major sub groups based on chemical makeup, size, and shape when viewed under microscope.

Squamous cell carcinoma: About 25% to 30% of all lung cancers are squamous cell carcinomas (SCC) [3]. SCCs start in flat squamous cells that form the inner layer of the airways in the lungs. They are often linked to smoking and often found in the middle of the lungs, near bronchus (Roth et al., 2011).

Adenocarcinoma: About 40% of lung cancers are adenocarcinomas. They start in squamous cells that secrete mucus like substance. This type of lung cancer also occurs mainly in current or former smokers, nevertheless, it is also the most common type of lung cancer seen in non-smokers. It is more common in women and it is more likely to occur in younger people than other types of lung cancer. It is usually found in outer parts of the lung. The tumor tends to grow slower than other types of lung cancer, and is more likely to be found before it has spread outside the lung.

In the work presented here we focus on the NSCLC (adenomcarcinoma). The data that is presented here resulted from a cooperation with a group led by Prof. Holger Sültmann at the National Center for Tumor Disease (NCT) in Heidelberg. All experiments were carried out in his group based on H1650 cells.

7.2 Motivation

Lung cancer is a highly heterogeneous malignancy. It is obligatory to know about the etiology and pathogenesis of the disease in order to control it. Understanding the activity of signaling molecules affecting cell growth and survival in lung cancer can serve as the key to future therapy work flow. The identification of a complex containing the tumor suppressor LKB1 as the critical upstream kinase required for the activation of AMP-activated protein kinase (AMPK) by metabolic stress has been reported representing the first clear link between AMPK and cancer (Hardie and Alessi, 2013; Hawley et al., 2003). Furthermore, the tumor genome sequencing data can be used to study frequent mutations in NSCLC (Thomas. et al., 2013). Genes like TP53, EGFR, STK11 (LKB1) etc. can have somatic mutations that affect cancer driving signaling pathways together with cellular processes representing putative drug target sites (Imielinski et al., 2012).

Epidermal growth factor receptor (EGFR) is one of the therapeutic targets for NSCLC. EGFR is a protein found on the cell surface and it helps the cells to grow and divide. Some NSCLC cells have too much EGFR, which causes them to grow faster. Several drugs e.g. Erlotinib, Cetuximab and Afatinab target EGFR in order to control tumor. Many of these therapies induce an initial positive response. Nevertheless, in many cases, the tumors turn insensitive to these therapies and evolve more aggressive and resistant phenotypes (Holohan et al., 2013; Mumenthaler et al., 2011). For example, Erlotinib turns off the EGFR pathway leading to tumor cell death but in many cases the therapeutic benefits lower due to EGFR mutation to a drug resistant state (Chong and Jänne, 2013). To overcome such treatment failures new target need to be discovered. These challenges posed by drug resistance in cancer can be met via

[3] http://www.cancer.org/cancer/lungcancer-non-smallcell/detailedguide/non-small-cell-lung-cancer-what-is-non-small-cell-lung-cancer; Accessed: January 2014

a polypharmacogenic approach (Innocenti, 2012; Al-Lazikani et al., 2012), by studying the combinatorial perturbation in-silico.

Nested Effects Model (NEM) can be useful for such studies with an aim to understand signaling pathway as well as effects of perturbation. Thus it makes an ideal scenario for NEM application. We focus here is on the AMPK/EGFR pathway and its subunits. The traditional paradigm for drug discovery in light of growing difficulties like insensitivity of EGFR based treatment and resistance development needs new drugs that can potentially slow the progression of NSCLC. Therefore, we exploit the inferred network to identify potential drug targets for therapeutic purposes in NSCLC. The choice of these subunits under study stems from prior, unpublished experiments carried out in the lab of Prof. Sültmann at NCT in Heidelberg (Germany), which demonstrated a link of these proteins to Erlotinib sensitivity.

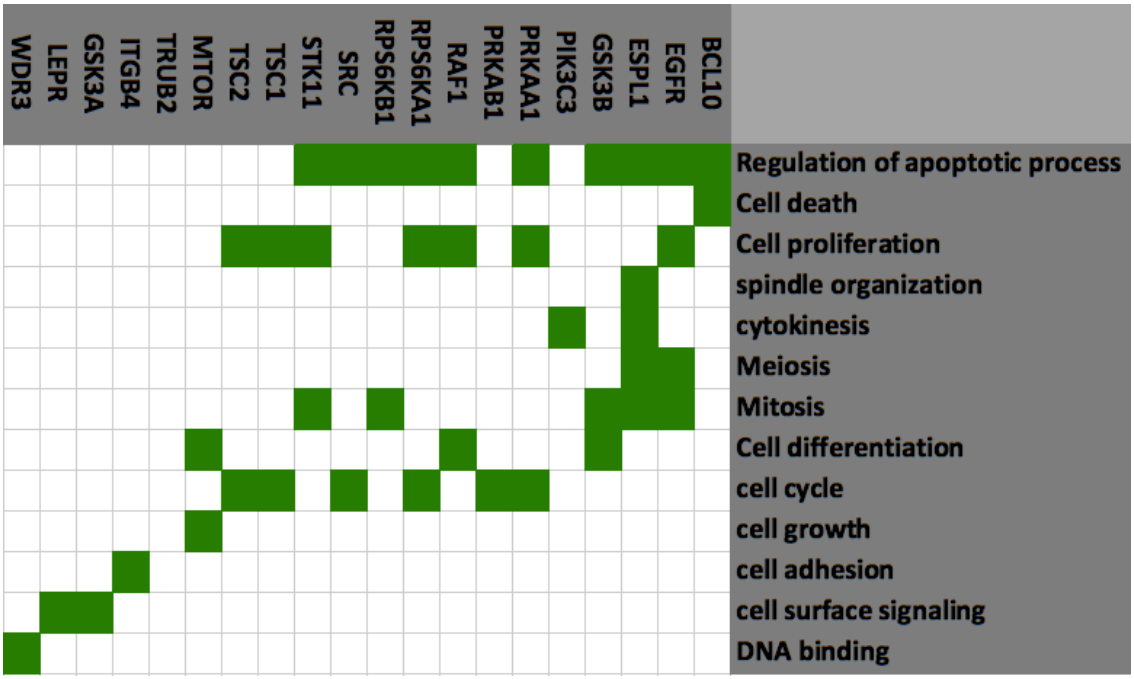

Figure 7.2.: List of genes selected for perturbation and summarised GO terms (focus on cell division, proliferation etc.). Green cells indicate annotation i.e. corresponding genes (columns) are annotated with the term (rows). A detailed list of the GO categories for these genes is available in appendix I.1

7.3 Data

7.3.1. Expression data

As mentioned in the last section the idea behind these experiments is to understand more about the AMPK-signaling activity and its influence in EGFR-based therapy resistance in lung cancer. A set of genes (Figure 7.2) for the knockdown experiments was selected with regard to their known or putative interaction with AMPK and EGFR or role in drug sensitivity. In total 20 genes were selected (See appendix I.1). Knockdowns for these selected genes via siRNA were conducted and the subsequent effects of downstream genes and proteins measured via gene expression microarray experiments. The knockdown-efficiency was proven before the array experiment with a level $> 70\%$ transcript reduction. The expression measurements were done with series in H1650 cells comprising 20 gene knockdowns and controls. All experiments were done in triplicates. The array platform used was Illumina Whole-Genome Gene Expression Bead Array Chips (Chen et al., 2012). It consists of oligonucleotides immobilized to beads held in micro wells on the surface of an array substrate. Further, the data was preprocessed by the LUMI software and quantile normalized.

BOX 7.1: Reverse Phase Protein Array (RPPA)

Reverse Phase Protein Arrays (RPPA) have been termed as direct descendent of miniaturized immunoassays (Mueller et al., 2010). Reverse phase protein microarrays are also referred to as 'protein microarrays', 'lysate arrays', and 'tissue lysate arrays', but not all protein microarrays are reverse phase microarrays. There exist three classes of protein microarrays (1) forward phase, (2) sandwich, and (3) reverse phase. A Forward phase (antibody array) is the one where, multiple antibodies are immobilized onto a surface to capture proteins from a sample. Sandwich arrays use a pair of antibodies to capture and detect the protein of interest. Each antibody in a sandwich assay detects unique epitopes of the same analyte in the sample. The reverse phase format consists of immobilizing the analyte protein on a surface and probing the array with a single antibody directed against the analyte of interest. RPPA has the advantage of higher throughput, higher sensitivity and better dynamic range compared to other protein measurement methods like western blot.

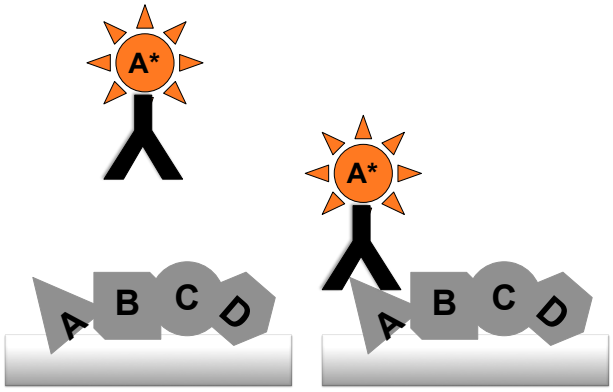

Figure 7.3.: Reverese Phase Protein Array

7.3.2. Protein array data

Furthermore, reverse-phase protein array technology RPPA (protein quantification; see BOX 7.1) via multiplex immunoassays (LUMINEX) and Western blotting was used to quantify the expression of 107 proteins in cancer relevant pathways (e.g. EGFR, MAPK, JAK/STAT, PI3K/AKT) in the same lung cancer cell line. 23 (14 unique) of these were the S-genes. These cells were processed to inhibit or activate AMPK protein by drugs in different concentrations (low, medium, high). Together with this, we also had knockdowns for AMPK subunits: PRKAA1 and PRKAB1 as well as EGFR, MTOR and RPS6KB by siRNA approach. Further annotations were given if the antibody recognized native or phosphorylated protein.

With this data, it would be very useful to apply the Nested Effects Model in order to get more insights about downstream effects and putative hierarchy of these genes in the AMPK signaling. More specifically, we here used the gene expression data for learning the network model and the RPPA data for independent validation.

7.4 Data Analysis

The expression data was processed and analyzed for differentially expressed genes for each perturbation using the *limma* package. This set of differentially expressed genes will serve as the E-genes for the NEM algorithm.

7.4.1. Data processing

The $lumi$ package (Du et al., 2008) from Bioconductor [4] was used to read in and prepare the data for analysis.

As mentioned before, the data has been quantile normalized. The data was then checked for its quality via diagnostic plots using the $arrayQualityMetrics$ package (Kauffmann et al., 2009) from Bioconductor. This created diagnostic plots like boxplots, MA plots, heatmap etc. The signs of batch effect were evident in these plots (Figure 7.4 (A)). Please recall that the data generation experiment was performed in two batches. To get rid of the batch effect we used the ComBat function from sva package (Johnson et al., 2007). The ComBat function uses an Empirical Bayes method to combine the batches. It estimates parameters for location and scale adjustment (LS) of each batch for each gene independently. Pooling information for multiple genes in each batch brings out the pattern that genes with similar expression follow. This information is then used to adjust the batches in order to eliminate the batch effect. Thus the batch effects were removed (Figure 7.4 (B)).

7.4.2. Selecting E-genes

Selection of E-genes is a critical process in order to get the best results from NEM. As mentioned before we selected differentially expressed genes (probes) for each perturbation in order to filter in the E-genes. This assured the selection of the probes that are significantly affected by corresponding perturbation. The $limma$ package (Smyth, 2004) from Bioconductor

[4] http://www.bioconductor.org/ ; Accessed: December, 2013

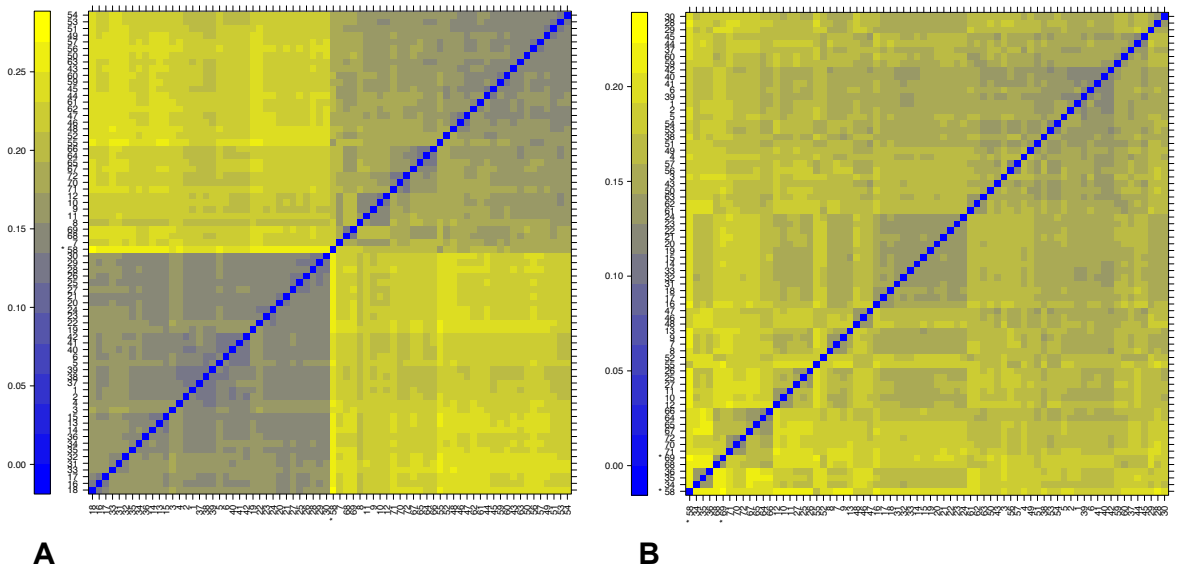

Figure 7.4.: Batch effect in the expression data. (A) Heat map for the data showing batch effect (2 batches) (B) Data after removing batch effect

was used to select the differentially expressed genes for each perturbation. $limma$ stands for Linear Models for Microarray Data. It implements several methods for linear modeling of microarray data that can be used to identify differentially expressed genes. After fitting a linear model for each probe in the data, it uses an empirical Bayes method for assessing differential expression. It computes statistical test and returns the p-values and log fold change for each gene. We selected a dual condition of p-values < 0.05 and log fold change > 1.5. This was repeated for all the perturbations. We could get 388 probes from all perturbations to be used E-genes.

7.4.3. BUM model fitting

Based on the p-values obtained after analyzing the expression we data we used a Beta-Uniform Mixture (BUM) model fit to get the p-value density for all perturbations. This is same as explained in chapter 4. An example BUM model fit (Histogram and QQ-plot) for ESPL1 has been shown in figure 7.5. The p-value density is a matrix of the dimension E-genes $\times$ S-genes. This matrix will serve as the input to NEM algorithm. The results showed that some genes (e.g. PRKAA1 and SRC) have highly pronounced effects and some other (e.g. GSK3A), showing minimum effect of perturbation (Figure 7.6).

7.5 Applying NEM

Once our input data was ready in terms of log p value density from the mRNA expression data, we applied the NEM algorithm to it.

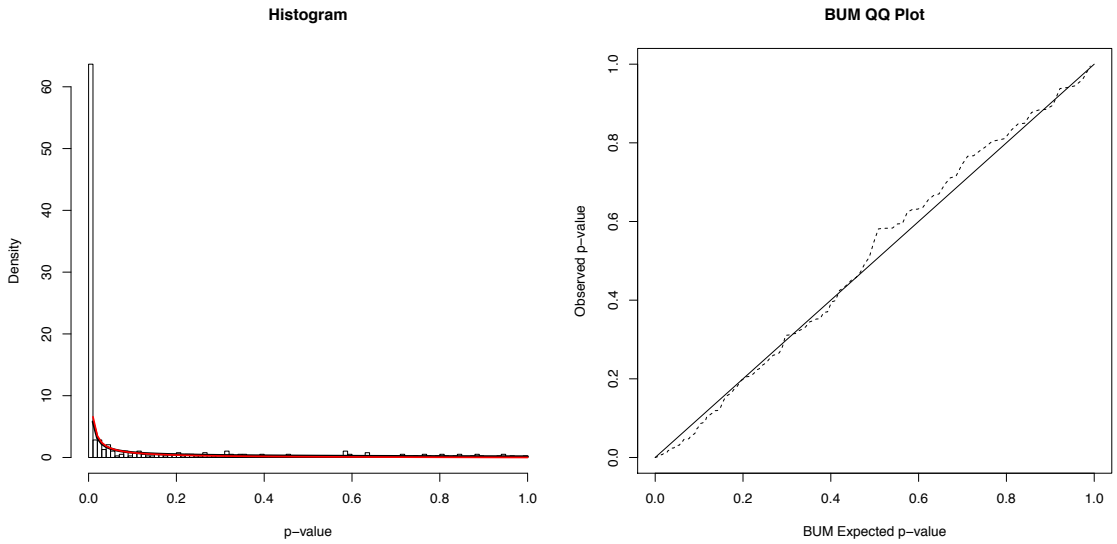

Figure 7.5.: BUM fit for ESPL1 perturbation: histogram and QQ-plot. In the histogram black line represents the mixture model curve and red: the extracted alternative distribution.

7.5.1. Running NEM with different S-gene priors

First of all we ran the NEM with all the priors for S-genes described in previous chapter (IP, MP, NOM, IP.RNK, MP.RNK, NOM.RNK and LFM). Together with these priors we also run NEMs with a sparsity prior as well as without any prior for comparison purpose. The knowledge sources integrated in our prior were: PPI data from PathwayCommons, KEGG pathway, Protein Domain knowledge and Gene Ontology annotations. The LFM priors were computed via the MCMC approach explained in the previous chapter (please see chapter 6) for 50,000 burnin and 200,000 sampling iterations. Other priors were also computed as explained (please see chapter 6).

We ran an NEM instance with every prior and the input data. For this an NEM algorithms were run indipendently with the input data i.e. the log p-value density and corresponding S-gene priors. Thus we had iferred network for a run with every type of prior defined earlier as well as sparsity prior and without any any prior.

7.5.2. Validation against literature

The results from each NEM instance were compared against a literature-derived network from MetaCore ^{TM}. The comparison was performed in a two-pronged approach. The first one reflected the edges in the NEM derived network and explained by literature, while the second traced out the pathways that exist in the literature network and explained by the NEM modes (Please recall that we did similar comparisons before in chapter 4,5 and 6). We refer to them as Knowledge View (KV) and Model View (MV) respectively. It must be noted that the literature used for validation was not used to compute the prior in order to minimize

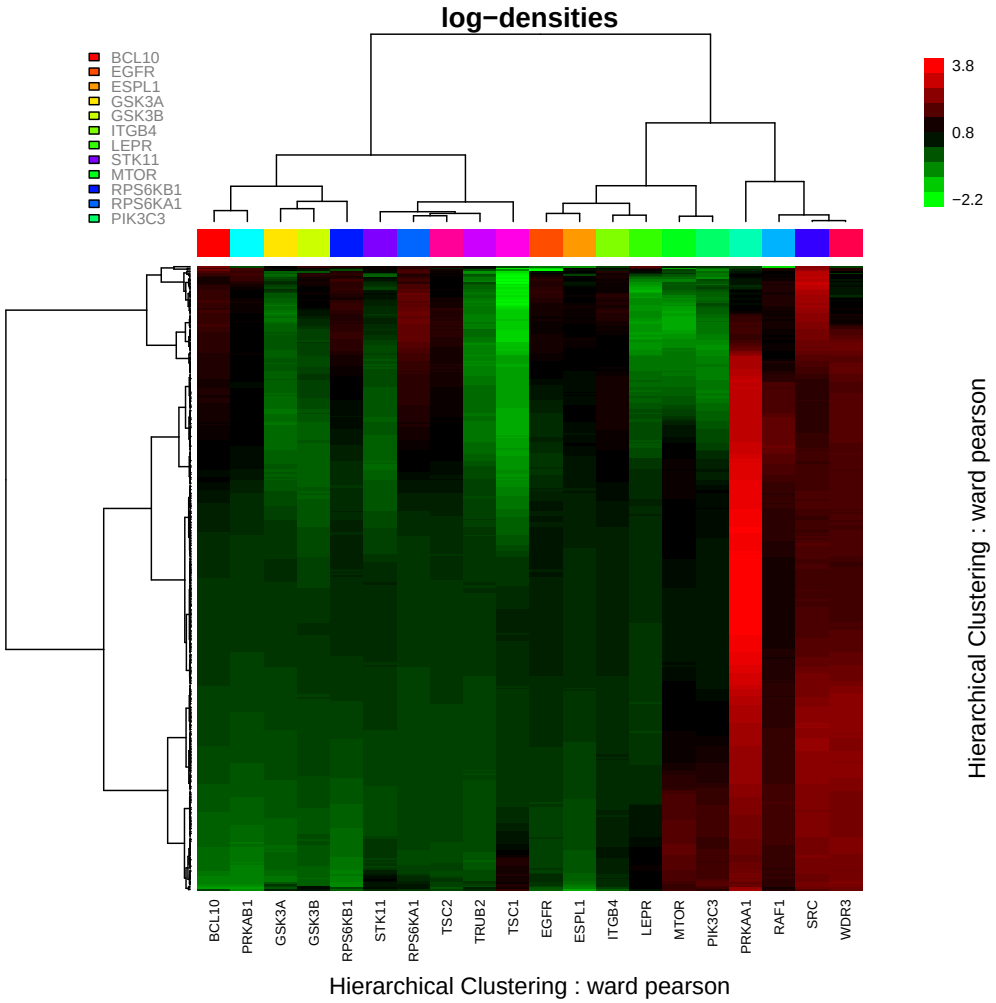

Figure 7.6.: Heatmap for the log p-value density matrix used as the input data.

bias during validation. Nonetheless, there are likely dependencies (for example between PPI and pathway information and the literature used in MetaCore ^{TM}). Hence, this is a principle limitation of the literature based validation discussed in the following.

The validation results outlined the effect of using prior for network inference via NEM very clearly (Figure 7.7). The NOM, NOM.RNK and LFM based networks could explain more edges existing in literature network than the model without using any prior (NP). The sparsity prior, as expected, showed fewer edges and hence explains very few edges. On the other hand KV score which reflects the inferred network edges being explained by the literature showed better results in case of IP, IP.RNK and LFM. On an average NOM and LFM seemed to perform better than other priors. The reason behind such discrepancies between LFM and NOM can be explained based on the nature of the two priors. LFM has higher specificity as we observed in chapter 6, whereas NOM is more sensitive. Therefore, LFM accepts fewer edges compared to the NOM or NOM.RNK. The high MV score of NOM are actually caused by this property of the prior. The IP and MP based priors are affected by numeric constraints of multiplication or

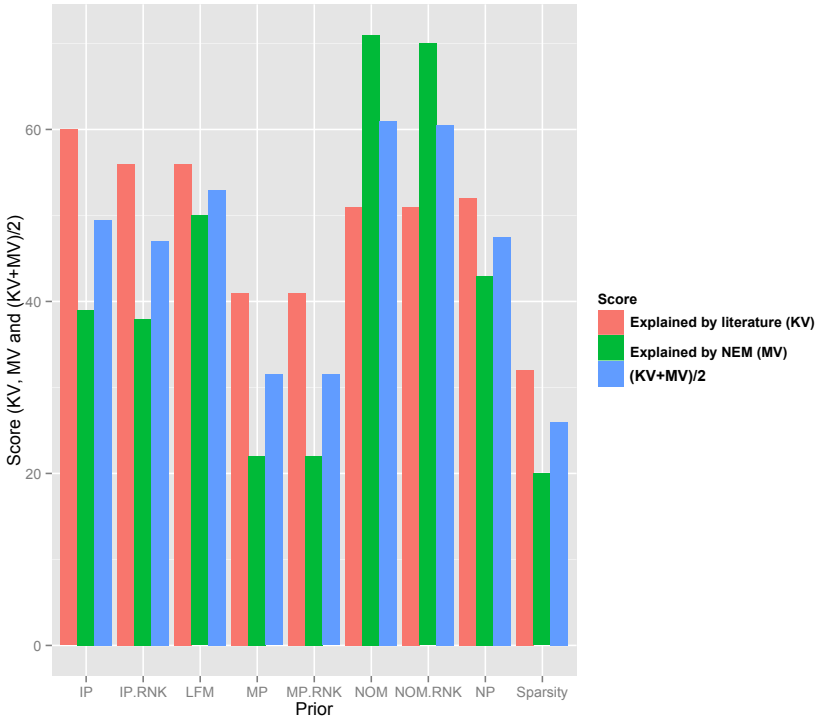

Figure 7.7.: Validation scores for NEM network reconstruction accuracy achieved with different priors. The bars depict KV, MV and the average of the KV and MV scores ((KV+MV)/2).

addition of small numbers. Therefore, they return high KV scores, but very low MV scores. The NEM network retrieved without using prior is completely data driven and performs well when validated against literature (compared to IP or MP based priors) but is outperformed by LFM and NOM based priors.

Furthermore, permutation test was conducted in order to check, if the network was inferred by chance. The node labels were permuted 1000 times and for each permuted network the likelihood was computed according to the NEM likelihood equation (Equation 7.1). The likelihoods of all permuted networks were found to be lower than that of the inferred network and hence our network was found to be significantly better than a random network (p-value $= 5.89 \times 10^{-4}$).

$$P(D|\Phi) = \prod_{k \in E} \sum_{s \in S} \prod_{t \in S} P(D_{tk}|\Phi, \Theta_{sk} = 1) P(\Theta_{sk} = 1) \tag{7.1}$$

For further studies we consider the network derived with NOM based priors as they show the best average for the two validation scores.

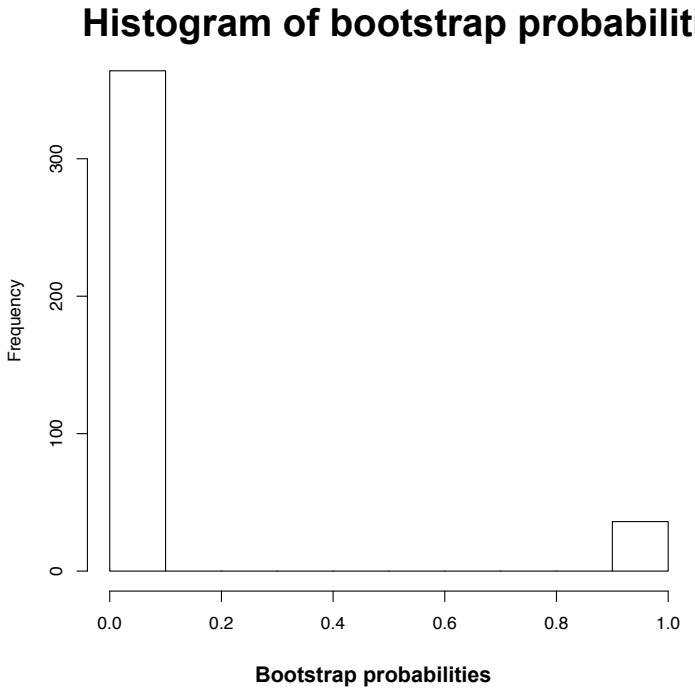

Figure 7.8.: Histogram for the bootstrap probabilities in the inferred network. The plot shows that the bootstrap probability is very high for few edges (close to 1) and low for many majority of edges (close to 0))

7.5.3. Bootstrap

To understand the robustness of our inferred network we ran a bootstrap for reconstruction of these networks with the best performing prior (NOM). 1000 bootstrap iteration steps were used to infer a bootstrapped network from NEM. The bootstrap process re-samples E-genes with replacement and reconstructs network with these samples to assess the statistical stability of the overall network. The results of the bootstrap returned the confidence in every edge of the inferred network. This confidence actually reflects the number of times the edge was accepted out of all bootstrap iterations, we term this as bootstrap edge probability. The bootstrap results confirmed that there were few edges with high confidence (> 0.5) and a large proportion of edges show a very small (close to 0) confidences (Figure 7.8). The edges with low bootstrap probability < 0.5 were not accepted in our network. A transitive closure of remaining edges was computed because in principle from static data the NEM model can only resolve the network structure up to equivalence classes. These equivalence classes are transitively closed graphs (see chapter 3). Figure 7.9 was our final network for analysis purpose.

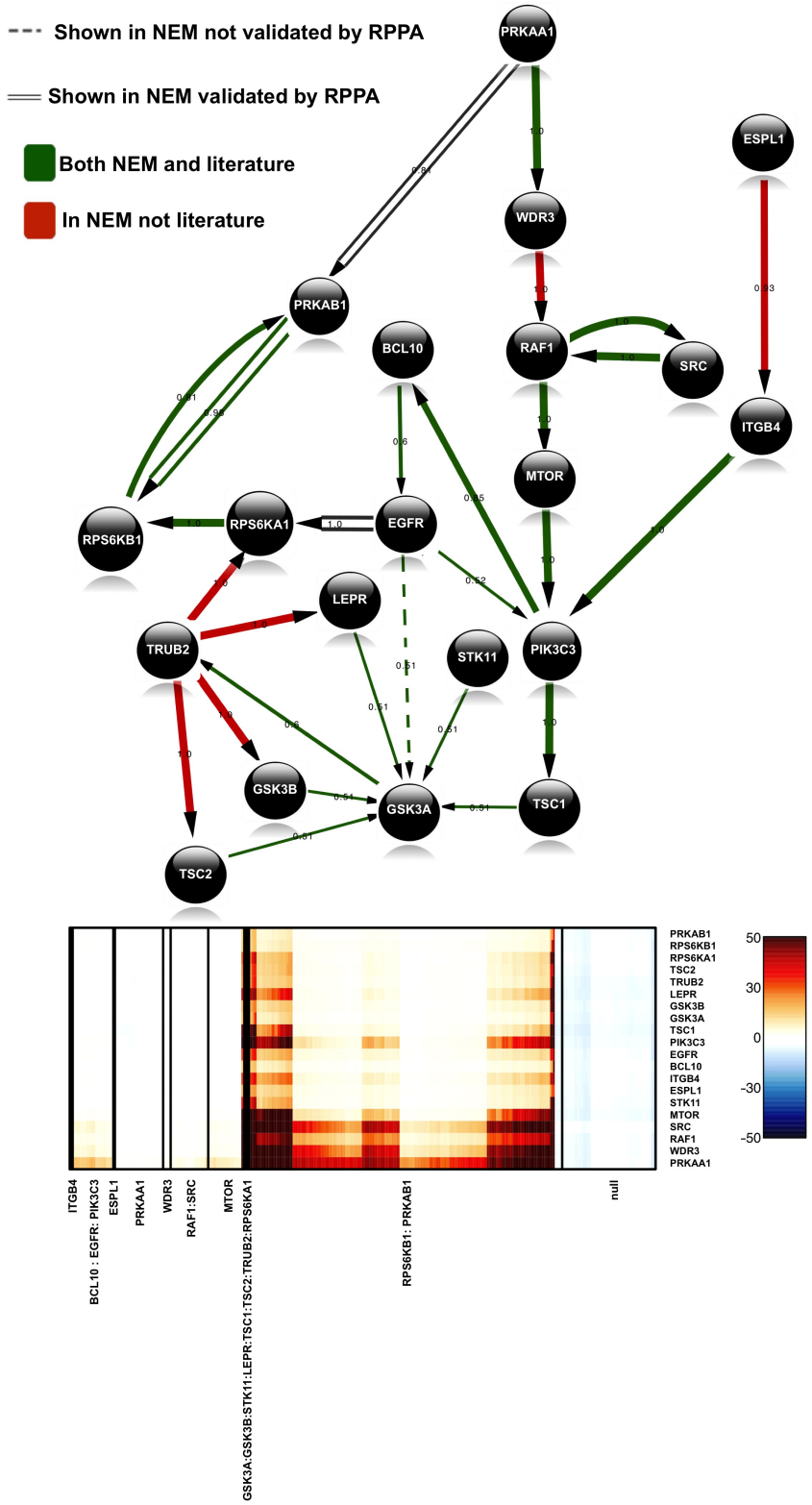

Figure 7.9.: Inferred network and E-gene map.(Top) Transitively reduced graph (Bottom) Effect mapping (effects plot) for E-genes (columns) on perturbation of every S-gene (along rows). The effects plot shows the upstream location of PRKAA1

7.6 Results

7.6.1. Inferred network

The network inferred showed many already known pathways in EGFR/AMPK signaling and found some interesting new links (Figure 7.9). As mentioned many of the edges were validated at protein level that represents signaling network in biological action. In total, six new edges were discovered. The network showed PRKAA1 as the main upstream regulator and many alternative cascades were detected throughout the signaling pathway.

As the network (Figure 7.9) is a graph after applying a transitive reduction (i.e. indirect edges removed), it is not unique. A detailed visualization of the network is in the figure 7.9 where the edge width represents the bootstrap confidence in each edge. The NEM could model not only the infer the network among S-genes, but also predicts effects on downstream on E-genes (Figure 7.9 (down)). Most downstream effects can be directly related to PRKAB1 and RPS6KB1. Knock-down of PRKAA1 yields effects on almost all other genes (except for ESPL1, ITGB4 and STIK11). This is reflected by the color code in the heatmap .

7.7 Validation

7.7.1. Literature based validation

We used same Metacore based networks again to validate the bootstrapped networks against literature. However, this time we look at two different properties, (1) Edge bootstrap probability and, (2) The length of path explaining every edge

As most of the edges as explained in the previous section, could be explained by literature pathways (Appendix J), we investigated the length of these paths in more detail. Out of all the explained edges $\sim$80% of edges were explained by path lengths ≤ 4. $\sim$ 40% of these explained pathways were mapped by literature based pathway of length ≤ 3 (Figure 7.11 A). The mean path length for explanation by literature was found to be <3. The NEM based network could explain the literature network mostly with longer paths. 60% of the explained edges in literature, were explained by had a path length shorter than or equal to 5 (Figure 7.11(B)).

7.7.2. Validating with experimental data

It is advantage to validate the retrieved network based on independent data set that itself was not used to infer the network. So far, we used other sources of network (here literature) to validate. However, an experimental validation is the best and most straightforward corroboration for our approach. We used the RPPA data described in the section 7.3 to validate parts of the inferred networks. The data consists of knockdowns for five different genes namely: PRKAA1 and PRKAB1, EGFR, MTOR and p70S6K. The data were normalized against FCP housekeeping protein. A limma analysis on the data returned the log fold changes in the expression of proteins and the corresponding p-values for all five knockdowns.

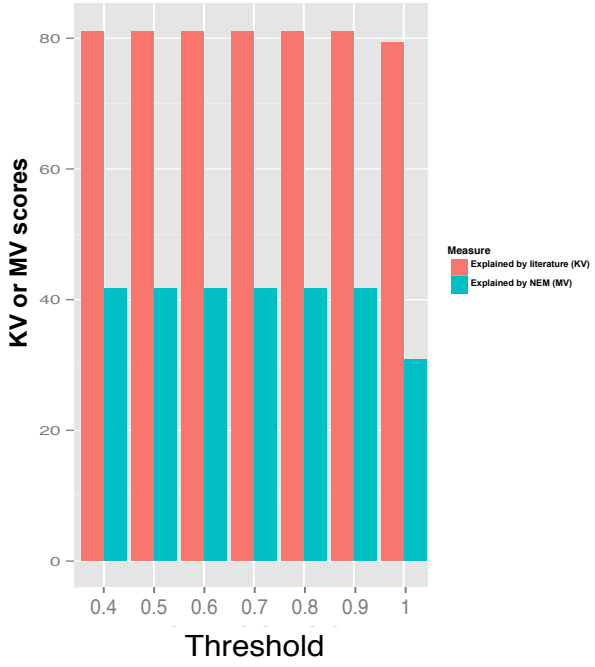

Figure 7.10.: The validation scores (Y-axis) for bootstrapped network for different thresholds (0.4 to 1) along X-axis for bootstrap probabilities of edges. The prior used for network reconstruction was NOM after evaluating it against all other priors (see section 7.5.2)

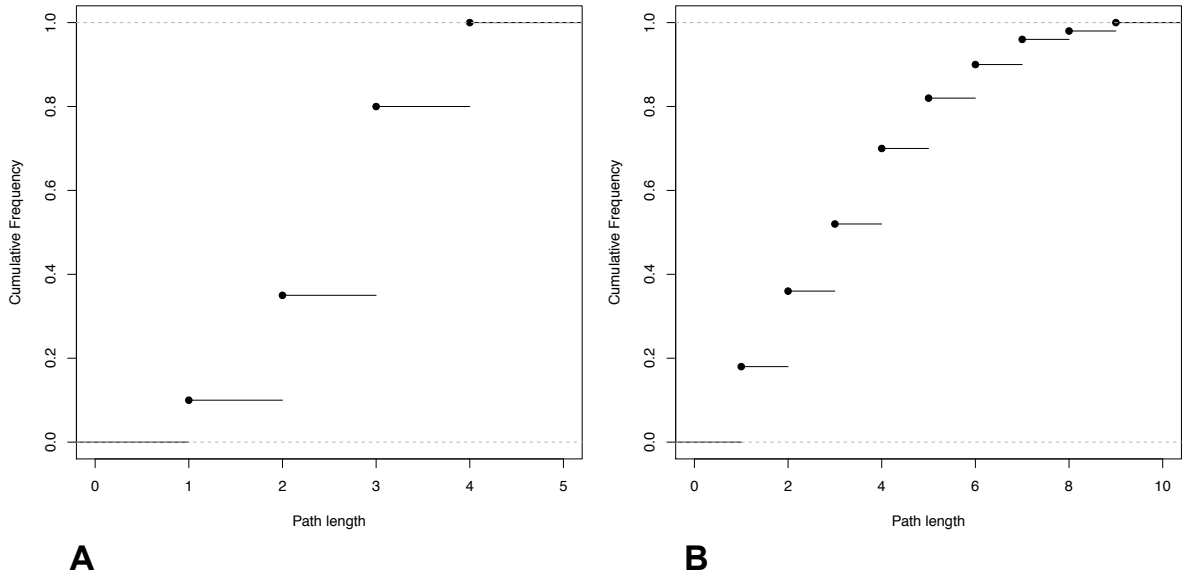

Figure 7.11.: Cumulative frequencies of path lengths. (A) Inferred network explained by literature (B) Literature network explained by inferred network.

First, we observed the overall effects of each perturbation. Figure 7.13, shows the log p-value densities for every knockdown in RPPA experiment. Knocking PRKAA1 down affected most of the measured protein, i.e. the effect was propagated throughout the other proteins in the signaling cascade. This is in agreement with our inferred network that positions PRKAA1 most upstream along a long path. The second rank in terms of most effective perturbation (in terms of propagation of effect along the cascade) went to MTOR. Looking at our NEM network, MTOR is also located at the upstream of most of the other genes, but downstream of PRKAA1. The heatmap shows that some protein measurements under MTOR is nested within PRKAA1. Nevertheless, in our inferred network, the path from PIK3C3 (immediate downstream of MTOR) shows a bifurcation. These observations at protein level boost our confidence in the inferred network.

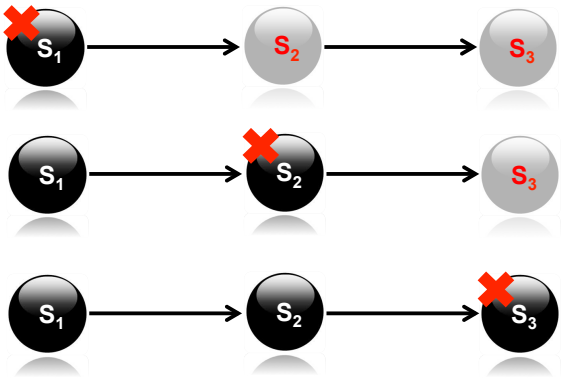

Figure 7.12.: Schema for validation of the network by looking at downstream nodes for every perturbation. The red cross on a node indicates its perturbation and the grey nodes depict the effect of perturbation on the downstream nodes. The affected downstrean nodes are computed by a breadth first search algorithm

From our NEM model we can make predictions on perturbation effects. If S-gene s is knocked-down, then all S-genes reachable from s (i.e. accessible via a directed path in the network) should exhibit significant effects as well as all their attached E-genes (Figure 7.12). The RPPA data allows to partialy validate these predictions. For each of the 5 knockdowns a number of phospho and total protein concentrations ($\sim$ 107) of network proteins were measured. Consequently, we can check, whether a knockdown of S-gene s on protein level indeed yields significant effects on S-gene q, placed downstream of s in the NEM inferred network.

The knockdown for EGFR showed a negative change in the protein kinase RPS6KB1 (Figure 7.14(A)). In our inferred network RPS6KB1 is located downstream of EGFR (Figure 7.9. The bootstrap probability for this edge was found to be 1 (Figure 7.9(B)). This means the edge was inferred during almost every bootstrap iteration. The EGFR knockdown shows the immediate effect at its downstream neighbor at protein level.

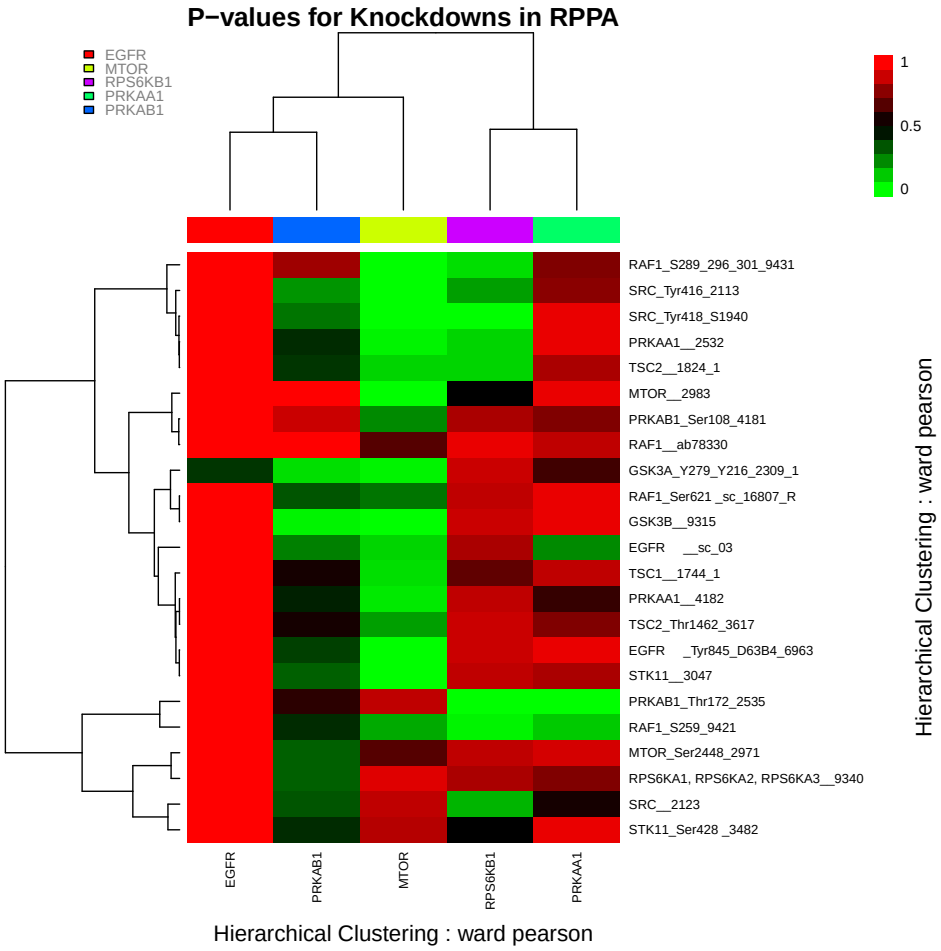

Figure 7.13.: Heatmap for log p-value densities observed in RPPA data for each of the five knockdown experiments.

The second knockdown for protein kinase RPS6KB1 (p70S6K) had two downstream S-genes PRKAB1 and itself. In the inferred network the bootstrap probability for the RPS6KB1 $\rightarrow$ PRKAB1 edge is 0.81 (Figure 7.9). The change for PRKAB1 after knocking down RPS6KB1 was found to negatively change by two folds (Figure 7.14(B)). This significant shift shows the effect of RPS6KB1 on PRKAB1 as suggested by our inferred network.

PRKAB1 has been found to be located downstream of PRKAA1 (catalytic subunit of AMPK) by NEM. Knocking down PRKAA1 (AMPK) displayed immediate effect on PRKAB1 with a log fold change of $\sim$ -1.5 (Figure 7.14(C)). The edge between PRKAA1 and PRKAB1 has a bootstrap probability of 1.0 (Figure 7.9). Thus, this observation at protein level is also in agreement with our network.

PRKAB1 is a regulatory subunit of the AMPK affected the protein kinase RPS6KB1. Interestingly, knockdown of PRKAB1 displays an effect for the phosphorylated RPS6KB1 (Figure 7.14(D)). The observed log fold change was $\sim$ -2 for phosphorylated protein, whereas much

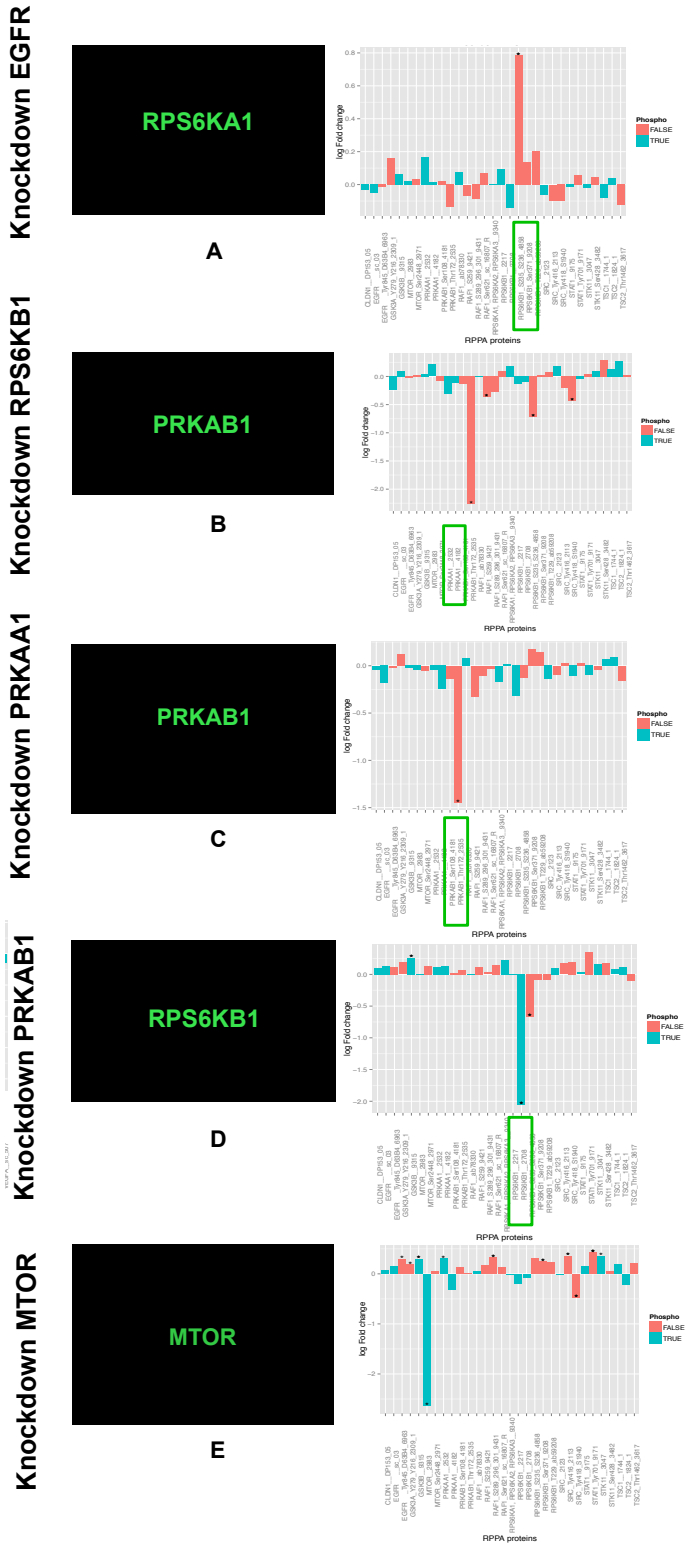

Figure 7.14.: Knock down for five genes (EGFR, RPS6KB1, PRKAA1 PRKAB1 and mTOR) in terms of log fold changes of network proteins via RPPA. The downstream genes showing significant changes have been provided in the box on the left hand side. The corresponding significant measured protein have been highlighted in green. (log fold changes of all proteins in L.2)

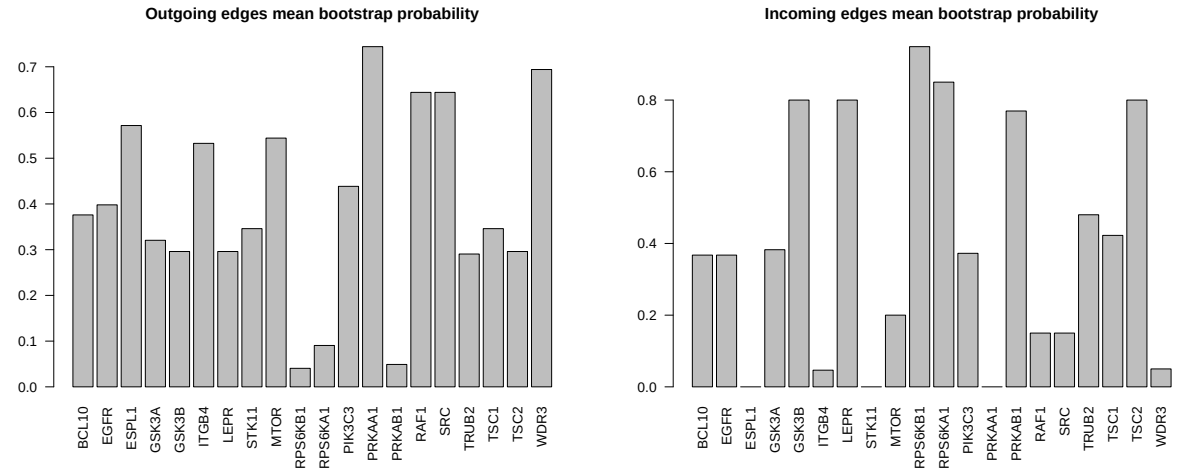

Figure 7.15.: Mean confidence for outgoing and in coming edges for nodes in the inferred network

smaller for non-phosphorylated protein. RPS6KB1 can be found downstream of PRKAB1 with a bootstrap edge probability of 0.98, which is extremely high. This observation proves another inference of NEM to be correct at protein level.

Knocking down MTOR did not produce any significant change for any protein than MTOR itself (negative fold change > 2) (Figure 7.14(E)). Here, the RPPA data did not provide any validation results.

The network is inferred with data at mRNA level and the validation with protein level data, depicting the signaling network in action and the effect of signaling cascade. Many of the edge we observed in our inferred network using knowledge and NEM were verified and found to be true at protein level. Since, the measurements were not made for all the proteins (E-genes and S-genes) in RPPA, our validation was limited to selected edges. Furthermore, a time resolved data can enhance the power of validation as it will actually take into account the time delays involved in signaling as well as translation.

7.7.3. Biological implications

Few vertices/nodes in the inferred network had higher mean probability for outgoing edge, while others showed higher mean probability for incoming edges (Figure 7.15). The nodes showing the former quality act as regulator for the downstream proteins and usually affect many genes directly or indirectly. Examples of such node are PRKAA1, SRC, RAF1 and WDR3. These proteins act as the major controllers for the downstream proteins and affecting these genes can propagate the effects along different directions in course of signaling. Other proteins like the protein kinases; p70S6K and p90S6K (RPS6KB1 and RPS6KA1) have higher probability for incoming edges, indicating that they can be regulated in many ways. This means that in order to cast an effect on then a direct or multi-target approach is needed.

PRKAA1 plays a key role in the control of cell growth, proliferation and autophagy through the regulation of mTOR activity, which is consistently deregulated in cancer cells (Chapuis et al., 2010). It has been shown by Guertin et al. that mTOR serves as a central integrator of nutrient and growth factors controlling cell growth in all eukaryotes and is deregulated in most human cancers (Guertin et al., 2009). It has also been shown that mTORC1 can be suppressed following AMPK activation (Hahn-Windgassen et al., 2005; Gwinn et al., 2008). The network inferred here is in compliance with these facts (PRKAA1 $\rightarrow$ WDR3 $\rightarrow$ RAF1 $\rightarrow$ MTOR).

PRKAA1 (AMPK Subunit α) in the inferred network is found at the most upstream position in the inferred network. It is known to regulate the activities of some key metabolic enzymes through phosphorylation (Shackelford and Shaw, 2009). WDR3 has been shown to be controlled via P53, and P53 in turn by PRKAA1 (McMahon et al., 2010). The WDR family of genes have been found to play critical role in cell cycle and apoptosis (Clemen et al., 2008). The edge PRKAA1 $\rightarrow$ WDR3 depicts the regulation of WDR3 by the PRKAA1 in the inferred network. Thus, the network displays the regulator role of the AMPK gene. Furthermore, the PRKAB1 (AMPK Subunit β) being a regulatory subunit of the AMP-activated protein kinase was shown to be directly in downstream to the PRKAA1. These two subunits together could regulate the ribosomal protein kinase RPS6KB1 (PRKAA1 $\rightarrow$ PRKAB $\Leftrightarrow$ RPS6KB1)

Subunits of Glycogen synthase kinase-3 (GSK3A and GSK3B) are known for their role in cell division, proliferation, motility and survival together with a number of pathological conditions including cancer and diabetes, and is increasingly seen as an important component of neurological diseases (Forde and Dale, 2007). Regulation of GSK-3 is important for the normal development, regulation of metabolism, neuronal growth and differentiation and modulation of cell death (Kockeritz et al., 2006). Our network shows GSK3A and GSK3B subunits to be present downstream of some key regulators like EGFR, PIK3C3, BCL10, MTOR, etc. This perhaps means that there exist many alternative ways to regulate GSKs.

p70S6K and p90S6K (RPS6KA1 and RPS6KB1 respectively) are located downstream of PIK3C3 in our inferred network. These kinase proteins play critical roles in translational regulation. mTOR/p70S6K pathway is considered a central regulator in various malignant tumors and can arrest the G0/G1 phase and induced apoptosis of cells (Ruvinsky et al., 2006). Our networks depict that there can be more than one pathway regulating p70S6K, specially outlining the role of LKB1 (STK11), TSC1 and EGFR.

7.7.4. AMPK as a Potential Drug Target for NSCLC

These findings based on our inferred network are not only consistent with previous studies, but also present many new interactions that are potentially interesting. These potential interactions can serve as the starting points for combinatorial treatment of NSCLC and probably for other tumors.

Tumorigenic transformation is perceived as a result of changes in signaling pathways caused by genetic or epigenetic factors (Shaw and Cantley, 2006). Several anti-cancer drugs target components these signaling pathways. An example is EGFR inhibitor Erlotinib (Paez et al., 2004). However, resistance to these targeted therapies is a major concern in cancer research

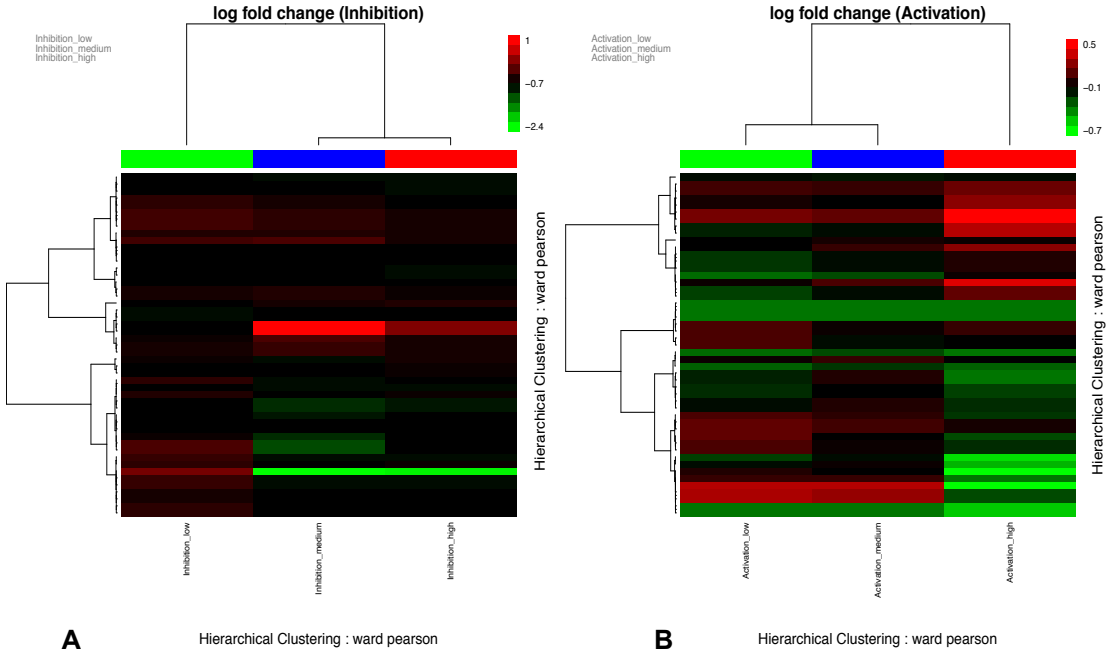

Figure 7.16.: Effect of drug treatment in different concentrations on log fold changes of network proteins. (A) Inhibition treatment at three concentrations, high (1 μM), medium (5 μM) and low (1 μM) (B) Activation treatment at three concentrations high (100 μM), medium (50 μM) and low (10 μM)

(Holohan et al., 2013). Combinatorial treatments against target holds considerable promise (Yarden and Pines, 2012). Our work presented here is a step in this direction. So far, we presented a method to infer signaling network involving AMPK and EGFR. This part of the chapter will lead in this direction. This suggests a central regulatory role of AMPK (PRKAA1 specifically) on the whole network. Hence, PRKAA1 may be an interesting drug target.

This hypothesis was validated via the available RPPA data. These data contains phosphorylation specific measurements of $\sim$ 107 proteins. We analyzed the GO annotation for these protein. Most network proteins are involved in cell-cycle, apopotosis and cell-proliferation (Appendix K). This suggests to see network proteins as markers for the success of a AMPK targeting therapy. We selected two compounds (1) Compound C, with inhibitory effect on PRKAA1, and (2) A-769662, with activation effect on PRKAA1. The log-fold changes of proteins with interesting GO terms were observed after PRKAA1 targeting drug with inhibitory and activating effects (Figure 7.17 and L.4). The results showed most of the measured proteins to be interesting as they were associated with the biological processes of our interest. Furthermore, the treatment with inhibiting drug showed down regulation of many of these proteins. This establishes the top level position of AMPK in our inferred network. This also indicates that the perturbation of some of these proteins can be interesting.

Our protein array data has the drug treatment in three concentrations low, medium and high for both activating and inhibitory actions. Observing the effect of these different levels of

concentrations indicates that a high concentration of AMPK inhibitor down-regulates most of the network proteins (S-genes + E-genes). The effect of the highest concentration ($100\mu M$) (Figure 7.16) is caused by the upstream location of PRKAA1 in the signaling network. This also shows that PRKAA1 is an interesting drug target being a master controller for most of the other proteins which has already been found to be interesting in our GO analysis. AMPK inhibition yields no effect on most cell proliferation markers. On the other hand AMPK activation with high ($10\mu M$) concentrations leads to a down-reglation of most cell proliferation markers. Hence, AMPK activation - in addition to EGFR treatment - could be an interesting therapeutic option.

7.8 Summary

Here we presented study of NSCLS cells after perturbation of key players of AMPK/EGFR pathways. Our application on perturbation effects in NSCLC at the methodology level proved the importance of using prior in NEMs. The results presented another proof that the use of consensus probabilistic prior computed from multiple sources of knowledge can drastically improve the accuracy of NEMs. This integrative approach not only inferred more of the known pathways, but also enabled us to recognize new pathway links that were unknown before.

At the level of biological knowledge the chapter unveiled many interesting links among the proteins which can serve as alternative links unknown to community until now, but may exist in real life cellular processes or filled up the missing links. The various ways that can control and regulate the functioning of a biological process not only enhanced our knowledge, but also created space for future research in treatment and management of tumors via combinatorial therapy. The use of experimental data to validate the inferred networks leveraged a stronger confidence in the hypothesis of the thesis that an integrative approach to reconstruct/reverse engineer cellular network is more powerful that solely data driven approaches.

Finally we presented the role of some crucial players in the EGFR/AMPK signaling pathway. The drug effect data for activation and inhibition of PRKAA1 supported it is critical role in the pathway. Based on the inferred network, drug-dose data and GO analysis we proposed PRKAA1 as a potential subject for studying its role in cancer as drug target.

This chapter presented an application of Nested Effects Models in combination of the developed consensus probabilistic prior from multiple knowledge sources. The results unveiled many interesting avenues in terms of new pathway links which were otherwise undiscovered without the use of prior (only data drive approach). The upcoming final chapter of the thesis will bring out the conclusive message of the thesis together with future outlook.

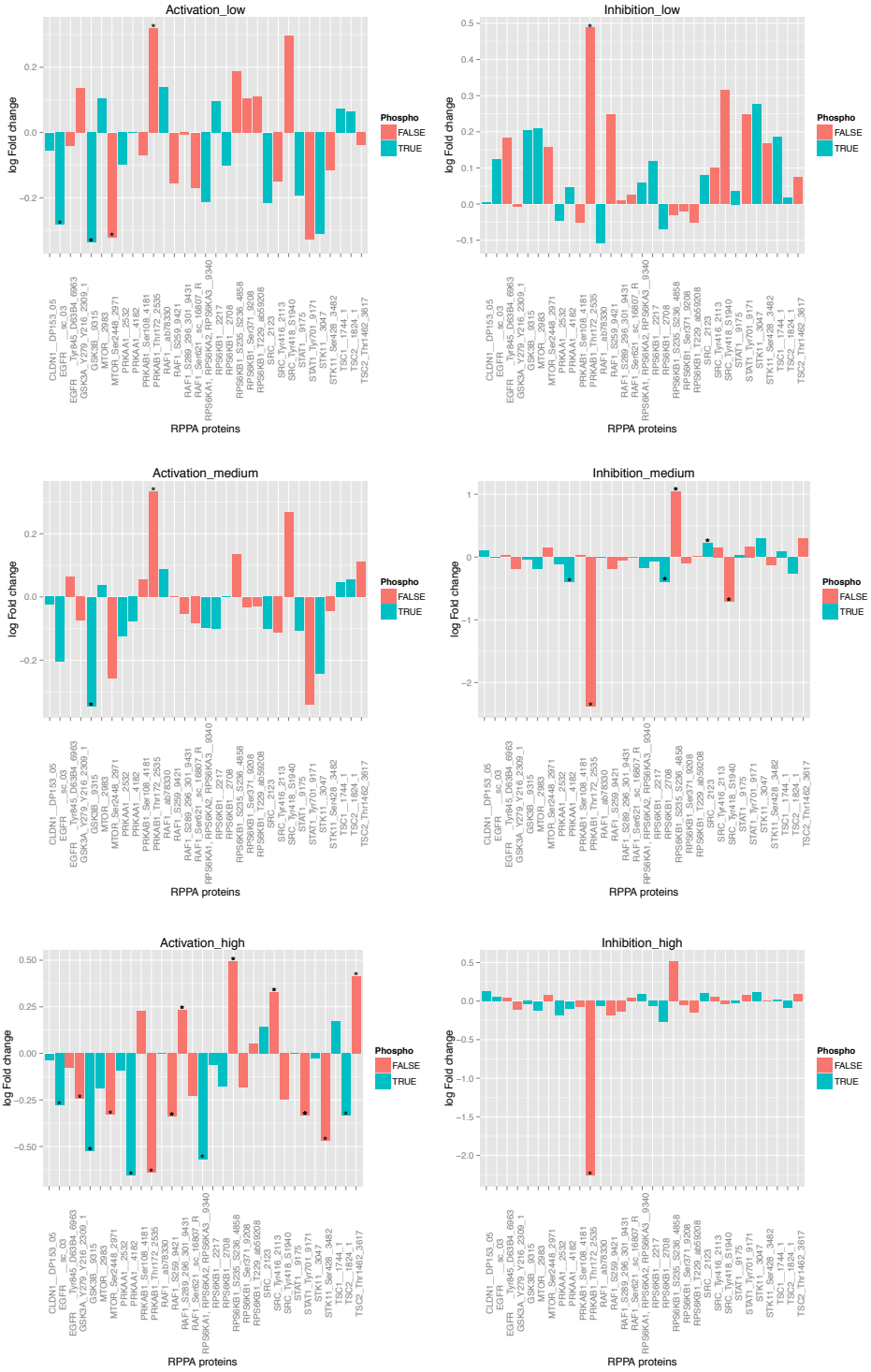

Figure 7.17.: log fold changes drug treatment for network proteins (S-genes). (Left) The plot for activation treatment (dose concentration: high (100 μM), medium (50 μM) and low (10 μM). (Right) The plot for inhibitory treatment (dose concentration: high (1 μM), medium (5 μM) and low (1 μM)). A complete plot for all proteins can be found in appendix L.

Chapter 8
Conclusion and Outlook

This chapter draws together, the conclusions reached based on the research presented in this thesis. First, the main accomplishments are summarized, followed by a discussion of the future outlook in the domain, entailing the potential avenues for further research. Finally, the chapter ends with the impact brought about by this thesis in the field of data plus knowledge driven reconstruction of cellular networks followed by a conclusive sum-up.

8.1 The urge for 'Integrative Systems Biology'

Systems biology is one of the latest trends in life science research. It has been driven by advances in technologies. These technologies provide a suite of $'omics'$ (Lederberg and Mccray, 2001) techniques generating data, which carries a whole range of measurements for genes, proteins and metabolites. The state of art methods today use this $-omics$ data as the key to disentangle the complex architecture of biological systems. However, besides the data, we also have established knowledge in literature and databases that can be also important to understand the structure and function of a biological system. The data coming from high throughput experiments, s typically noisy and not always reproducible, the knowledge is restricted and incomplete. Nevertheless, if they can be used complementarily the reconstruction and hence understanding biological systems may become more realistic. Therefore, the emerging issue is to integratively use generated data (one or more sources) together with knowledge to get more comprehensive insights into biological systems. The incentive expected is that all of these, when combined in some way, will provide detailed picture of important biological processes, such as development, aging, disease, to the benefit of the augmentation of new knowledge and disease treatment/management.

8.2 Accomplishments

This thesis aimed to work in this direction by developing new methods to learn the architecture and design of biological systems from different types of data and knowledge integratively. The following subsections summarize the contributions of this thesis made with respect to the reverse engineering of biological networks based on Nested Effects Models (NEM) (Markowetz, 2005).

8.2.1. Network from time-course data

In this thesis we introduced a new algorithm called dynoNEM for network inference. dynoNEMs extended the NEM framework towards the use of time course gene expression data coming from targeted RNAi perturbation experiments (Fröhlich et al., 2011). The dynoNEMs enabled the exploitation of temporal data with their ability to use the time factor together with downstream perturbation effects. DynoNEMs model these dynamic effects of perturbation by unrolling the network structure over time. The dynoNEM returns a directed edge-weighted graph where, the edge weight refers to the time lag (time-point interval).

The dynoNEM method allows distinguishing direct perturbation effects from indirect effect and also facilitates the resolution of feedback loops. These advantages are of great help in biology for improving the network reconstruction accuracy (Feizi et al., 2013). This effect was demonstrated in the dynoNEM chapter with many simulations in order to study the dependency of the algorithm on parameters like network size (S-genes), number of E-genes (measurement of effect genes), number of time points etc. The reconstruction of murine stem cell network served as an application example.

8.2.2. Learning from phenotype data

The dynoNEMs could use the time course expression data to learn the networks among the perturbed genes. However, such perturbation observation and measurements are not limited to expression data. Phenotype data in the form of cellular images can also represent perturbation effects. To harness such data the thesis proposed an extension of dynoNEM in the form of MovieNEM (Failmezger et al., 2013).

The underlying principle of MovieNEM approach is that perturbation can yield differences in cellular phenotype. These differences can be depicted in terms of measurable morphological features. The MovieNEM method quantifies these morphological features in terms of image features from time laps microscopic movies. As it uses the images from movie, the time factor is inherent in the data. These features are extracted from the image and then are statistically tested for their association with perturbation. These associated features can be used to infer the cellular networks. The dynoNEM uses these features as an analogue to the E-genes and leverages its network inference algorithm to this data to infer the involved network that caused these phenotypes.

MovieNEM approach is a step towards better exploitation of this information-rich data to reverse engineer cellular networks. It widens the scope of applicability of NEMs. During the research presented in the thesis, time lapse cell imaging data resulting from RNAi knock downs was used to infer cellular networks involved in cell cycle. The resulting network showed very high reconstruction accuracy when compared against literature based networks. The MovieNEM also showed its robustness against non-informative features. As omics data is not always available or affordable, MovieNEM as an image-based techniques can be considered as an alternative.

8.2.3. Integrating knowledge

High throughput data bears the problem of inherent noise together with a comparably low number of samples, limiting the accuracy of network reconstruction. These bottlenecks of using only high throughput data may be addressed by incorporating an informative prior based on existing scientific knowledge, during network reconstruction. Therefore, after paving the way for new types of data (time-course and phenotype data) to infer networks, the next goal was to achieve the integration of biological knowledge in network inference algorithm. This required an approach to integrate knowledge from different sources ranging from database and knowledge-base to literature. Different methods were proposed to compute and integrated prior knowledge by combining knowledge from multiple sources. Latent Factor Models (LFM) and Noisy-OR Models (NOM) along with few other approaches were proposed to form an informative prior (Praveen and Fröhlich, 2013). These methods generate a matrix of confidence values ranging from 0 to 1 for each edge in the network based on the existing knowledge.

The simulations proved their usability on artificial as well as real experimental data. The proposed frameworks allowed the use of diverse information sources, like pathway databases, GO terms and protein domain data, etc and were found to be flexible enough to integrate new sources of knowledge. The performance was observed to reconstruct random KEGG graphs with high accuracy and later with artificial expression data. The method was applied on the murine stem cell data, where we used dynoNEM together with biological knowledge from different databases and other sources of knowledge to infer the network. The work showed advantage of using prior knowledge in network inference by increasing the reconstruction accuracy compared to the network inferred without using prior knowledge. Integrating knowledge into computer systems can solve complex problems that normally require a high-level of human expertise (Feigenbaum and McCorduck, 1983). But such integration requires knowledge acquisition, computable representation, refinement and validation (Payne, 2012). In this thesis we have shown these concepts in action.

8.2.4. Applying the integrative prior

As mentioned above the drawbacks of using only experimental data to reconstruct biological networks can be overcome by incorporating informative prior from multiple knowledge sources. After proposing methods to combine different in the inference process, these methods were applied using NEM. The data used for application was from a targeted perturbation experiment for 20 genes involved in AMPK pathway.

An empirical evaluation, showed that the use of integrative informative prior knowledge obtained using the proposed approaches can improve the network reconstruction. Specially LFM and NOM boosted both the number of true regulatory interactions present and the predictive performance of the learnt model. The comparison was done with to a network reconstructed solely from expression data and some other priors via validation against literature based networks. The data were also validated experimentally using reverse phase protein arrays. The results established that use of biological knowledge enhances the power of NEM to reconstruct pathways.

8.2.5. Identifying therapeutic targets in NSCLC

The resulting model for the AMPK/EGFR signaling obtained via the developed integrative approach was used to identify new potential drug targets for NSCLC followed by their experimental validation. Furthermore, these targets were experimentally validated, with two compounds for activation and inhibition of PRKAA1 and were also validated on mouse models. This not only proposed the alternative therapy in case of loss of sensitivty for EGFR targeting treatments in NSCLC but also added the polypharmacogeny perspective in the treatment. Thus, the thesis could meet one of the most desired goals of therapeutic application in systems biology.

8.3 Impact

In chapter 1, we presented the generalized workflow for network reverse engineering (Figure 1.4). The overall impact of the research performed in this thesis are new blocks added to this framework as well as improvements in some existing blocks (Figure 8.1). At the level of the method was empowered to be able to use the time course data and the phenotype data was also added the image/movie data into the picture. The new network inference methods appended to the framework were dynoNEM and movieNEM. Finally the aiding the network reconstruction process with established knowledge was another important key impact of the thesis. Thus, improving the entire network reverse engineering framework in many ways.

8.4 Future outlook

The task of cellular pathway reconstruction is one of the biggest challenges of the post-genomic era. The work described in this dissertation are only few advancements and steps towards the goal of automated reconstruction of cellular pathways. The following subsections outline potential avenues for future research in the domain.

8.4.1. Data Integration

The methods discussed in this thesis considered a single source of data and knowledge sources. However, Surowiecki's philosophy of wisdom of crowds discusses multiple intelligence (Surowiecki, May 25, 2004). With this he reasons that a properly structured group can come up with the better or right answer to a problem, than an individual. Applying the same principle in reverse engineering of biological networks is a reasonable approach. The problem to be addressed here is the presence or absence of an edge in a potential network and the crowd can be various sources of data or various reverse engineering approaches. The key to model such method is to structure the crowd in order to reach the best possible decision. Certain ways have been suggested performing such integrative reverse engineering (Lemmens et al., 2009; Marbach et al., 2012).

The methods proposed to integrate prior knowledge from multiple sources can be adapted and applied to combine data from multiple experiments. Furthermore, the proposed integration methods can also be used to combine the networks inferred by different approaches, thus yielding the effect analogous to boosting in statistical learning. This will more rational way

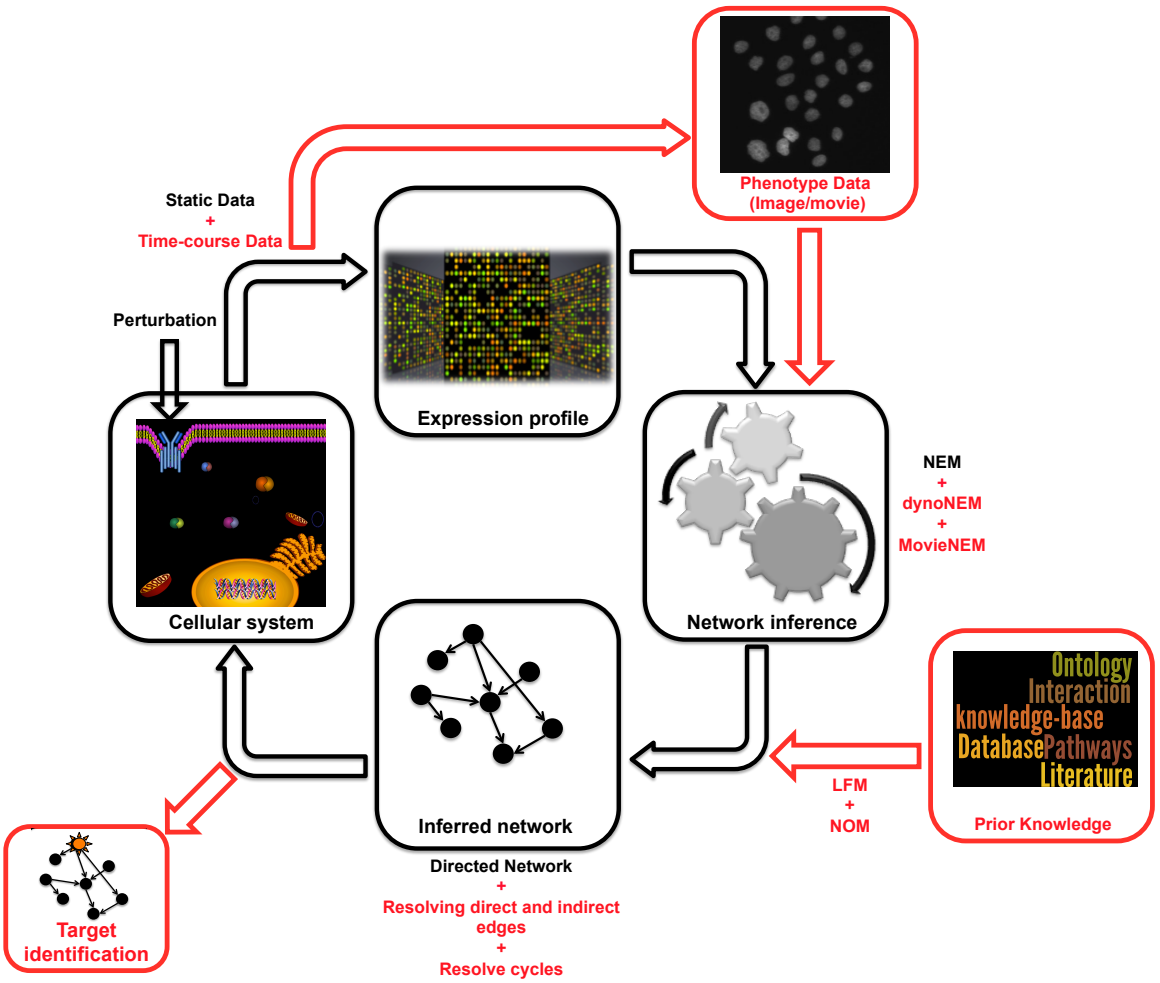

Figure 8.1.: Contributing towards the network reverse engineering and modeling cycle in systems biology. In black is the existing pipeline before and in red is the contribution of this thesis.

as the proposed methods (LFM) assumes the different sources to be a result of real related biological process.

8.4.2. Going beyond network inference

The work explored mainly the integrative approach for network inference. However, beside being used in network inference the integrative approaches can also be used in other purposes like clustering. A clustering approach integrated with knowledge can lead to the evolution of a semi supervised clustering method. Such approach will not use merely data, but the knowledge as well to qualify data points in biologically more sensible clusters. The integration of network topology with functional enrichment similarities (KEGG and GO data) can also aid in recognition of functional modules in biological networks extending the concept of topological modules (Langfelder and Horvath, 2007).

8.4.3. Generating testable hypotheses

The discovery of new pathways is not always possible by using existing knowledge or experimental data. Getting new information out of biological systems often requires new experiments. In case of NEMs we talk in terms of new perturbation experiments. The experiments are based on certain hypotheses which are actually tested with the experiments. Deriving such new testable hypotheses from a single set of experimental data has become a major obstacle (Larsen et al., 2007). Generating such testable hypothesis are primarily based on the intuition of researchers. An emerging and increasingly viable solution to this challenge is the use of existing knowledge together with available data.

The NEMs yields a cellular pathway based on data. Furthermore, we can introduce high confidence edges from established knowledge (prior) into the inferred network in order to get a more comprehensive picture. Interesting genes in the network can be perturbed in-silico to observe the effects. Interesting effects can be then tested experimentally in laboratories for verification. A multiple perturbation can also be performed in-silico for a more clear insight. A Boolean network based approach can be adopted for simulating such in-silico perturbations. The work in this thesis is a step in this direction.

8.5 Summing it up

In this dissertation we dealt with the domains of network inference and knowledge integration, focusing mainly on targeted perturbation data. This thesis explored how to recover features of cellular pathways from data and existing knowledge integratively. It was an attempt to go beyond the gene expression data to infer cellular network. It looked forward to diverse data type ranging from high throughput $-omic$ data (NEM and dynoNEM on gene and protein expression data) to phenotypic data (MovieNEM). Furthermore, it uncovered the potentials of existing knowledge for improving networks inference focusing primarily on NEMs.

All in all, this thesis attempts to answer three key questions

1. How can we widen the scope data driven network inference beyond expression profiling?
2. Can we combine heterogeneous knowledge existing in biology in a consistent and quantitative way ?
3. Does the use of an informative prior improves the performance of network reconstruction method ?

The thesis and the research within do not claim to have found and proposed the best or the only solutions towards these problems. Nevertheless, the thesis aimed to explore the space around these issues and discover solutions to improve the existing methods or proposed some possible approaches to handle the above mentioned problems. After proposing the methods through this thesis, it remains an open argument to find better and alternative solutions. There is a lot to be explored in the domain of network inference and data-knowledge integration.

The most exciting challenge of all, is to explore the ways to use the reconstructed models to better understand life; to understand how biological regulation works, how cross talk between metabolic pathways and the perturbations/stimuli can control levels of metabolites in the cell? It will lead to develop an understanding of diseases such as cancer, and more importantly how to possibly rectify such failures will be the ultimate goal.

Finally the thesis opens up new areas to be explored. Pathway reconstruction or network reverse engineering is not merely an issue of more advanced and sophisticated inference techniques, but also a matter of careful experimental planning and design. Well designed experiments focus on a pathway of interest and probe the information flow caused by interventions.

Bibliography

Tarmo Aijö and Harri Lähdesmäki. Learning gene regulatory networks from gene expression measurements using non-parametric molecular kinetics. *Bioinformatics*, 25(22): 2937–2944, 2009. doi: 10.1093/bioinformatics/btp511. URL http://bioinformatics.oxfordjournals.org/content/25/22/2937.abstract.

T Akutsu, S Miyano, and S Kuhara. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In *Pac Symp Biocomput.*, 1999.

Bissan Al-Lazikani, Udai Banerji, and Paul Workman. Combinatorial drug therapy for cancer in the post-genomic era. *Nat Biotech*, 30(7):679–692, 07 2012. URL http://dx.doi.org/10.1038/nbt.2284.

Réka Albert and Zoltán N Oltvai. Shaping specificity in signaling networks. *Nature genetics*, 39(3):286–7, March 2007. ISSN 1061-4036. doi: 10.1038/ng0307-286. URL http://www.ncbi.nlm.nih.gov/pubmed/17325675.

U Alon. Biological networks: the tinkerer as an engineer. *Science (New York, N.Y.)*, 301 (5641):1866–7, September 2003. ISSN 1095-9203. doi: 10.1126/science.1089072. URL http://www.ncbi.nlm.nih.gov/pubmed/14512615.

Uri Alon. *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall/CRC Mathematical & Computational Biology, 2006.

Benedict Anchang, Mohammad J. Sadeh, Juby Jacob, Achim Tresch, Marcel O. Vlad, Peter J. Oefner, and Rainer Spang. Modeling the temporal interplay of molecular signaling and gene expression by using dynamic nested effects models. *Proceedings of the National Academy of Sciences*, 106(16):6447–6452, 2009. doi: 10.1073/pnas.0809822106. URL http://www.pnas.org/content/106/16/6447.abstract.

Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50(1-2):5–43, 2003.

Silvana Badaloni, Barbara Di Camillo, and Francesco Sambo. Qualitative reasoning for biological network inference from systematic perturbation experiments. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(5):1482–1491, 2012. ISSN 1545-5963. doi: http://doi.ieeecomputersociety.org/10.1109/TCBB.2012.69.

Marc Bailly-Bechet, Alfredo Braunstein, Andrea Pagnani, Martin Weigt, and Riccardo Zecchina. Inference of sparse combinatorial-control networks from gene-expression data: a message passing approach. *BMC Bioinformatics*, 11:355, 2010.

Chris Bakal, John Aach, George Church, and Norbert Perrimon. Quantitative morphological signatures define local signaling networks regulating cell morphology. *Science (New York, N.Y.)*, 316(5832):1753–1756, June 2007. ISSN 1095-9203. doi: 10.1126/science.1140324. URL http://dx.doi.org/10.1126/science.1140324.

Mukesh Bansal, Vincenzo Belcastro, Alberto Ambesi-Impiombato, and Diego di Bernardo. How to infer gene networks from expression profiles. *Molecular systems biology*, 3(78):78, January 2007. ISSN 1744-4292. doi: 10.1038/msb4100120. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1828749&tool=pmcentrez&rendertype=abstract.

Katia Basso, Adam a Margolin, Gustavo Stolovitzky, Ulf Klein, Riccardo Dalla-Favera, and Andrea Califano. Reverse engineering of regulatory networks in human b cells. *Nature genetics*, 37(4):382–90, May 2005. ISSN 1061-4036. doi: 10.1038/ng1532. URL http://www.ncbi.nlm.nih.gov/pubmed/15778709.

Jeff Bilmes. Dynamic bayesian multinets. In *UAI*, pages 38–45, 2000.

Jesse S Boehm and William C Hahn. Towards systematic functional characterization of cancer genomes. *Nature reviews. Genetics*, 12(7):487–98, July 2011. ISSN 1471-0064. doi: 10.1038/nrg3013. URL http://www.ncbi.nlm.nih.gov/pubmed/21681210.

Michael V. Boland, Mia K. Markey, and Robert F. Murphy. Automated recognition of patterns characteristic of subcellular structures in fluorescence microscopy images. *Cytometry*, 33(3):366–375, 1998. doi: 10.1002/(sici)1097-0320(19981101)33:3\%3C366::aid-cyto12\%3E3.0.co;2-r. URL http://dx.doi.org/10.1002/(sici)1097-0320(19981101)33:3%3C366::aid-cyto12%3E3.0.co;2-r.

Hamid Bolouri. *Computational Modelling Of Gene Regulatory Networks – A Primer*. Imperial College Press, 1 edition, August 2008. ISBN 1848162219. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1848162200.

Dragan Bonaki, Maximilian R. Odenbrett, Anton Wijs, Willem P. A. Ligtenberg, and Peter A. J. Hilbers. Efficient reconstruction of biological networks via transitive reduction on general purpose graphics processors. *BMC Bioinformatics*, 13:281, 2012.

Michael Boutros, Herv? Agaisse, and Norbert Perrimon. Sequential activation of signaling pathways during innate immune responses in drosophila. *Dev Cell*, 3(5):711–722, November 2002. ISSN 1534-5807. URL http://linkinghub.elsevier.com/retrieve/pii/S1534580702003258.

Pierre Brémaud. *An Introduction to Probabilistic Modeling*. Springer, 1998.

P. O. Brown and D. Botstein. Exploring the new world of the genome with DNA microarrays. *Nature Genetics*, 21:33–37, 1999.

Frank J Bruggeman and Hans V Westerhoff. The nature of systems biology. *Trends in microbiology*, 15(1):45–50, January 2007. ISSN 0966-842X. doi: 10.1016/j.tim.2006.11.003. URL http://www.ncbi.nlm.nih.gov/pubmed/17113776.

Helena Brunel, Joan-Josep Gallardo-Chacn, Alfonso Buil, Montserrat Vallverd, Jos Manuel Soria, Pere Caminal, and Alexandre Perera. Miss: a non-linear methodology based on mutual information for genetic association studies in both population and sib-pairs analysis. *Bioinformatics*, 26(15):1811–1818, 2010. doi: 10.1093/bioinformatics/btq273. URL http://bioinformatics.oxfordjournals.org/content/26/15/1811.abstract.

Wray Buntine. Theory refinement on bayesian networks. pages 52–60. Morgan Kaufmann, 1991.

Anne Carpenter, Thouis Jones, Michael Lamprecht, Colin Clarke, In Kang, Ola Friman, David Guertin, Joo Chang, Robert Lindquist, Jason Moffat, Polina Golland, and David Sabatini. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, 7(10):R100, 2006. ISSN 1465-6906. doi: 10.1186/gb-2006-7-10-r100. URL http://genomebiology.com/2006/7/10/R100.

Javier Carrera, Guillermo Rodrigo, Alfonso Jaramillo, and Santiago Elena. Reverse-engineering the arabidopsis thaliana transcriptional network under changing environmental conditions. *Genome Biology*, 10(9):R96, 2009. ISSN 1465-6906. doi: 10.1186/gb-2009-10-9-r96. URL http://genomebiology.com/2009/10/9/R96.

George Casella and Edward I. George. Explaining the Gibbs Sampler. *The American Statistician*, 46(3):167–174, 1992. ISSN 00031305. doi: 10.2307/2685208. URL http://dx.doi.org/10.2307/2685208.

Ethan G Cerami, Benjamin E Gross, Emek Demir, Igor Rodchenkov, zagin Babur, Nadia Anwar, Nikolaus Schultz, Gary D Bader, and Chris Sander. Pathway commons, a web resource for biological pathway data. *Nucleic Acids Research*, 39:D685–D690, 2011.

N Chapuis, J Tamburini, A S Green, L Willems, V Bardet, S Park, C Lacombe, P Mayeux, and D Bouscary. Perspectives on inhibiting mtor as a future treatment strategy for hematological malignancies. *Leukemia*, 24(10):1686–1699, 10 2010. URL http://dx.doi.org/10.1038/leu.2010.170.

R. Martin Chavez and Gregory F. Cooper. A randomized approximation algorithm for probabilistic inference on bayesian belief networks. *Networks*, 20(5):661–685, 1990. ISSN 1097-0037. doi: 10.1002/net.3230200510. URL http://dx.doi.org/10.1002/net.3230200510.

Jing Chen, Craig S. April, and Jian-Bing Fan. mirna expression profiling using illumina universal beadchips. *Next-Generation MicroRNA Expression Profiling Technology*, 822, 2012. URL http://www.springerprotocols.com/Abstract/doi/10.1007/978-1-61779-427-8_7.

Katherine C. Chen, Laurence Calzone, Attila Csikasz-Nagy, Frederick R. Cross, Bela Novak, and John J. Tyson. Integrative analysis of cell cycle control in budding yeast. *Molecular Biology of the Cell*, 15(8):3841–3862, 2004. doi: 10.1091/mbc.E03-11-0794. URL http://www.molbiolcell.org/content/15/8/3841.abstract.

Xiaowei Chen, Xiaobo Zhou, and Stephen T. C. Wong. Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy. *IEEE Transactions on Biomedical Engineering*, 2006:762–766, 2006.

D. M. Chickering, D. Geiger, and D. Heckerman. Learning Bayesian networks: Search methods and experimental results. In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, pages 112–128, January 1995.

David Chickering, Dan Geiger, and David Heckerman. Learning bayesian networks is np-hard. Technical report, 1994.

David M Chickering, David Heckerman, and Christopher Meek. A bayesian approach to learning bayesian networks with local structure. *Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence*, (MSR-TR-97-07):80–89,

1997. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.157.3189&rep=rep1&type=pdf.

Curtis R Chong and Pasi a Jänne. The quest to overcome resistance to egfr-targeted therapies in cancer. *Nature medicine*, 19(11):1389–400, 2013. URL http://www.ncbi.nlm.nih.gov/pubmed/24202392.

ChristophS. Clemen, Ludwig Eichinger, Vasily Rybakin, and TempleF. Smith. *Subcellular Biochemistry*, volume 48, pages 20–30. Springer New York, 2008. ISBN 978-0-387-09594-3. doi: 10.1007/978-0-387-09595-0{_}3. URL http://dx.doi.org/10.1007/978-0-387-09595-0_3.

Gregory F. Cooper and Tom Dietterich. A bayesian method for the induction of probabilistic networks from data. In *Machine Learning*, pages 309–347, 1992.

Gregory F. Cooper and Edward Herskovits. A bayesian method for constructing bayesian belief networks from databases. In *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*, UAI'91, pages 86–94, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc. ISBN 1-55860-203-8. URL http://dl.acm.org/citation.cfm?id=2100662.2100674.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991. ISBN 0-471-06259-6.

Rónón Daly, Qiang Shen, and Stuart Aitken. Learning bayesian networks: approaches and issues. *The Knowledge Engineering Review*, 26:99–157, 4 2011. ISSN 1469-8005. doi: 10.1017/S0269888910000251. URL http://journals.cambridge.org/article_S0269888910000251.

Sandra P. D'Angelo and M. Catherine Pietanza. The molecular pathogenesis of small cell lung cancer. *Cancer Biology & Therapy*, 10(1):1–10, 07 2010. URL http://www.landesbioscience.com/journals/cbt/article/12045/.

John S. Breese David Heckerman. Causal independence for probability assessment and inference using bayesian networks. *IEEE Trans. on Systems, Man and Cybernetics*, 1994.

Eric Davidson and Michael Levin. Gene regulatory networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(14):4935, 2005. doi: 10.1073/pnas.0502024102. URL http://www.pnas.org/content/102/14/4935.short.

Hidde de Jong. Qualitative simulation and related approaches for the analysis of dynamic systems. *Knowledge Eng. Review*, 19(2):93–132, 2004.

Riet De Smet and Kathleen Marchal. Advantages and limitations of current network inference methods. *Nature reviews. Microbiology*, 8(10):717–29, October 2010. ISSN 1740-1534. doi: 10.1038/nrmicro2419. URL http://www.ncbi.nlm.nih.gov/pubmed/20805835.

A. P. Dempster. Covariance selection. *Biometrics*, 28(1):pp. 157–175, 1972. ISSN 0006341X. URL http://www.jstor.org/stable/2528966.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39 (1):1–38, 1977.

Diego di Bernardo, Michael J Thompson, Timothy S Gardner, Sarah E Chobot, Erin L Eastwood, Andrew P Wojtovich, Sean J Elliott, Scott E Schaus, and James J Collins. Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. *Nat Biotech*, 23(3):377–383, March 2005. ISSN 1087-0156. URL http://dx.doi.org/10.1038/nbt1075.

Rui Dilao and Daniele Muraro. A software tool to model genetic regulatory networks. applications to the modeling of threshold phenomena and of spatial patterning in ¡italic¿drosophila¡/italic¿. *PLoS ONE*, 5(5):e10743, 05 2010. doi: 10.1371/journal.pone.0010743. URL http://dx.doi.org/10.1371%2Fjournal.pone.0010743.

G P Dimri, M Nakanishi, P Y Desprez, J R Smith, and J Campisi. Inhibition of e2f activity by the cyclin-dependent protein kinase inhibitor p21 in cells expressing or lacking a functional retinoblastoma protein. *Molecular and Cellular Biology*, 16(6):2987–97, 1996. URL http://mcb.asm.org/content/16/6/2987.abstract.

Sundar Dorai-Raj. binom: Binomial confidence intervals for several parameterizations. Technical report, Bioconductor R package (Version 1.1-1), 2014.

Pan Du, Warren A. Kibbe, and Simon M. Lin. lumi: a pipeline for processing illumina microarray. *Bioinformatics*, 24(13):1547–1548, 2008. doi: 10.1093/bioinformatics/btn224. URL http://bioinformatics.oxfordjournals.org/content/24/13/1547.abstract.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, New York, 2 edition, 2001.

Vjekoslav Dulic, Gretchen H. Stein, Dariush Farahi Far, Steven I. Reed, Mol Cell Biol, Vjekoslav Duli?, Gretchen H. Stein, Dariush Farahi Far, and Steven I. Reed. Nuclear accumulation of p21cip1 at the onset of mitosis: a role at the g2/m-phase transition. *Mol. Cell. Biol*, pages 546–557, 1998.

Patrik Dhaeseleer, Shoudan Liang, and Roland Somogyi. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8):707–726, 2000. doi: 10.1093/bioinformatics/16.8.707. URL http://bioinformatics.oxfordjournals.org/content/16/8/707.abstract.

Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998. URL http://www.pnas.org/content/95/25/14863.abstract.

B. Ellis. *Inference on Bayesian Network Structures*. Harvard University, 2006. URL http://books.google.de/books?id=Hmt_NAAACAAJ.

Laura L. Elo, Henna JŁrvenpŁŁ, Matej Ore?i?, Riitta Lahesmaa, and Tero Aittokallio. Systematic construction of gene coexpression networks with applications to human t helper cell differentiation process. *Bioinformatics*, 23(16):2096–2103, 2007. doi: 10.1093/bioinformatics/btm309. URL http://bioinformatics.oxfordjournals.org/content/23/16/2096.abstract.

Frank Emmert-Streib, Galina Glazko, Altay G?okmen, and Ricardo De Matos Simoes. Statistical inference and reverse engineering of gene regulatory networks from observational expression data. *Frontiers in Genetics*, 3(8), 2012. ISSN 1664-8021. doi: 10.3389/fgene.2012.00008. URL http://www.frontiersin.org/bioinformatics_and_computational_biology/10.3389/fgene.2012.00008/abstract.

Kevin M Esvelt and Harris H Wang. Genome-scale engineering for systems and synthetic biology. *Mol Syst Biol*, 9:–, January 2013. URL http://dx.doi.org/10.1038/msb.2012.66.

L. Evals, H. Sailem, P. Pascual Vargas, and C. Bakal. Inferring signalling networks from images. *Journal of Microscopy*, pages n/a–n/a, 2013. ISSN 1365-2818. doi: 10.1111/jmi.12062. URL http://dx.doi.org/10.1111/jmi.12062.

PL. Eyad Almasri, Chen Guanrao, and Dai Yang. Incorprating literatureknowledge in baysian network for inferring gene networks with gene expression data. In *Proceeding of the 4th International Symposium on Bioinformatics Research and Applications*, 2008.

Henrik Failmezger, Paurush Praveen, Achim Tresch, and Holger Frhlich. Learning gene network structure from time laps cell imaging in rnai knockdowns. *Bioinformatics*, 2013. doi: 10.1093/bioinformatics/btt179. URL http://bioinformatics.oxfordjournals.org/content/early/2013/04/17/bioinformatics.btt179.abstract.

Jeremiah J Faith, Boris Hayete, Joshua T Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J Collins, and Timothy S Gardner. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biol*, 5(1):e8, 01 2007. doi: 10.1371/journal.pbio.0050008. URL http://dx.doi.org/10.1371%2Fjournal.pbio.0050008.

Juan Pablo Fededa and Daniel W. Gerlich. Molecular control of animal cell cytokinesis. *NATURE CELL BIOLOGY*, 14(5):440–447, MAY 2012. ISSN 1465-7392. doi: {10.1038/ncb2482}.

Edward Feigenbaum and Pamela McCorduck. *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1983. ISBN 0-201-11519-0.

Soheil Feizi, Daniel Marbach, Muriel Médard, and Manolis Kellis. Network deconvolution as a general method to distinguish direct dependencies in networks. *Nature Biotechnology*, (8):726?733, 2013. doi: 10.1038/nbt.2635. URL http://www.nature.com/nbt/journal/v31/n8/full/nbt.2635.html.

Andrew Fire, SiQun Xu, Mary K. Montgomery, Steven A. Kostas, Samuel E. Driver, and Craig C. Mello. Potent and specific genetic interference by double-stranded rna in caenorhabditis elegans. *Nature*, 391(6669):806–811, February 1998. ISSN 0028-0836. URL http://dx.doi.org/10.1038/35888.

Ildikó Flesch and Peter Lucas. Markov equivalence in bayesian networks. *Advances in probabilistic graphical models*, pages 3–38, 2007.

J. E. Forde and T. C. Dale. Glycogen synthase kinase 3: A key regulator of cellular fate. 64 (15):1930–1944, 2007. doi: 10.1007/s00018-007-7045-7. URL http://dx.doi.org/10.1007/s00018-007-7045-7.

Brendan J Frey and Nebojsa Jojic. A comparison of algorithms for inference and learning in probabilistic graphical models. *IEEE transactions on pattern analysis and machine intelligence*, 27(9):1392–1416, 2005.

Nir Friedman. The bayesian structural em algorithm. In Gregory F Cooper and Serafin Moral, editors, *Computer*, volume 98, pages 129–138. Citeseer, 1998. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.30.2130&rep=rep1&type=pdf.

Nir Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805, 2004. doi: 10.1126/science.1094068. URL http://www.sciencemag.org/content/303/5659/799.abstract.

Nir Friedman and Daphne Koller. Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine Learning*, 50:95–125, 2003.

Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. In *Proceedings of the fourth annual international conference on Computational molecular biology*, RECOMB '00, pages 127–135, New York, NY, USA, 2000. ACM. ISBN 1-58113-186-0. doi: 10.1145/332306.332355. URL http://doi.acm.org/10.1145/332306.332355.

Holger Froehlich, Mark Fellmann, Holger Sueltmann, Annemarie Poustka, and Tim Beissbarth. Large scale statistical inference of signaling pathways from rnai and microarray data. *BMC Bioinformatics*, 8(1):386, 2007. ISSN 1471-2105. doi: 10.1186/1471-2105-8-386. URL http://www.biomedcentral.com/1471-2105/8/386.

H. Fröhlich, N. Speer, A. Poustka, and T. Beissbarth. Gosim–an r-package for computation of information theoretic go similarities between terms and gene products. *BMC Bioinformatics*, 8:166, 2007a. Frohlich, Holger Speer, Nora Poustka, Annemarie Beissbarth, Tim Research Support, Non-U.S. Gov't England BMC bioinformatics BMC Bioinformatics. 2007 May 22;8:166.

Holger Fröhlich, Mark Fellman, Holger Sültman, Annemarie Poustka, and Tim Beissbarth. Large scale statistical inference of singnaling pathways from rnai and microarray data. *BMC Bioinformatics*, 8(386), October 2007b.

Holger Fröhlich, Mark Fellman, Holger Sültman, and Tim Beissbarth. Predicting pathway membership via domain sinatures. *Bioinformatics*, 24:2137–2142, 2008a.

Holger Fröhlich, Mark Fellmann, Holger Sültmann, Annemarie Poustka, and Tim Beissbarth. Estimating large-scale signaling networks through nested effect models with intervention effects from microarray data. *Bioinformatics*, 24(22):2650–2656, 2008b. doi: 10.1093/bioinformatics/btm634. URL http://bioinformatics.oxfordjournals.org/content/24/22/2650.abstract.

Holger Fröhlich, Achim Tresch, and Tim Beissbarth. Nested effects models for learning signaling networks from perturbation data. *Biometrical Journal*, 51(2):304–323, 2009. ISSN 1521-4036. doi: 10.1002/bimj.200800185. URL http://dx.doi.org/10.1002/bimj.200800185.

Holger Fröhlich, Paurush Praveen, and Achim Tresch. Fast and efficient dynamic nested effects models. *Bioinformatics*, 27(2):238–244, 2011. URL http://www.ncbi.nlm.nih.gov/pubmed/21068003.

S. Gao and X. Wang. Quantitative utilization of prior biological knowledge in the bayesian network modeling of gene expression data. *BMC Bioinformatics*, 12:359, 2011.

Zoubin Ghahramani. Learning dynamic bayesian networks. In *Summer School on Neural Networks*, pages 168–197, 1997.

Estelle Glory and Robert F. Murphy. Automated subcellular location determination and high-throughput microscopy. *Developmental Cell*, 12(1):7 – 16, 2007. ISSN 1534-5807. doi: http://dx.doi.org/10.1016/j.devcel.2006.12.007. URL http://www.sciencedirect.com/science/article/pii/S1534580706005703.

Peter Goldstraw, David Ball, James R Jett, Thierry Le Chevalier, Eric Lim, Andrew G Nicholson, and Frances A Shepherd. Non-small-cell lung cancer. *The Lancet*, 378 (9804):1727–1740, 11 2011. URL http://linkinghub.elsevier.com/retrieve/pii/S0140673610621010.

A. Gonzalez Gonzalez, A. Naldi, L. Snchez, D. Thieffry, and C. Chaouiya. Ginsim: A software suite for the qualitative modelling, simulation and analysis of regulatory networks. *Biosystems*, 84(2):91 – 100, 2006. ISSN 0303-2647. doi: 10.1016/j.biosystems.2005.10.003. URL http://www.sciencedirect.com/science/article/pii/S0303264705001693. ¡ce:title¿Dynamical Modeling of Biological Regulatory Networks¡/ce:title¿.

Marco Grzegorczyk and Dirk Husmeier. Improving the structure mcmc sampler for bayesian networks by introducing a new edge reversal move. *Machine Learning*, 71: 265–305, 2008. ISSN 0885-6125. doi: 10.1007/s10994-008-5057-7. URL http://dx.doi.org/10.1007/s10994-008-5057-7.

David A. Guertin, Deanna M. Stevens, Maki Saitoh, Stephanie Kinkel, Katherine Crosby, Joon-Ho Sheen, David J. Mullholland, Mark A. Magnuson, Hong Wu, and David M. Sabatini. mtor complex 2 is required for the development of prostate cancer induced by pten loss in mice. *Cancer cell*, 15(2):148–159, 02 2009. URL http://linkinghub.elsevier.com/retrieve/pii/S1535610808004364.

J. Silvio Gutkind. Regulation of mitogen-activated protein kinase signaling networks by g protein-coupled receptors. *Sci. STKE*, 2000(40):re1, 2000. doi: 10.1126/stke.2000.40.re1. URL http://stke.sciencemag.org/cgi/content/abstract/sigtrans;2000/40/re1.

Dana M. Gwinn, David B. Shackelford, Daniel F. Egan, Maria M. Mihaylova, Annabelle Mery, Debbie S. Vasquez, Benjamin E. Turk, and Reuben J. Shaw. Ampk phosphorylation of raptor mediates a metabolic checkpoint. *Molecular cell*, 30(2):214–226, 04 2008. URL http://linkinghub.elsevier.com/retrieve/pii/S109727650800169X.

Hendrik Hache, Hans Lehrach, and Ralf Herwig. Reverse engineering of gene regulatory networks: A comparative study. *EURASIP J. Bioinformatics and Systems Biology*, 2009, 2009.

Annett Hahn-Windgassen, Veronique Nogueira, Chia-Chen Chen, Jennifer E. Skeen, Nahum Sonenberg, and Nissim Hay. Akt activates the mammalian target of rapamycin by regulating cellular atp level and ampk activity. *Journal of Biological Chemistry*, 280 (37):32081–32089, 2005. doi: 10.1074/jbc.M502876200. URL http://www.jbc.org/content/280/37/32081.abstract.

Florian Hahne, Alexander Mehrle, Dorit Arlt, Annemarie Poustka, Stefan Wiemann, and Tim Beißbarth. Extending pathways based on gene lists using interpro domain signatures. *BMC Bioinformatics*, 9(1):3, 2008. ISSN 1471-2105. doi: 10.1186/1471-2105-9-3. URL http://www.biomedcentral.com/1471-2105/9/3.

Gregory J. Hannon. Rna interference. *Nature*, 418(6894):244–251, July 2002. ISSN 0028-0836. URL http://dx.doi.org/10.1038/418244a.

D Hardie and Dario Alessi. Lkb1 and ampk and the cancer-metabolism link - ten years after. *BMC Biology*, 11(1):36, 2013. ISSN 1741-7007. doi: 10.1186/1741-7007-11-36. URL http://www.biomedcentral.com/1741-7007/11/36.

Alexander J Hartemink. Reverse engineering gene regulatory networks. *Nature biotechnology*, 23(5):554–5, May 2005. ISSN 1087-0156. doi: 10.1038/nbt0505-554. URL http://www.ncbi.nlm.nih.gov/pubmed/15877071.

Alexander J. Hartemink, David K. Gifford, Tommi Jaakkola, and Richard A. Young. Bayesian methods for elucidating genetic regulatory networks. *IEEE Intelligent Systems*, 17(2):37–43, 2002a.

Alexander J. Hartemink, David K. Gifford, Tommi Jaakkola, and Richard A. Young. Combining location and expression data for principled discovery of genetic regulatory network models. In *Pacific Symposium on Biocomputing*, pages 437–449, 2002b.

Simon Hawley, Jerome Boudeau, Jennifer Reid, Kirsty Mustard, Lina Udd, Tomi Makela, Dario Alessi, and D Grahame Hardie. Complexes between the lkb1 tumor suppressor, stradalpha/beta and mo25alpha/beta are upstream kinases in the amp-activated protein kinase cascade. *Journal of Biology*, 2(4):28, 2003. ISSN 1475-4924. doi: 10.1186/1475-4924-2-28. URL http://jbiol.com/content/2/4/28. Correspondence regarding LKB1 should be addressed to Dario Alessi and regarding AMPK to Grahame Hardie.

D Heckerman, D Geiger, and D M Chickering. Learning bayesian networks - the combination of knowledge and statistical-data. *Machine Learning*, 20(3):197–243, 1995a. ISSN 08856125. URL http://www.vanderbilt.edu/viibre/members/documents/12454-Heckerman-ML-Preprint-1995.pdf.

David Heckerman, Dan Geiger, and David M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, September 1995b. ISSN 0885-6125. doi: 10.1007/BF00994016. URL http://link.springer.com/10.1007/BF00994016.

David Heckerman, E. H. Mamdani, and Michael P. Wellman. Real-world applications of bayesian networks - introduction. *Commun. ACM*, 38(3):24–26, 1995c.

Alireza Sadeghi Hesar, Hamid Tabatabaee, and Mehrdad Jalali. Structure learning of bayesian networks using heuristic methods. In *International Conference on Information and Knowledge Management*, volume 45, pages 246–250, Singapore, 2012.

Thomas P. Hettmansperger. *Robust nonparametric statistical methods*. CRC Press, 2011. ISBN 9781439809082. URL http://www.worldcat.org/isbn/9781439809082.

A. Hoffmann, A. Levchenko, M. L. Scott, and D. Baltimore. The i?b-nf-?b signaling module: temporal control and selective gene activation. *Science*, 2002. doi: 10.1126/science.1071914.

Caitriona Holohan, Sandra Van Schaeybroeck, Daniel B Longley, and Patrick G Johnston. Cancer drug resistance: an evolving paradigm. *Nature reviews. Cancer*, 13(10):714–26, 2013. URL http://www.ncbi.nlm.nih.gov/pubmed/24060863.

Steve Horvath and Jun Dong. Geometric interpretation of gene coexpression network analysis. *PLoS Comput Biol*, 4(8):e1000117, 08 2008. doi: 10.1371/journal.pcbi.1000117. URL http://dx.doi.org/10.1371%2Fjournal.pcbi.1000117.

Dirk Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks. *Bioinformatics*, 19(17): 2271–2282, 2003. doi: 10.1093/bioinformatics/btg313. URL http://bioinformatics.oxfordjournals.org/content/19/17/2271.abstract.

Trey E. Ideker, Vesteinn Thorsson, and Richard M. Karp. Discovery of regulatory interactions through perturbation: Inference and experimental design. In *Proceedings of the Pacific Symposium on Biocomputing*. World Scientific Press, 2000.

Marcin Imielinski, Alice H. Berger, Peter S. Hammerman, Bryan Hernandez, Trevor J. Pugh, Eran Hodis, Jeonghee Cho, James Suh, Marzia Capelletti, Andrey Sivachenko, Carrie Sougnez, Daniel Auclair, Michael S. Lawrence, Petar Stojanov, Kristian Cibulskis, Kyusam Choi, Luc de Waal, Tanaz Sharifnia, Angela Brooks, Heidi Greulich, Shantanu Banerji, Thomas Zander, Danila Seidel, Frauke Leenders, Sascha AnsÈn, Corinna Ludwig, Walburga Engel-Riedel, Erich Stoelben, J?rgen Wolf, Chandra Goparju, Kristin Thompson, Wendy Winckler, David Kwiatkowski, Bruce E. Johnson, Pasi A. Jenne, Vincent A. Miller, William Pao, William D. Travis, Harvey I. Pass, Stacey B. Gabriel, Eric S. Lander, Roman K. Thomas, Levi A. Garraway, Gad Getz, and Matthew Meyerson. Mapping the hallmarks of lung adenocarcinoma with massively parallel sequencing. *Cell*, 150(6):1107–1120, 09 2012. URL http://linkinghub.elsevier.com/retrieve/pii/S0092867412010616.

S. Imoto, T. Goto, and S. Miyano. Estimation of genetic networks and functional structures between genes by using bayesian networks and nonparametric regression. *Pacific Symposium on Biocomputing*, 2002:175–86, 2002.

S. Imoto, Y. Tamada, S. Miyano, K. Yashuda, CG. Print, DS. Charnock-Jones, D. Sanders, CJ. Savoie, and K. Tashiro. Computational strategy for discovering druggable gene networks from genome wide rna expression profile. *Pacific Symposium on Biocomputing*, pages 559–571, 2006.

F Innocenti. Polypharmacology in drug discovery. *Clin Pharmacol Ther*, 92(3):279–280, 09 2012. URL http://dx.doi.org/10.1038/clpt.2012.129.

N Ivanova, R Dobrin, R Lu, I Kotenko, J Levorse, C DeCoste, X Schafer, Y Lun, and I R Lemischka. Dissecting self-renewal in stem cells with rna interference. *Nature*, 442 (7102):533–538, August 2006. doi: 10.1038/nature04915. URL http://www.ncbi.nlm.nih.gov/pubmed/16767105.

Katherine James, Anil Wipat, and Jennifer Hallinan. *Integration of Full-Coverage Probabilistic Functional Networks with Relevance to Specific Biological Processes*. DILS '09. Springer-Verlag, Berlin, Heidelberg, 2009.

Marc Johannes, Holger Fröhlich, Holger Sültmann, and Tim Beissbarth. pathclass: an r-package for integration of pathway knowledge into support vector machines for biomarker discovery. *Bioinformatics*, 27(10):1442–1443, May 2011. doi: 10.1093/bioinformatics/btr157. URL http://dx.doi.org/10.1093/bioinformatics/btr157.

W. Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, 2007. doi: 10.1093/biostatistics/kxj037. URL http://biostatistics.oxfordjournals.org/content/8/1/118.abstract.

K. Junker and I. Petersen. Kleinzelliges lungenkarzinom. *Der Pathologe*, 30(2):131–140, 2009. ISSN 0172-8113. doi: 10.1007/s00292-008-1115-y. URL http://dx.doi.org/10.1007/s00292-008-1115-y.

Lars Kaderali, Eva Dazert, Ulf Zeuge, Michael Frese, and Ralf Bartenschlager. Reconstructing signaling pathways from rnai data using probabilistic boolean threshold networks. *Bioinformatics*, 25(17):2229–2235, 2009.

Lee Kamentsky, Thouis R. Jones, Adam Fraser, Mark-Anthony Bray, David J. Logan, Katherine L. Madden, Vebjorn Ljosa, Curtis Rueden, Kevin W. Eliceiri, and Anne E. Carpenter. Improved structure, function and compatibility for cellprofiler. *Bioinformatics*, 27(8):1179–1180, April 2011. ISSN 1367-4803. doi: 10.1093/bioinformatics/btr095. URL http://dx.doi.org/10.1093/bioinformatics/btr095.

Minoru Kanehisa, Susumu Goto, Yoko Sato, Masayuki Kawashima, Miho Furumichi, and Mao Tanabe. Data, information, knowledge and principle: back to metabolism in kegg. *Nucleic Acids Research*, 42(D1):D199–D205, 2014. doi: 10.1093/nar/gkt1076. URL http://nar.oxfordjournals.org/content/42/D1/D199.abstract.

Kenneth J Kauffman, Purusharth Prakash, and Jeremy S Edwards. Advances in flux balance analysis. *Current Opinion in Biotechnology*, 14(5):491 – 496, 2003. ISSN 0958-1669. doi: 10.1016/j.copbio.2003.08.001. URL http://www.sciencedirect.com/science/article/pii/S0958166903001174.

Stuart A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.

Audrey Kauffmann, Robert Gentleman, and Wolfgang Huber. arrayqualitymetrics?a bioconductor package for quality assessment of microarray data. *Bioinformatics*, 25(3):415–416, 2009. doi: 10.1093/bioinformatics/btn647. URL http://bioinformatics.oxfordjournals.org/content/25/3/415.abstract.

Boris N Kholodenko. Cell-signalling dynamics in time and space. *Nature reviews. Molecular cell biology*, 7(3):165–76, March 2006. ISSN 1471-0072. doi: 10.1038/nrm1838. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1679905&tool=pmcentrez&rendertype=abstract.

Sun Yong Kim, Seiya Imoto, and Satoru Miyano. Inferring gene networks from time series microarray data using dynamic bayesian networks. *Briefings in bioinformatics*, 4(3):228–235, 2003.

Hiroaki Kitano. overview computational systems biology. *Nature*, 420:206–210, 2002. doi: doi:10.1038/nature01254.

Hiroaki Kitano. Biological robustness. *Nat Rev Genet*, 5(11):826–837, November 2004. ISSN 1471-0056. URL http://dx.doi.org/10.1038/nrg1471.

Lisa Kockeritz, Bradley Doble, Satish Patel, and James R Woodgett. Glycogen synthase kinase-3–an overview of an over-achieving protein kinase. *Current drug targets*, 7(11):1377–1388, 2006.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

Natalia L. Komarova, Xiufen Zou, Qing Nie, and Lee Bardwell. A theoretical framework for specificity in cell signaling. *Mol Syst Biol*, 16:E1–E5, 2005.

R I Kondor and J Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the ICML*, 2002.

Frank R. Kschischang, Brendan J. Frey, and H-A Loeliger. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2):498–519, 2001.

Harri Lähdesmäki, Ilya Shmulevich, and Olli Yli-Harja. On learning gene regulatory networks under the boolean network model. In *Machine Learning*, pages 147–167, 2003.

The Lancet. Lung cancer: a global scourge. *Lancet*, 382(9893):659–, August 2013. ISSN 0140-6736. URL http://linkinghub.elsevier.com/retrieve/pii/S0140673613617596.

Peter Langfelder and Steve Horvath. Eigengene networks for studying the relationships between co-expression modules. *BMC Systems Biology*, 1(1):54, 2007. ISSN 1752-0509. doi: 10.1186/1752-0509-1-54. URL http://www.biomedcentral.com/1752-0509/1/54.

Peter Larsen, Eyad Almasri, Guanrao Chen, and Yang Dai. A statistical method to incorporate biological knowledge for generating testable novel gene regulatory interactions from microarray experiments. *BMC Bioinformatics*, 8(1):317, 2007. ISSN 1471-2105. doi: 10.1186/1471-2105-8-317. URL http://www.biomedcentral.com/1471-2105/8/317.

Joshua Lederberg and Alexa T. Mccray. Ome sweet 'omics– a genealogical treasury of words. *The Scientist*, 17(7), 2001.

Homin K. Lee, Amy K. Hsu, Jon Sajdak, Jie Qin, and Paul Pavlidis. Coexpression analysis of human genes across many microarray data sets. *Genome Research*, 14(6):1085–1094, 2004. doi: 10.1101/gr.1910904. URL http://genome.cshlp.org/content/14/6/1085.abstract.

Wei-Po Lee and Wen-Shyong Tzou. Computational methods for discovering gene networks from expression data. *Briefings in bioinformatics*, 10(4):408–23, July 2009. ISSN 1477-4054. doi: 10.1093/bib/bbp028. URL http://www.ncbi.nlm.nih.gov/pubmed/19505889.

Karen Lemmens, Tijl De Bie, Thomas Dhollander, Sigrid De Keersmaecker, Inge Thijs, Geert Schoofs, Ami De Weerdt, Bart De Moor, Jos Vanderleyden, Julio Collado-Vides, Kristof Engelen, and Kathleen Marchal. Distiller: a data integration framework to reveal condition dependency of complex regulons in escherichia coli. *Genome Biology*, 10(3):R27, 2009. ISSN 1465-6906. doi: 10.1186/gb-2009-10-3-r27. URL http://genomebiology.com/2009/10/3/R27.

Fangting Li, Tao Long, Ying Lu, Qi Ouyang, and Chao Tang. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, 101(14):4781–4786, 2004. doi: 10.1073/pnas.0305937101. URL http://www.pnas.org/content/101/14/4781.abstract.

Shenghua Li, Paul Brazhnik, Bruno Sobral, and John J Tyson. A quantitative study of the division cycle of ¡named-content xmlns:xlink="http://www.w3.org/1999/xlink" content-type="genus-species" xlink:type="simple"¿caulobacter crescentus¡/named-content¿ stalked cells. *PLoS Comput Biol*, 4(1):e9, 01 2008. doi: 10.1371/journal.pcbi.0040009. URL http://dx.plos.org/10.1371%2Fjournal.pcbi.0040009.

D Lin. An information-theoretic definition of similarity. In Morgan Kaufmann, editor, *Proceedings of the 15th International Conference on Machine Learning*, volume 1, pages 296–304, San Francisco, CA, 1998.

Zhi-Ping Liu, Wanwei Zhang, Katsuhisa Horimoto, and Luonan Chen. A gaussian graphical model for identifying significantly responsive regulatory networks from time series gene expression data. In *IEEE 6th International Conference on Systems Biology (ISB)*, 2012.

James C W Locke, Megan M Southern, Laszlo Kozma-Bognar, Victoria Hibberd, Paul E Brown, Matthew S Turner, and Andrew J Millar. Extension of a genetic network model by iterative experimentation and mathematical analysis. *Mol Syst Biol*, 1:–, June 2005. URL `http://dx.doi.org/10.1038/msb4100018`.

David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

Lesley T. MacNeil and Albertha J.M. Walhout. Gene regulatory networks and the role of robustness and stochasticity in the control of gene expression. *Genome Research*, 21(5): 645–657, 2011. doi: 10.1101/gr.097378.109. URL `http://genome.cshlp.org/content/21/5/645.abstract`.

David Madigan and Fred Hutchinson. Enhancing the predictive performance of bayesian graphical models. In *Communications in Statistics Theory and Methods*, 1995.

Norberto Malpica, Carlos Ortiz de Solorzano, Juan Jose Vaquero, Andres Santos, Isabel Vallcorba, Jose Miguel Garcia-Sagredo, and Francisco del Pozo. Applying watershed algorithms to the segmentation of clustered nuclei. *Cytometry*, 28(4):289–297, 1997. ISSN 1097-0320. doi: 10.1002/(SICI)1097-0320(19970801)28:4⟨289::AID-CYTO3⟩3.0.CO;2-7. URL `http://dx.doi.org/10.1002/(SICI)1097-0320(19970801)28:4<289::AID-CYTO3>3.0.CO;2-7`.

Daniel Marbach, James C Costello, Robert Küffner, Nicole M Vega, Robert J Prill, Diogo M Camacho, Kyle R Allison, The Dream Consortium, Manolis Kellis, James J Collins, and Gustavo Stolovitzky. Wisdom of crowds for robust gene network inference. *Nature Methods*, 9(8), 2012. doi: 10.1038/nMeth.2016.

Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7 Suppl 1:S7, 2006. doi: 10.1186/1471-2105-7-S1-S7. URL `http://dx.doi.org/10.1186/1471-2105-7-S1-S7`.

Florian Markowetz. *Probabilistic Models for Gene Silencing Data*. PhD thesis, Free University of Berlin, 2005.

Florian Markowetz and Rainer Spang. Evaluating the effect of perturbations in reconstructing network topologies. *DSC Working Papers*, pages 1–7, 2003.

Florian Markowetz and Rainer Spang. Inferring cellular networks–a review. *BMC bioinformatics*, 8 Suppl 6:S5, January 2007. ISSN 1471-2105. doi: 10.1186/1471-2105-8-S6-S5. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1995541&tool=pmcentrez&rendertype=abstract`.

C.J Marshall. Specificity of receptor tyrosine kinase signaling: Transient versus sustained extracellular signal-regulated kinase activation. *Cell*, 80(2):179–185, January 1995. ISSN 0092-8674. URL http://linkinghub.elsevier.com/retrieve/pii/0092867495904018.

Ryo Matoba, Hitoshi Niwa, Shinji Masui, Satoshi Ohtsuka, Mark G. Carter, Alexei A. Sharov, and Minoru S.H. Ko. Dissecting oct3/4-regulated gene networks in embryonic stem cells by expression profiling. *PLoS ONE*, 1(1):e26, 12 2006. doi: 10.1371/journal.pone.0000026. URL http://dx.plos.org/10.1371/journal.pone.0000026.

Mary McMahon, Veronica Aylln, Kostya I. Panov, and Rosemary O'Connor. Ribosomal 18s rna processing by the igf-i-responsive wdr3 protein is integrated with p53 function in cancer cell proliferation. *Journal of Biological Chemistry*, 2010. doi: 10.1074/jbc.M110.108555. URL http://www.jbc.org/content/early/2010/04/14/jbc.M110.108555.abstract.

Patrick Meyer, Frederic Lafitte, and Gianluca Bontempi. minet: A r/bioconductor package for inferring large transcriptional networks using mutual information. *BMC Bioinformatics*, 9(1):461, 2008. ISSN 1471-2105. doi: 10.1186/1471-2105-9-461. URL http://www.biomedcentral.com/1471-2105/9/461.

Jason Moffat, Dorre A. Grueneberg, Xiaoping Yang, So Young Kim, Angela M. Kloepfer, Gregory Hinkle, Bruno Piqani, Thomas M. Eisenhaure, Biao Luo, Jennifer K. Grenier, Anne E. Carpenter, Shi Yin Foo, Sheila A. Stewart, Brent R. Stockwell, Nir Hacohen, William C. Hahn, Eric S. Lander, David M. Sabatini, and David E. Root. A lentiviral {RNAi} library for human and mouse genes applied to an arrayed viral high-content screen. *Cell*, 124(6):1283 – 1298, 2006. ISSN 0092-8674. doi: http://dx.doi.org/10.1016/j.cell.2006.01.040. URL http://www.sciencedirect.com/science/article/pii/S0092867406002388.

E. R. Morrissey, M. A. Jurez, K. J. Denby, and N. J. Burroughs. On reverse engineering of gene interaction networks using time course data with repeated measurements. *Bioinformatics*, 26(18):2305–2312, 2010. doi: 10.1093/bioinformatics/btq421. URL http://bioinformatics.oxfordjournals.org/content/26/18/2305.abstract.

Claudius Mueller, Lance A. Liotta, and Virginia Espina. Reverse phase protein microarrays advance to use in clinical trials. *Molecular oncology*, 4(6):461–481, 12 2010. URL http://linkinghub.elsevier.com/retrieve/pii/S1574789110001018?showall=true.

Sach Mukherjee and Terence P. Speed. Network inference using informative priors. *Proceedings of the National Academy of Sciences*, 105(38):14313–14318, 2008.

Nicola J Mulder, Rolf Apweiler, Terri K Attwood, Amos Bairoch, Alex Bateman, David Binns, Margaret Biswas, Paul Bradley, Peer Bork, Phillip Bucher, Richard Copley, Emmanuel Courcelle, Richard Durbin, Laurent Falquet, Wolfgang Fleischmann, Jerome Gouzy, Sam Griffith-Jones, Daniel Haft, Henning Hermjakob, Nicolas Hulo, Daniel Kahn, Alexander Kanapin, Maria Krestyaninova, Rodrigo Lopez, Ivica Letunic, Sandra Orchard, Marco Pagni, David Peyruc, Chris P Ponting, Florence Servant, Christian J A Sigrist, and InterPro Consortium. Interpro: an integrated documentation resource for protein families, domains and functional sites. *Brief Bioinform*, 3(3):225–235, Sep 2002.

Shannon M Mumenthaler, Jasmine Foo, Kevin Leder, Nathan C Choi, David B Agus, William Pao, Parag Mallick, and Franziska Michor. Evolutionary modeling of combination treatment strategies to overcome resistance to tyrosine kinase

inhibitors in non-small cell lung cancer. *Molecular pharmaceutics*, 8(6):2069–79, 2011. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3230244&tool=pmcentrez&rendertype=abstract.

Kevin Murphy and Saira Mian. Modelling gene expression data using dynamic bayesian networks. Technical report, 1999.

Kevin Patrick Murphy. Dynamic bayesian networks: Representation, inference and learning, 2002.

Leon O. Murphy and John Blenis. Mapk signal specificity: the right place at the right time. *Trends in Biochemical Sciences*, 31(5):268 – 275, 2006. ISSN 0968-0004. doi: 10.1016/j.tibs.2006.03.009. URL http://www.sciencedirect.com/science/article/pii/S0968000406000867.

N Nariai, S. Kim, S Imoto, and S. Miyano. Using protein-protein interaction for refining gene networks estimated from microarray data by bayesian networks. In *Pacific Symposium on Biocomputing*, 2004.

Richard E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2003.

Chris J Needham, James R Bradford, Andrew J Bulpitt, and David R Westhead. A primer on learning in bayesian networks for computational biology. *PLoS Comput Biol*, 3(8):e129, 08 2007. doi: 10.1371/journal.pcbi.0030129. URL http://dx.doi.org/10.1371%2Fjournal.pcbi.0030129.

Sven Nelander, Weiqing Wang, Bjorn Nilsson, Qing-Bai She, Christine Pratilas, Neal Rosen, Peter Gennemark, and Chris Sander. Models from experiments: combinatorial drug perturbations of cancer cells. *Mol Syst Biol*, 4:–, September 2008. URL http://dx.doi.org/10.1038/msb.2008.53.

D.L. Nelson, A.L. Lehninger, and M.M. Cox. *Lehninger Principles of Biochemistry*. Lehninger Principles of Biochemistry. W. H. Freeman, 2008. ISBN 9780716771081. URL http://books.google.de/books?id=5Ek9J4p3NfkC.

Beate Neumann, Michael Held, Urban Liebel, Holger Erfle, Phill Rogers, Rainer Pepperkok, and Jan Ellenberg. High-throughput rnai screening by time-lapse imaging of live human cells. *Nat Meth*, 3(5):385–390, May 2006. ISSN 1548-7091. URL http://dx.doi.org/10.1038/nmeth876.

Beate Neumann, Thomas Walter, Jean-Karim K. Hériché, Jutta Bulkescher, Holger Erfle, Christian Conrad, Phill Rogers, Ina Poser, Michael Held, Urban Liebel, Cihan Cetin, Frank Sieckmann, Gregoire Pau, Rolf Kabbe, Annelie Wünsche, Venkata Satagopam, Michael H. Schmitz, Catherine Chapuis, Daniel W. Gerlich, Reinhard Schneider, Roland Eils, Wolfgang Huber, Jan-Michael M. Peters, Anthony A. Hyman, Richard Durbin, Rainer Pepperkok, and Jan Ellenberg. Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. *Nature*, 464(7289):721–727, April 2010. ISSN 1476-4687. doi: 10.1038/nature08869. URL http://dx.doi.org/10.1038/nature08869.

Huck-Hui Ng and M. Azim Surani. The transcriptional and signalling networks of pluripotency. *Nat Cell Biol*, 13(5):490–496, May 2011. ISSN 1465-7392. URL http://dx.doi.org/10.1038/ncb0511-490.

Theresa Niederberger, Stefanie Etzold, Michael Lidschreiber, Kerstin C. Maier, Dietmar E. Martin, Holger Fr??hlich, Patrick Cramer, and Achim Tresch. Mc eminem maps the interaction landscape of the mediator. *PLoS Comput Biol*, 8(6):e1002568, 06 2012. doi: 10.1371/journal.pcbi.1002568. URL http://dx.doi.org/10.1371%2Fjournal.pcbi.1002568.

O. Nir, C. Bakal, N. Perrimon, and B. Berger. Inference of rhogap/gtpase regulation using single-cell morphological data from a combinatorial rnai screen. *Genome Res*, 20(3): 372–80, 2010.

Takeshi Obayashi, Yasunobu Okamura, Satoshi Ito, Shu Tadaka, Ikuko N. Motoike, and Kengo Kinoshita. Coxpresdb: a database of comparative gene coexpression networks of eleven species for mammals. *Nucleic Acids Research*, 41(D1):D1014–D1020, 2013. doi: 10.1093/nar/gks1014. URL http://nar.oxfordjournals.org/content/41/D1/D1014.abstract.

Yoshikazu Ohya, Jun Sese, Masashi Yukawa, Fumi Sano, Yoichiro Nakatani, Taro L. Saito, Ayaka Saka, Tomoyuki Fukuda, Satoru Ishihara, Satomi Oka, Genjiro Suzuki, Machika Watanabe, Aiko Hirata, Miwaka Ohtani, Hiroshi Sawai, Nicolas Fraysse, Jean-Paul Latg, Jean M. Franois, Markus Aebi, Seiji Tanaka, Sachiko Muramatsu, Hiroyuki Araki, Kintake Sonoike, Satoru Nogami, and Shinichi Morishita. High-dimensional and large-scale phenotyping of yeast mutants. *Proceedings of the National Academy of Sciences of the United States of America*, 102(52):19015–19020, 2005. doi: 10.1073/pnas.0509436102. URL http://www.pnas.org/content/102/52/19015.abstract.

Michael C. Oldham, Steve Horvath, and Daniel H. Geschwind. Conservation and evolution of gene coexpression networks in human and chimpanzee brains. *Proceedings of the National Academy of Sciences*, 103(47):17973–17978, 2006. doi: 10.1073/pnas.0605938103. URL http://www.pnas.org/content/103/47/17973.abstract.

Irene M. Ong, Jeremy D. Glasner, and David Page. Modelling regulatory pathways in e. coli from time series expression profiles. *Bioinformatics*, 18(suppl 1):S241–S248, 2002. doi: 10.1093/bioinformatics/18.suppl_1.S241. URL http://bioinformatics.oxfordjournals.org/content/18/suppl_1/S241.abstract.

Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979. doi: 10.1109/TSMC.1979.4310076. URL http://dx.doi.org/10.1109/TSMC.1979.4310076.

J Guillermo Paez, Pasi A Jnne, Jeffrey C Lee, Sean Tracy, Heidi Greulich, Stacey Gabriel, Paula Herman, Frederic J Kaye, Neal Lindeman, Titus J Boggon, and et al. Egfr mutations in lung cancer: correlation with clinical response to gefitinib therapy. *Science (New York, N.Y.)*, 304(5676):1497–1500, 2004.

Bernhard Palsson. *Systems biology : properties of reconstructed networks*. Cambridge University Press, Cambridge, New York, 2006. ISBN 0-521-85903-4. URL http://opac.inria.fr/record=b1121203.

Philip R. O. Payne. Chapter 1: Biomedical knowledge integration. *PLoS Comput Biol*, 8 (12):e1002826, 12 2012. doi: 10.1371/journal.pcbi.1002826. URL http://dx.doi.org/10.1371%2Fjournal.pcbi.1002826.

Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc, San Francisco, 1 edition, 1988.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, March 2000. ISBN 0521773628. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0521773628.

Juan M. Pedraza and Alexander van Oudenaarden. Noise propagation in gene networks. *Science*, 307(5717):1965–1969, 2005. doi: 10.1126/science.1109090. URL http://www.sciencemag.org/content/307/5717/1965.abstract.

Dana Peér and Nir Hacohen. Principles and strategies for developing network models in cancer. *Cell*, 144(6):864–73, 2011. ISSN 1097-4172. URL http://www.biomedsearch.com/nih/Principles-strategies-developing-network-models/21414479.html.

Dana Peér, Aviv Regev, Gal Elidan, and Nir Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17(suppl 1):S215–S224, 2001. doi: 10.1093/bioinformatics/17.suppl_1.S215. URL http://bioinformatics.oxfordjournals.org/content/17/suppl_1/S215.abstract.

Christopher A. Penfold and David L. Wild. How to infer gene networks from expression profiles, revisited. *Interface Focus*, 1(6):857–870, 2011. doi: 10.1098/rsfs.2011.0053. URL http://rsfs.royalsocietypublishing.org/content/1/6/857.abstract.

Zachary E. Perlman, Michael D. Slack, Yan Feng, Timothy J. Mitchison, Lani F. Wu, and Steven J. Altschuler. Multidimensional drug profiling by automated microscopy. *Science*, 306(5699):1194–1198, 2004. doi: 10.1126/science.1100709. URL http://www.sciencemag.org/content/306/5699/1194.abstract.

Bruno-Edouard Perrin, Liva Ralaivola, Aurlien Mazurie, Samuele Bottani, Jacques Mallet, and Florence dAlchBuc. Gene networks inference using dynamic bayesian networks. *Bioinformatics*, 19(suppl 2):ii138–ii148, 2003. doi: 10.1093/bioinformatics/btg1071. URL http://bioinformatics.oxfordjournals.org/content/19/suppl_2/ii138.abstract.

Mariaelena Pierobon, Claudio Belluco, LanceA. Liotta, and III Petricoin, EmanuelF. Reverse phase protein microarrays for clinical applications. In Ulrike Korf, editor, *Protein Microarrays*, volume 785 of *Methods in Molecular Biology*, pages 3–12. Humana Press, 2011. ISBN 978-1-61779-285-4. doi: 10.1007/978-1-61779-286-1_1. URL http://dx.doi.org/10.1007/978-1-61779-286-1_1.

Stan Pounds and Stephan W. Morris. Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distribution of p-values. *Bioinformatics*, 19(10):1236–1242, 2003. doi: 10.1093/bioinformatics/btg148. URL http://bioinformatics.oxfordjournals.org/content/19/10/1236.abstract.

Paurush Praveen and Holger Fröhlich. Boosting probabilistic graphical model inference by incorporating prior knowledge from multiple sources. *PLoS ONE*, 8(6):e67410, 06 2013. doi: 10.1371/journal.pone.0067410. URL http://dx.doi.org/10.1371%2Fjournal.pone.0067410.

Angela Presson, Eric Sobel, Jeanette Papp, Charlyn Suarez, Toni Whistler, Mangalathu Rajeevan, Suzanne Vernon, and Steve Horvath. Integrated weighted gene co-expression network analysis with an application to chronic fatigue syndrome. *BMC Systems Biology*, 2(1):95, 2008. ISSN 1752-0509. doi: 10.1186/1752-0509-2-95. URL http://www.biomedcentral.com/1752-0509/2/95.

B. Raghavachari, A. Tasneem, T. M. Przytycka, and R. Jothi. Domine: a database of protein domain interactions. *Nucleic Acids Res*, 36(Database issue):D656–61, 2008.

M Raman, W Chen, and M H Cobb. Differential regulation and properties of mapks. *Oncogene*, 26(22):3100–3112, 2007. ISSN 0950-9232. URL http://dx.doi.org/10.1038/sj.onc.1210392.

Norbert Perrimon Ramanuj Dasgupta. Using rnai to catch drosophila genes in a web of interactions: insights into cancer research. *Oncogene*, (51):8359?8365, 2004. doi: 10.1038/sj.onc.1208028. URL http://www.nature.com/onc/journal/v23/n51/full/1208028a.html.

Sridhar Rao and Stuart Orkin. Unraveling the transcriptional network controlling es cell pluripotency. *Genome Biology*, 7(8):230, 2006. ISSN 1465-6906. doi: 10.1186/gb-2006-7-8-230. URL http://genomebiology.com/2006/7/8/230.

Andrea Rau, Florence Jaffrzic, Jean-Louis Foulley, and Rebecca W Doerge. An Empirical Bayesian Method for Estimating Biological Networks from Temporal Microarray Data. *Statistical Applications in Genetics and Molecular Biology*, 9, xx 2010. doi: 10.2202/1544-6115.1513. URL http://dx.doi.org/10.2202/1544-6115.1513.

Martin Reck, David F Heigener, Tony Mok, Jean-Charles Soria, and Klaus F Rabe. Management of non-small-cell lung cancer: recent developments. *The Lancet*, 382(9893):709–719, 8 2013. URL http://linkinghub.elsevier.com/retrieve/pii/S0140673613615020.

Dirk Repsilber, Jan T. Kim, Hans Liljenstrm, and Thomas Martinetz. Using coarse-grained, discrete systems for data-driven inference of regulatory gene networks: Perspectives and limitations for reverse engineering. In *Proceedings of the Fifth German Workshop on Artificial Life*, pages 67–76, 2002.

Aurlien Rizk, Gregory Batt, Franois Fages, and Sylvain Soliman. A general computational method for robustness analysis with applications to synthetic gene networks. *Bioinformatics*, 25(12):i169–i178, 2009. doi: 10.1093/bioinformatics/btp200. URL http://bioinformatics.oxfordjournals.org/content/25/12/i169.abstract.

Christian P. Robert and George Casella. *Monte Carlo statistical methods*. Springer, 2 edition, 2004.

Christian P. Robert and George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 0387212396.

Christian P. Robert and George Casella. *Introducing Monte Carlo Methods with R (Use R)*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2009. ISBN 1441915753, 9781441915757.

Barbara P. Homeier Robert S. Porter, Justin L. Kaplan. *The Merck Manual Home Health Handbook*. Merck, 3 edition, 2009.

Jack A. Roth, James D. Cox, and Waun Ki Hong. *Lung Cancer*. John Wiley and Sons, August 2011.

Heist RS, Sequist LV, and Engelman JA. Genetic changes in squamous cell lung cancer: a review. *J Thorac Oncol*, 7(5):924–33, May 2012.

Jianhua Ruan, Angela Dean, and Weixiong Zhang. A general co-expression network-based approach to gene expression analysis: comparison and applications. *BMC Systems Biology*, 4(1):8, 2010. ISSN 1752-0509. doi: 10.1186/1752-0509-4-8. URL http://www.biomedcentral.com/1752-0509/4/8.

Stuart J. Russell, Peter Norvig, John F. Candy, Jitendra M. Malik, and Douglas D. Edwards. *Artificial intelligence: a modern approach*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996. ISBN 0-13-103805-2.

Igor Ruvinsky, Igor Ruvinsky, Oded Meyuhas, and Oded Meyuhas. Ribosomal protein s6 phosphorylation: from protein synthesis to cell size. *Trends in biochemical sciences*, 31 (6):342–8, 2006. URL http://www.ncbi.nlm.nih.gov/pubmed/16679021.

Mark Schena, Dari Shalon, Ronald W Davis, and Patrick O Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270 (5235):467–470, 1995.

Rebekka Schlatter, Kathrin Schmich, Ima Avalos Vizcarra, Peter Scheurich, Thomas Sauter, Christoph Borner, Michael Ederer, Irmgard Merfort, and Oliver Sawodny. On/off and beyond - a boolean model of apoptosis. *PLoS Comput Biol*, 5(12):e1000595, 12 2009. doi: 10.1371/journal.pcbi.1000595. URL http://dx.doi.org/10.1371%2Fjournal.pcbi.1000595.

Andreas Schlicker, Francisco S Domingues, Jörg Rahnenführer, and Thomas Lengauer. A new measure for functional similarity of gene products based on gene ontology. *BMC Bioinformatics*, 7:302, 2006. doi: 10.1186/1471-2105-7-302. URL http://dx.doi.org/10.1186/1471-2105-7-302.

David B. Shackelford and Reuben J. Shaw. The lkb1-ampk pathway: metabolism and growth control in tumour suppression. *Nat Rev Cancer*, 9(8):563–575, 08 2009. URL http://dx.doi.org/10.1038/nrc2676.

Claude E. Shannon and Warren Weaver. *A Mathematical Theory of Communication*. University of Illinois Press, Champaign, IL, USA, 1963. ISBN 0252725484.

Reuben J Shaw and Lewis C Cantley. Ras, pi(3)k and mtor signalling controls tumour cell growth. *Nature*, 441(7092):424–430, 2006.

Ilya Shmulevich, Edward R Dougherty, Seungchan Kim, and Wei Zhang. Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, 2002.

Jose Silva, Jennifer Nichols, Thorold W. Theunissen, Ge Guo, Anouk L. van Oosten, Ornella Barrandon, Jason Wray, Shinya Yamanaka, Ian Chambers, and Austin Smith. Nanog is the gateway to the pluripotent ground state. *Cell*, 138(4):722–737, 2009. ISSN 0092-8674.

G. K. Smyth. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 2004.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*, volume 81. MIT press, 2001.

Iwona Stelniec-Klotz, Stefan Legewie, Oleg Tchernitsa, Franziska Witzel, Bertram Klinger, Christine Sers, Hanspeter Herzel, Nils Blüthgen, and Reinhold Schäfer. Reverse engineering a hierarchical regulatory network downstream of oncogenic kras. *Molecular systems biology*, 8(601):601, January 2012. ISSN 1744-4292. doi: 10.1038/msb.2012.32. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3421447&tool=pmcentrez&rendertype=abstract.

R. Steuer, J. Kurths, C. O. Daub, J. Weise, and J. Selbig. The mutual information: Detecting and evaluating dependencies between variables. *Bioinformatics*, 18(suppl 2):S231–S240, 2002. doi: 10.1093/bioinformatics/18.suppl_2.S231. URL http://bioinformatics.oxfordjournals.org/content/18/suppl_2/S231.abstract.

Ralf Steuer, Steffen Waldherr, Victor Sourjik, and Markus Kollmann. Robust signal processing in living cells. *PLoS Comput Biol*, 7(11):e1002218, 11 2011. doi: 10.1371/journal.pcbi.1002218. URL http://dx.doi.org/10.1371%2Fjournal.pcbi.1002218.

G. Stolovitsky and A. Califano. *Reverse Engineering Biological Networks: Opportunities and Challenges in Computational Methods for Pathway Inference*. Annals of the New York Academy of Sciences. Wiley-Blackwell, 2007. ISBN 9781573316897. URL http://books.google.de/books?id=6mceAQAAIAAJ.

Gustavo Stolovitzky, Don Monroe, and Andrea Califano. Dialogue on reverse-engineering assessment and methods: the dream of high-throughput pathway inference. *Annals of the New York Academy of Sciences*, 1115(914):1–22, December 2007. ISSN 0077-8923. doi: 10.1196/annals.1407.021.

Joshua M Stuart, Eran Segal, Daphne Koller, and Stuart K Kim. A gene-coexpression network for global discovery of conserved genetic modules. *Science (New York, N.Y.)*, 302 (5643):249–55, October 2003. ISSN 1095-9203. URL http://www.ncbi.nlm.nih.gov/pubmed/12934013.

James Surowiecki. *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*. Knopf Doubleday Publishing Group, May 25, 2004.

Y. Tamada, SunYong Kim, H. Bannai, S Imoto, Kauske Tashiro, Satoru Kuhara, and Miyano S. Estimating gene networks fom gene expression data by combining with bayesian network models with promoter element detection. *Bioinformatics*, 19:ii227–ii236, 2003.

Y. Tamada, H. Banai, S. Imoto, T. Katayama, M. Kanehisa, and S. Miyano. Utilizing evolutionary information and gene expression data for estimating gene networks with bayesian network models. *J. Bioinform Comput Biol*, 3:1295–1313, 2005.

Jesper Tegnér, M. K. Stephen Yeung, Jeff Hasty, and James J. Collins. Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *Proceedings of the National Academy of Sciences*, 100(10):5944–5949, 2003. doi: 10.1073/pnas.0933416100. URL http://www.pnas.org/content/100/10/5944.abstract.

Muneesh Tewari, Patrick J. Hu, Jin Sook Ahn, Nono Ayivi-Guedehoussou, Pierre-Olivier Vidalain, Siming Li, Stuart Milstein, Chris M. Armstrong, Mike Boxem, Maurice D. Butler, Svetlana Busiguina, Jean-Franois Rual, Nieves Ibarrola, Sabrina T. Chaklos, Nicolas Bertin, Philippe Vaglio, Mark L. Edgley, Kevin V. King, Patrice S. Albert, Jean Vandenhaute, Akhilesh Pandey, Donald L. Riddle, Gary Ruvkun, and Marc Vidal. Systematic

interactome mapping and genetic perturbation analysis of a c. elegans tgf-β signaling network. *Molecular Cell*, 13(4):469 – 482, 2004. ISSN 1097-2765. doi: 10.1016/S1097-2765(04)00033-4. URL http://www.sciencedirect.com/science/article/pii/S1097276504000334.

R Thomas., R Büttner., and J Wolf. A genomics-based classification of human lung tumors. *Science Translational Medicine*, 5(209):209ra153, 2013. doi: 10.1126/scitranslmed.3006802. URL http://stm.sciencemag.org/content/5/209/209ra153.abstract.

Hiroyuki Toh and Katsuhisa Horimoto. Inference of a genetic network by a combined approach of cluster analysis and graphical gaussian modeling. *Bioinformatics*, 18(2):287–297, 2002. doi: 10.1093/bioinformatics/18.2.287. URL http://bioinformatics.oxfordjournals.org/content/18/2/287.abstract.

A. Tresch and F. Markowetz. Structure learning in nested effects models. In *Statistical Applications in Genetics and Molecular Biology*, 2008.

Jan P van Meerbeeck, Dean A Fennell, and Dirk KM De Ruysscher. Small-cell lung cancer. *Lancet*, 378(9804):1741–1755, November 2011. ISSN 0140-6736. URL http://linkinghub.elsevier.com/retrieve/pii/S0140673611601657.

Charles J Vaske, Carrie House, Truong Luu, Bryan Frank, Chen-Hsiang Yeang, Norman H Lee, and Joshua M Stuart. A factor graph nested effects model to identify networks from genetic perturbations. *PLoS Computational Biology*, 5(1):e1000274, 2009. URL http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=19180177.

Christian von Mering, Martijn Huynen, Daniel Jaeggi, Steffen Schmidt, Peer Bork, and Berend Snel. String: a database of predicted functional associations between proteins. *Nucleic Acids Res*, 31(1):258–261, Jan 2003.

Gilks W R, Richardson S, and Spiegelhalter David, editors. *Markov Chain Monte Carlo in Practice (Chapman & Hall/CRC Interdisciplinary Statistics)*. Chapman and Hall/CRC, 1 edition, December 1995. ISBN 0412055511. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0412055511.

Andreas Wagner. Estimating coarse gene network structure from large-scale gene perturbation data. *Genome Research*, 12(2):309–315, 2002. doi: 10.1101/gr.193902. URL http://genome.cshlp.org/content/12/2/309.abstract.

Yong Wang, Trupti Joshi, Xiang-Sun Zhang, Dong Xu, and Luonan Chen. Inferring gene regulatory networks from multiple microarray datasets. *Bioinformatics*, 22(19):2413–2420, 2006. doi: 10.1093/bioinformatics/btl396. URL http://bioinformatics.oxfordjournals.org/content/22/19/2413.abstract.

Z. Wang, D.D. Rao, N. Senzer, and J. Nemunaitis. Rna interference and cancer therapy. *Pharm Res*, 28(12):2983–95, 2011.

Adriano V Werhli and Dirk Husmeier. Reconstructing gene regulatory networks with bayesian networks by combining expression data with multiple sources of prior knowledge. *Stat Appl Genet Mol Biol*, 6:Article 15, 2007. doi: 10.2202/1544-6115.1282. URL http://dx.doi.org/10.2202/1544-6115.1282.

Adriano V. Werhli, Marco Grzegorczyk, and Dirk Husmeier. Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks. *Bioinformatics*, 22(20):2523–2531, 2006. doi: 10.1093/bioinformatics/btl391. URL http://bioinformatics.oxfordjournals.org/content/22/20/2523.abstract.

Jung whan Kim and Chi V. Dang. Multifaceted roles of glycolytic enzymes. *Trends in Biochemical Sciences*, 30(3):142 – 150, 2005. ISSN 0968-0004. doi: 10.1016/j.tibs.2005.01.005. URL http://www.sciencedirect.com/science/article/pii/S0968000405000289.

Christopher T. Workman, H. Craig Mak, Scott McCuine, Jean-Bosco Tagne, Maya Agarwal, Owen Ozier, Thomas J. Begley, Leona D. Samson, and Trey Ideker. A systems approach to mapping dna damage response pathways. *Science*, 312(5776):1054–1059, 2006. doi: 10.1126/science.1122088. URL http://www.sciencemag.org/content/312/5776/1054.abstract.

Ramon Xulvi-Brunet and Hongzhe Li. Co-expression networks: graph properties and topological comparisons. *Bioinformatics*, 26(2):205–214, 2010. doi: 10.1093/bioinformatics/btp632. URL http://bioinformatics.oxfordjournals.org/content/26/2/205.abstract.

Yosef Yarden and Gur Pines. The erbb network: at last, cancer therapy meets systems biology. *Nature Reviews Cancer*, 12(8):553–563, 2012.

Chen-Hsiang Yeang and Martin Vingron. A joint model of regulatory and metabolic networks. *BMC Bioinformatics*, 7:332, 2006. URL http://dblp.uni-trier.de/db/journals/bmcbi/bmcbi7.html#YeangV06.

Jing Yu, V. Anne Smith, Paul P. Wang, Alexander J. Hartemink, and Erich D. Jarvis. Advances to bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, 20(18):3594–3603, 2004. doi: 10.1093/bioinformatics/bth448. URL http://bioinformatics.oxfordjournals.org/content/20/18/3594.abstract.

Cordula Zeller, Holger Fröhlich, and Achim Tresch. A bayesian network view on nested effects models. *EURASIP journal on bioinformatics systems biology*, page 195272, 2009. URL http://www.ncbi.nlm.nih.gov/pubmed/19148294.

Marino Zerial and Yannis Kalaidzidis. The power of image analysis for systems biology. *Systemsbiologie*, (03):8–11, 2011. URL http://www.systembiologie.de/fileadmin/media/magazine/systembiologie_magazine_issue03.pdf.

Jitao David Zhang and Stefan Wiemann. Kegggraph: a graph approach to kegg pathway in r and bioconductor. *Bioinformatics*, 25(11):1470–1471, Jun 2009. doi: 10.1093/bioinformatics/btp167. URL http://dx.doi.org/10.1093/bioinformatics/btp167.

Pietro Zoppoli, Sandro Morganella, and Michele Ceccarelli. Timedelay-aracne: Reverse engineering of gene networks from time-course data by an information theoretic approach. *BMC Bioinformatics*, 11(1):154, 2010. ISSN 1471-2105. doi: 10.1186/1471-2105-11-154. URL http://www.biomedcentral.com/1471-2105/11/154.

Appendices

Appendix A
Overlap for selected KEGG pathways

List of KEGG pathways used for figure 1.2

S.No	Pathway ID	Name	Nodes	Edges
1	hsa00010	Glycolysis / Gluconeogenesis - *Homo sapiens*	65	258
2	hsa04012	ErbB signaling pathway - *Homo sapiens*	87	214
3	hsa04020	Calcium signaling pathway - *Homo sapiens*	180	511
4	hsa04115	p53 signaling pathway - *Homo sapiens*	69	86
5	hsa04120	Ubiquitin mediated proteolysis - *Homo sapiens*	151	775
6	hsa04310	Wnt signaling pathway - *Homo sapiens*	85	226
7	hsa04350	TGF-beta signaling pathway - *Homo sapiens*	167	77
8	hsa05010	Alzheimer's disease - *Homo sapiens*	327	1104
9	hsa05200	Pathways in cancer - *Homo sapiens*	62	104
10	hsa05213	Endometrial cancer - *Homo sapiens*	52	87
11	hsa05216	Thyroid cancer - *Homo sapiens*	29	49
12	hsa05217	Basal cell carcinoma - *Homo sapiens*	55	310

Plotting figure 1.2

The nodes for all these networks were extracted and a pairwise overlap analysis was done for each pair. Finally the overlap was divided by the maximum number of nodes that can be shared between two networks. The resulting matrix η with node overlap fraction was then plotted to get figure 1.2 (left)

$$\eta_{i,j} = \frac{V_i \cap V_j}{min(V_i, V_j)}$$

A similar matrix ζ for edges was computed. all the edges for corresponding graph pair was extracted and then an intersection was computed which was later normalized between 0 and 1 by divided with maximum number of edges that can be shared.

$$\zeta_{i,j} = \frac{E_i \cap E_j}{min(E_i, E_j)}$$

Appendix B
Wilcoxon rank test dynoNEM

Pairwise Wilcoxon rank Test for dynoNEM comparison						
	TPR		1-FPR		BAC	
Test results for n= 3	E-genes=50					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1.14E-01	-	0.6179701	-	0.6179701	-
simple.DNEM	2.17E-06	2.17E-06	0.5604616	0.5604616	0.5604616	0.5604616
Test results for n= 3	E-genes=100					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	3.36E-02	-	-	-	-	-
simple.DNEM	5.00E-05	1.23E-05	0.2794854	0.2794854	0.2794854	0.2794854
Test results for n= 3	E-genes=200					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	0.067685673	-	1	-	1	-
simple.DNEM	0.000159612	4.32E-05	1	1	1	1
Test results for n= 3	E-genes=500					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1.00E+00	-	-	-	-	-
simple.DNEM	7.08E-06	7.08E-06	0.4992379	0.4992379	0.4992379	0.4992379
Test results for n= 3	Parameter=0.2					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	-N	-	1	-	1	-
simple.DNEM	0.4992379	0.4992379	0.134529	0.134529	0.134529	0.134529
Test results for n= 3	Parameter=0.5					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1	-	-	-	-	-
simple.DNEM	0.134529	0.134529	1	1	1	1
Test results for n= 3	Parameter=0.8					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	-N	-	-	-	-	-
simple.DNEM	1	1	0.1003482	0.1003482	0.1003482	0.1003482
Test results for n= 3	Time points=3					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1.74E-01	-	0.37109337	-	0.37109337	-
simple.DNEM	7.56E-06	4.66E-06	0.08297354	0.06868435	0.08297354	0.06868435
Test results for n= 3	Time points=5					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	0.342781711	-	1	-	1	-
simple.DNEM	0.000235536	0.000162477	0.03559865	0.1542944	0.03559865	0.1542944
Test results for n= 3	Time points=10					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1	-	1	-	1	-
simple.DNEM	1	1	0.00089748	0.00089748	0.00089748	0.00089748
Test results for n= 5	E-genes=50					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	2.14E-03	-	0.002170287	-	0.002170287	-
simple.DNEM	4.19E-10	1.16E-10	0.05451414	0.4441013	0.05451414	0.4441013
Test results for n= 5	E-genes=100					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1.71E-02	-	2.18E-02	-	2.18E-02	-
simple.DNEM	5.39E-07	8.02E-08	3.44E-05	0.000112709	3.44E-05	0.000112709
Test results for n= 5	E-genes=200					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	3.68E-03	-	7.52E-01	-	7.52E-01	-
simple.DNEM	9.95E-08	6.42E-09	8.18E-06	8.18E-06	8.18E-06	8.18E-06
Test results for n= 5	E-genes=500					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1.19E-02	-	0.269294137	-	0.269294137	-
simple.DNEM	6.73E-06	7.60E-07	0.001647255	0.000603592	0.001647255	0.000603592
Test results for n= 5	Parameter=0.2					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	0.269294137	-	0.3155159	-	0.3155159	-
simple.DNEM	0.001647255	0.000603592	0.3558843	0.9849142	0.3558843	0.9849142

Test results for n= 5	Parameter=0.5					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	0.3155159	-	2.79E-01	-	2.79E-01	-
simple.DNEM	0.3558843	0.9849142	8.18E-06	5.21E-06	8.18E-06	5.21E-06
Test results for n= 5	Parameter=0.8					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	2.79E-01	-	7.87E-01	-	7.87E-01	-
simple.DNEM	8.18E-06	5.21E-06	2.24E-10	2.44E-10	2.24E-10	2.44E-10
Test results for n= 5	Time points=3					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	8.61E-03	-	0.002901784	-	0.002901784	-
simple.DNEM	9.38E-12	8.71E-12	0.171599173	0.002901784	0.171599173	0.002901784
Test results for n= 5	Time points=5					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1.02E-03	-	1.00E+00	-	1.00E+00	-
simple.DNEM	1.14E-06	2.15E-08	7.84E-05	7.84E-05	7.84E-05	7.84E-05
Test results for n= 5	Time points=10					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1	-	7.87E-01	-	7.87E-01	-
simple.DNEM	0.9838023	0.9838023	5.18E-11	1.19E-11	5.18E-11	1.19E-11
Test results for n= 10	E-genes=50					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	7.87E-15	-	2.51E-11	-	2.51E-11	-
simple.DNEM	5.33E-15	4.07E-17	5.52E-11	4.83E-15	5.52E-11	4.83E-15
Test results for n= 10	E-genes=100					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	2.46E-09	-	0.002194879	-	0.002194879	-
simple.DNEM	5.27E-03	1.74E-08	0.124807856	0.01204133	0.124807856	0.01204133
Test results for n= 10	E-genes=200					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1.86E-07	-	0.042724849	-	0.042724849	-
simple.DNEM	3.24E-01	0.000127633	0.004019756	0.004019756	0.004019756	0.004019756
Test results for n= 10	E-genes=500					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	2.95E-05	-	4.71E-01	-	4.71E-01	-
simple.DNEM	7.14E-03	0.627663	3.23E-06	1.14E-06	3.23E-06	1.14E-06
Test results for n= 10	Parameter=0.2					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	4.71E-01	-	0.0001097	-	0.0001097	-
simple.DNEM	3.23E-06	1.14E-06	0.144713916	0.01254526	0.144713916	0.01254526
Test results for n= 10	Parameter=0.5					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	0.0001097	-	0.2238252	-	0.2238252	-
simple.DNEM	0.144713916	0.01254526	0.01481653	0.06186102	0.01481653	0.06186102
Test results for n= 10	Parameter=0.8					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	0.2238252	-	0.699381222	-	0.699381222	-
simple.DNEM	0.01481653	0.06186102	0.000418986	0.000418986	0.000418986	0.000418986
Test results for n= 10	Time points=3					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	3.67E-04	-	0.006445709	-	0.006445709	-
simple.DNEM	1.35E-07	6.33E-08	0.009643426	0.00318709	0.009643426	0.00318709
Test results for n= 10	Time points=5					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	0.016736	-	0.8081358	-	0.8081358	-
simple.DNEM	0.1667312	0.02079715	0.2180439	0.2666198	0.2180439	0.2666198
Test results for n= 10	Time points=10					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	0.05517928	-	0.717245184	-	0.717245184	-
simple.DNEM	0.01079932	0.04770171	0.000130782	0.000250868	0.000130782	0.000250868
Test results for n= 15	E-genes=50					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	3.20E-09	-	1.68E-09	-	1.68E-09	-
simple.DNEM	1.72E-09	1.72E-09	1.68E-09	1.68E-09	1.68E-09	1.68E-09

Test results for n= 15	E-genes=100					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1.03E-07	-	3.03E-08	-	3.03E-08	-
simple.DNEM	9.69E-09	7.75E-09	1.27E-08	1.24E-08	1.27E-08	1.24E-08
Test results for n= 15	E-genes=200					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1.40E-06	-	0.000143561	-	0.000143561	-
simple.DNEM	9.32E-03	0.000212338	0.061717127	0.006325821	0.061717127	0.006325821
Test results for n= 15	E-genes=500					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	3.08E-06	-	6.17E-05	-	6.17E-05	-
simple.DNEM	6.60E-01	0.2216799	6.14E-01	0.8813869	6.14E-01	0.8813869
Test results for n= 15	Parameter=0.2					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	6.17E-05	-	1.09E-06	-	1.09E-06	-
simple.DNEM	6.14E-01	0.8813869	1.34E-03	6.27E-05	1.34E-03	6.27E-05
Test results for n= 15	Parameter=0.5					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1.09E-06	-	1.83E-05	-	1.83E-05	-
simple.DNEM	1.34E-03	6.27E-05	5.71E-01	0.08615819	5.71E-01	0.08615819
Test results for n= 15	Parameter=0.8					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1.83E-05	-	0.000287038	-	0.000287038	-
simple.DNEM	5.71E-01	0.08615819	0.177236178	0.02410612	0.177236178	0.02410612
Test results for n= 15	Time points=3					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	2.43E-07	-	2.50E-07	-	2.50E-07	-
simple.DNEM	6.19E-09	4.98E-09	1.57E-07	4.41E-08	1.57E-07	4.41E-08
Test results for n= 15	Time points=5					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1.28E-05	-	3.16E-05	-	3.16E-05	-
simple.DNEM	4.22E-02	0.000367713	1.29E-01	0.0769776	1.29E-01	0.0769776
Test results for n= 15	Time points=10					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	1.56E-05	-	0.000874993	-	0.000874993	-
simple.DNEM	7.43E-02	0.02649255	0.818528517	0.8185285	0.818528517	0.8185285
Test results for n= 20	Parameter=0.2					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	0.000874993	-	2.82E-14	-	2.82E-14	-
simple.DNEM	0.818528517	0.8185285	8.49E-03	2.44E-10	8.49E-03	2.44E-10
Test results for n= 20	Parameter=0.5					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	2.82E-14	-	3.84E-16	-	3.84E-16	-
simple.DNEM	8.49E-03	2.44E-10	7.60E-02	1.46E-12	7.60E-02	1.46E-12
Test results for n= 20	Parameter=0.8					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	3.84E-16	-	6.66E-17	-	6.66E-17	-
simple.DNEM	7.60E-02	1.46E-12	7.77E-01	1.06E-14	7.77E-01	1.06E-14
Test results for n= 20	Time points=3					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	4.62E-16	-	1.36E-17	-	1.36E-17	-
simple.DNEM	4.19E-17	2.54E-17	8.68E-18	8.68E-18	8.68E-18	8.68E-18
Test results for n= 20	Time points=5					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	2.29E-15	-	2.47E-16	-	2.47E-16	-
simple.DNEM	9.61E-04	1.20E-12	5.85E-02	2.36E-12	5.85E-02	2.36E-12
Test results for n= 20	Time points=10					
	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC	dynoNEM.HC	dynoNEM.MCMC
dynoNEM.MCMC	2.03E-15	-	2.28E-15	-	2.28E-15	-
simple.DNEM	4.64E-02	0.6502722	8.49E-08	0.002772096	8.49E-08	0.002772096

Appendix C
Evaluating MSC network against literature

GHC			MCMC		
Knowledge View					
#	Inferred Edge	Explained by	#	Inferred Edge	Explained by
1	Nanog->Oct4	Nanog->Oct4	1	Nanog->Oct4	Nanog->Oct4
2	Oct4->Tcl1	Oct4->Tcl1	2	Nanog->Tcl1	Nanog->Oct4->Tcl1
3	Sox2->Oct4	Sox2->Oct4	3	Oct4->Tcl1	Oct4->Tcl1
4	Sox2->Tcl1	Sox2->Oct4->Tcl1	4	Sox2->Oct4	Sox2->Oct4
5	Esrrb->Oct4	Esrrb->Oct4	5	Sox2->Tcl1	Sox2->Oct4->Tcl1
6	Esrrb->Tcl1	Esrrb->Oct4->Tcl1	6	Esrrb->Oct4	Esrrb->Oct4
7	Tbx3->Oct4	Tbx3->Oct4	7	Tbx3->Oct4	Tbx3->Oct4
8	Tbx3->Esrrb	Tbx3->Oct4->Esrrb	8	Tbx3->Esrrb	Tbx3->Oct4->Esrrb
			9	Tbx3->Tcl1	Tbx3->Oct4->Tcl1
			10	Tcl1->Oct4	---
Model View					
#	Literature path	Explained by model	#	Literature path	Explained by model
1	Nanog->Oct4	Nanog->Oct4	1	Nanog->Oct4	Nanog->Oct4
2	Nanog->Sox2	---	2	Nanog->Sox2	---
3	Nanog->Esrrb	---	3	Nanog->Esrrb	---
4	Nanog->Tbx3	---	4	Nanog->Tbx3	---
5	Nanog->Tcl1	Nanog->Oct4->Tcl1	5	Nanog->Tcl1	Nanog->Tcl1
6	Oct4->Nanog	---	6	Oct4->Nanog	---
7	Oct4->Sox2	---	7	Oct4->Sox2	---
8	Oct4->Esrrb	---	8	Oct4->Esrrb	---
9	Oct4->Tbx3	---	9	Oct4->Tbx3	---
10	Oct4->Tcl1	Oct4->Tcl1	10	Oct4->Tcl1	Oct4->Tcl1
11	Sox2->Nanog	---	11	Sox2->Nanog	---
12	Sox2->Oct4	Sox2->Oct4	12	Sox2->Oct4	Sox2->Oct4
13	Sox2->Esrrb	---	13	Sox2->Esrrb	---
14	Sox2->Tbx3	---	14	Sox2->Tbx3	---
15	Sox2->Tcl1	Sox2->Tcl1	15	Sox2->Tcl1	Sox2->Tcl1
16	Esrrb->Nanog	---	16	Esrrb->Nanog	---
17	Esrrb->Oct4	Esrrb->Oct4	17	Esrrb->Oct4	Esrrb->Oct4
18	Esrrb->Sox2	---	18	Esrrb->Sox2	---
19	Esrrb->Tbx3	---	19	Esrrb->Tbx3	---
20	Esrrb->Tcl1	Esrrb->Tcl1	20	Esrrb->Tcl1	Esrrb->Oct4->Tcl1
21	Tbx3->Nanog	---	21	Tbx3->Nanog	---
22	Tbx3->Oct4	Tbx3->Oct4	22	Tbx3->Oct4	Tbx3->Oct4
23	Tbx3->Sox2	---	23	Tbx3->Sox2	---
24	Tbx3->Esrrb	Tbx3->Esrrb	24	Tbx3->Esrrb	Tbx3->Esrrb
25	Tbx3->Tcl1	Tbx3->Oct4->Tcl1	25	Tbx3->Tcl1	Tbx3->Tcl1

Figure C.1.: Data for Figure 4.13, showing the pathway in inferred network and its explainability by literature and vie-versa.

Appendix D
Wilcoxon rank test pathway reconstruction

Table D.1.: Pairwise Wilcoxon test for model performance comparison (false discovery rates) for KEGG sub-graphs with $m = 10$ nodes

	Methods	IP	IP.RNK	LFM	MP	MP.RNK	NOM	NOM.RNK
10	IP.RNK	-	-	-	-	-	-	-
	LFM	0.0041	0.0041	-	-	-	-	-
	MP	0.0203	0.0203	0.0352	-	-	-	-
	MP.RNK	0.0423	0.0423	0.0203	0.0304	-	-	-
	NOM	0.0041	0.0041	0.2974	0.0041	0.0041	-	-
	NOM.RNK	0.0041	0.0041	1.0000	0.0099	0.0041	0.0041	-
	STRING	0.0070	0.0070	0.0041	0.0945	0.5111	0.0041	0.0041
	Methods	IP	IP.RNK	LFM	MP	MP.RNK	NOM	NOM.RNK
20	IP.RNK	0.0036	-	-	-	-	-	-
	LFM	0.0036	0.2712	-	-	-	-	-
	MP 0.0144	0.0064	0.0260	-	-	-	-	
	MP.RNK	0.0036	0.6250	0.4200	0.0348	-	-	-
	NOM	0.0036	0.0036	0.5773	0.0036	0.0091	-	-
	NOM.RNK	0.0036	0.0036	0.4648	0.0036	0.0064	0.1022	-
	STRING	0.0036	0.0036	0.0036	0.0260	0.0036	0.0036	0.0036
	Methods	IP	IP.RNK	LFM	MP	MP.RNK	NOM	NOM.RNK
40	IP.RNK	0.0068	-	-	-	-	-	-
	LFM	0.0039	0.0980	-	-	-	-	-
	MP	0.0594	0.2166	0.0091	-	-	-	-
	MP.RNK	0.0039	0.4038	0.0594	0.0201	-	-	-
	NOM	0.0039	0.0495	0.6481	0.0039	0.0039	-	-
	NOM.RNK	0.0039	0.0039	0.9219	0.0039	0.0039	0.0091	-
	STRING	0.0039	0.0068	0.0039	0.0383	0.0039	0.0039	0.0039

Appendix E
Feature computed for movieNEM image data

Table E.1.: Features computed for the movieNEM image data using CellProfiler

Feature Class	Feature Name
AreaShape	Zernike
Texture	InfoMeas1
AreaShape	Compactness
Location	MaxIntensity
RadialDistribution	RadialCV
RadialDistribution	FracAtD
AreaShape	Eccentricity
RadialDistribution	RadialCV
RadialDistribution	FracAtD
AreaShape	MaxFeretDiameter
AreaShape	EulerNumber
Intensity	StdIntensityEdge
Intensity	MinIntensityEdge
AreaShape	MinorAxisLength
Texture	Entropy
RadialDistribution	RadialCV
RadialDistribution	MeanFrac
RadialDistribution	RadialCV
AreaShape	Perimeter
AreaShape	FormFactor
RadialDistribution	MeanFrac
Location	CenterMassIntensity
Intensity	StdIntensity
Texture	Contrast
RadialDistribution	MeanFrac
Texture	SumEntropy
Intensity	MedianIntensity
Texture	DifferenceVariance
RadialDistribution	FracAtD

Table E.2.: Features computed for the movieNEM image data using CellProfiler (Continued from last page)

Feature Class	Feature Name
Texture	DifferenceEntropy
RadialDistribution	FracAtD
Intensity	MinIntensity
Intensity	IntegratedIntensityEdge
Intensity	MaxIntensityEdge
Intensity	UpperQuartileIntensity
Intensity	MassDisplacement
Intensity	MeanIntensity
Intensity	MaxIntensity
Texture	Correlation
Texture	AngularSecondMoment
Texture	SumVariance
Texture	Variance
AreaShape	Area
AreaShape	MinFeretDiameter
Intensity	LowerQuartileIntensity
Texture	SumAverage
Texture	InfoMeas2
AreaShape	MedianRadius
AreaShape	Extent
Texture	InverseDifferenceMoment
RadialDistribution	MeanFrac
AreaShape	MajorAxisLength
AreaShape	MeanRadius
Texture	Gabor
AreaShape	MaximumRadius
Intensity	IntegratedIntensity
Intensity	MeanIntensityEdge

Appendix F
MetaCore^{TM}network used for MovieNEM validation

The literature based network used to compare and validate the NEM inferred network for cell cycle network was extracted from the database tool MetaCore ^{TM}. To retrieve the network the list of input gene was same as the S-genes for the used in our MovieNEM network (chapter 5) data i.e the perturbed genes (see table F.1)

Table F.1.: List of input genes to retrieve network from MetaCore ^{TM}

PRC1	EZH2	CDK1	CDC25B	CIZ1
PDIA3	MTA2	CTR9	RUNX2	FOXM1
CDK2	CDC25A	SKP2	CDKN1A	ING1
CDK4	SP1	MDM2	YY1	STAT3
E2F5	CREB1			

Since it was not always possible to detect a direct relation between two genes of interest, we used a shortest path algorithm implemented in the MetaCore ^{TM}tool. This algorithm takes the nodes of interest and tries to identify all the paths available between every pair of input genes. We set a maximum path length of 2. This means if there was a path between a pair of input genes, with a path legth equal to 2 it was included in our network. Thus we could figure out direct and indirect relations among the genes. Doing so also retrieved some genes which were not a part of our list but were connecting the input genes. The small molecules in the network thus retrieved were ommitted if they did not break the network. The entire network computed has been shown in figure F.1. In the shown network, the genes that were the part of the input list are shown in red and grey nodes are the connecting genes.

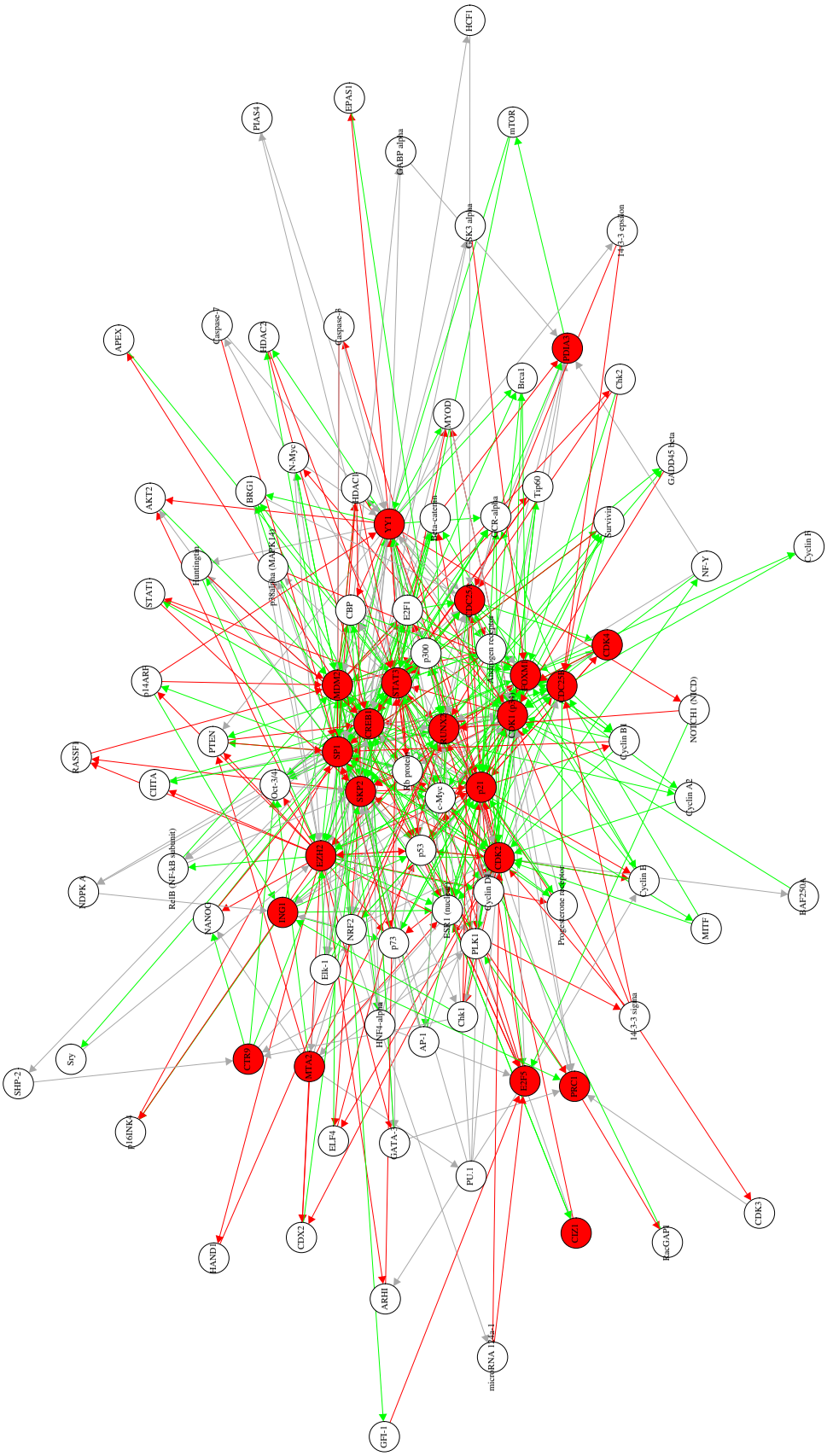

Figure F.1.: Literature network for MovieNEM genes. S-genes have been colored in red. Red edges indicate inhibition whereas, green indicates activation

Appendix G
Literature validation: Cell Cycle network

Table G.1.: The edges in retrieved network by MovieNEM for cell cycle and their explanation by MetaCore ™

Inferred Edge	Explained By
PRC1 → CDC25A	PRC1 → RacGAP1 → STAT3 → CDC25A
PRC1 → SKP2	PRC1 → RacGAP1 → STAT3 → SKP2
PRC1 → E2F5	PRC1 → PLK1 → MDM2 → HNF4-alpha → E2F5
EZH2 → CDK1 (p34)	EZH2 → 14-3-3 sigma → CDK1 (p34)
EZH2 → CDC25B	EZH2 → 14-3-3 sigma → CDC25B
EZH2 → CDC25A	EZH2 → NANOG → CDC25A
EZH2 → SKP2	EZH2 → RelB (NF-kB subunit) → SKP2
EZH2 → CDK4	EZH2 → p21 → CDK4
EZH2 → SP1	EZH2 → PTEN → SP1
EZH2 → YY1	EZH2 → p14ARF → YY1
EZH2 → E2F5	EZH2 → microRNA 124a-1 → E2F5
CDK1(p34) → CIZ1	CDK1(p34) → PLK1 → CIZ1
CDK1(p34) → CDC25A	CDK1(p34) → CDC25A
CDC25B → CDK1 (p34)	CDC25B → CDK1 (p34)
CDC25B → CDC25A	CDC25B → CDK1 (p34) → CDC25A
CDC25B → MDM2	CDC25B → Androgen receptor → YY1 → MDM2
CDC25B → E2F5	CDC25B → CDK1 (p34) → Rb protein → E2F5
CIZ1 → CDK1(p34)	CIZ1 → p21 → CDK1(p34)
CIZ1 → CTR9	CIZ1 → p21 → PLK1 → CTR9
CIZ1 → CDC25A	CIZ1 → p21 → CDC25A
CIZ1 → E2F5	CIZ1 → p21 → CDK2 → E2F5
PDIA3 → PRC1	PDIA3 → p21 → CDK3 → PRC1
PDIA3 → CDK1(p34)	PDIA3 → CDC25A → CDK1(p34)
PDIA3 → CIZ1	PDIA3 → p21 → PLK1 → CIZ1
PDIA3 → CDC25A	PDIA3 → CDC25A
MTA2 → FOXM1	MTA2 → p53 → FOXM1
MTA2 → CDC25A	MTA2 → NANOG → CDC25A
CTR9 → CDK1 (p34)	CTR9 → p21 → CDK1 (p34)
CTR9 → CDC25A	CTR9 → Oct-3/4 → CDC25A

Table G.2.: The edges in retrieved network and their explanation by MetaCore ™continued from last page

Inferred Edge	Explained By
RUNX2 → PRC1	RUNX2 → Rb protein → CDK1 (p34) → PRC1
RUNX2 → EZH2	RUNX2 → Rb protein → EZH2
RUNX2 → CDK1 (p34)	RUNX2 → Rb protein → CDK1 (p34)
RUNX2 → CDC25B	RUNX2 → Survivin → CDC25B
RUNX2 → PDIA3	RUNX2 → Rb protein → CDK1 (p34) → PDIA3
RUNX2 → MTA2	RUNX2 → Caspase-7 → SP1 → MTA2
RUNX2 → CDC25A	RUNX2 → Rb protein → CDC25A
RUNX2 → SKP2	RUNX2 → Rb protein → SKP2
RUNX2 → CDK4	RUNX2 → CDK4
RUNX2 → SP1	RUNX2 → Caspase-7 → SP1
RUNX2 → MDM2	RUNX2 → Rb protein → MDM2
RUNX2 → STAT3	RUNX2 → Rb protein → STAT3
RUNX2 → E2F5	RUNX2 → Rb protein → E2F5
FOXM1 → SP1	FOXM1 → Cyclin B1 → SP1
CDK2 → CDK1 (p34)	CDK2 → BAF250A → CDK1 (p34)
CDK2 → CDC25B	CDK2 → CDC25B
CDK2 → CIZ1	CDK2 → ESR1 (nuclear) → CIZ1
CDK2 → PDIA3	CDK2 → NF-Y → PDIA3
CDK2 → MTA2	CDK2 → MTA2
CDK2 → CTR9	CDK2 → Chk1 → CTR9
CDK2 → FOXM1	CDK2 → FOXM1
CDK2 → CDC25A	CDK2 → CDC25A
CDK2 → ING1	CDK2 → c-Myc → ING1
CDK2 → CDK4	CDK2 → SKP2 → CDK4
CDK2 → MDM2	CDK2 → MDM2
CDK2 → STAT3	CDK2 → SP1 → STAT3
CDC25A → CDK1 (p34)	CDC25A → CDK1 (p34)
SKP2 → CDK1 (p34)	SKP2 → p21 → CDK1 (p34)
SKP2 → CIZ1	SKP2 → p21 → PLK1 → CIZ1
SKP2 → CDC25A	SKP2 → p21 → CDC25A
SKP2 → SP1	SKP2 → c-Myc → SP1
SKP2 → MDM2	SKP2 → RASSF1 → MDM2
p21 → CDK1 (p34)	p21 → CDK1 (p34)
p21 → CIZ1	p21 → PLK1 → CIZ1
p21 → PDIA3	p21 → CDK1 (p34) → PDIA3
p21 → CTR9	p21 → PLK1 → CTR9

Table G.3.: The edges in retrieved network and their explanation by MetaCore ™continued from last page

Inferred Edge	Explained By
p21 → FOXM1	p21 → Cyclin E → FOXM1
p21 → CDC25A	p21 → CDC25A
p21 → SKP2	p21 → STAT3 → SKP2
p21 → CDK4	p21 → CDK4
p21 → SP1	p21 → Caspase-3 → SP1
p21 → MDM2	p21 → Caspase-3 → MDM2
p21 → YY1	p21 → Caspase-3 → YY1
p21 → STAT3	p21 → STAT3
p21 → E2F5	p21 → CDK2 → E2F5
p21 → CREB1	p21 → c-Myc → CREB1
ING1 → PRC1	ING1 → p53 → PRC1
ING1 → EZH2	ING1 → p53 → EZH2
ING1 → CDK1 (p34)	ING1 → p73 → CDK1 (p34)
ING1 → CDC25B	ING1 → p53 → CDC25B
ING1 → CIZ1	ING1 → ESR1 (nuclear) → CIZ1
ING1 → MTA2	ING1 → p16INK4 → SP1 → MTA2
ING1 → CTR9	ING1 → p21 → PLK1 → CTR9
ING1 → FOXM1	ING1 → ESR1 (nuclear) → FOXM1
ING1 → CDC25A	ING1 → p21 → CDC25A
ING1 → CDK4	ING1 → p21 → CDK4
ING1 → SP1	ING1 → p16INK4 → SP1
ING1 → MDM2	ING1 → p73 → MDM2
ING1 → E2F5	ING1 → p53 → E2F5
CDK4 → PRC1	CDK4 → p21 → CDK3 → PRC1
CDK4 → EZH2	CDK4 → RUNX2 → Rb protein → EZH2
CDK4 → CDK1 (p34)	CDK4 → p21 → CDK1 (p34)
CDK4 → CDC25B	CDK4 → FOXM1 → CDC25B
CDK4 → CDC25A	CDK4 → p21 → CDC25A
CDK4 → SKP2	CDK4 → FOXM1 → SKP2
CDK4 → SP1	CDK4 → p21 → STAT3 → SP1
CDK4 → MDM2	CDK4 → RUNX2 → Rb protein → MDM2
CDK4 → YY1	CDK4 → RUNX2 → Rb protein → YY1
CDK4 → E2F5	CDK4 → RUNX2 → Rb protein → E2F5
SP1 → CDC25A	SP1 → CDC25A

Table G.4.: The edges in retrieved network and their explanation by MetaCore ™continued from last page

Inferred Edge	Explained By
SP1 → YY1	SP1 → YY1
SP1 → E2F5	SP1 → GFI-1 → E2F5
MDM2 → EZH2	MDM2 → p53 → EZH2
MDM2 → CDK1 (p34)	MDM2 → p53 → CDK1 (p34)
MDM2 → CDC25B	MDM2 → Chk2 → CDC25B
MDM2 → CIZ1	MDM2 → ESR1 (nuclear) → CIZ1
MDM2 → CDC25A	MDM2 → Chk2 → CDC25A
MDM2 → SKP2	MDM2 → p53 → SKP2
MDM2 → SP1	MDM2 → SP1
MDM2 → YY1	MDM2 → p53 → YY1
MDM2 → E2F5	MDM2 → p53 → E2F5
YY1 → CDK1 (p34)	YY1 → CDK1 (p34)
YY1 → CDC25A	YY1 → HCF1 → CDC25A
YY1 → SKP2	YY1 → SKP2
E2F5 → CDK1 (p34)	E2F5 → CDK1 (p34)
E2F5 → CDC25B	E2F5 → c-Myc → CDC25B
E2F5 → CDC25A	E2F5 → c-Myc → CDC25A
E2F5 → SKP2	E2F5 → CDK1 (p34) → SKP2
CREB1 → PRC1	CREB1 → Elk-1 → PRC1
CREB1 → CDK1 (p34)	CREB1 → CDK1 (p34)
CREB1 → CIZ1	CREB1 → SP1 → ESR1 (nuclear) → CIZ1
CREB1 → CDC25A	CREB1 → SP1 → CDC25A
CREB1 → SKP2	CREB1 → STAT1 → SKP2

Appendix H
Merged KEGG graphs used for simulation

Table H.1.: Merged KEGG pathways used for NEM simulation)

ID	Pathway Name
04014	Ras signaling pathway
04015	Rap1 signaling pathway
04010	MAPK signaling pathway
04012	ErbB signaling pathway
04310	Wnt signaling pathway
04330	Notch signaling pathway
04340	Hedgehog signaling pathway
04350	TGF-beta signaling pathway
04390	Hippo signaling pathway
04370	VEGF signaling pathway
04630	Jak-STAT signaling pathway
04064	NF-kappa B signaling pathway
04668	TNF signaling pathway
04066	HIF-1 signaling pathway
04068	FoxO signaling pathway
04020	Calcium signaling pathway
04070	Phosphatidylinositol signaling system
04151	PI3K-Akt signaling pathway
04150	mTOR signaling pathway
05200	Pathways in cancer
05202	Transcriptional misregulation in cancer
05206	MicroRNAs in cancer
05205	Proteoglycans in cancer
05204	Chemical carcinogenesis
05203	Viral carcinogenesis
05210	Colorectal cancer
05212	Pancreatic cancer
05214	Glioma
05216	Thyroid cancer

Table H.2.: Merged KEGG pathways used for NEM simulation-contd)

ID	Pathway Name
05310	Asthma
05322	Systemic lupus erythematosus
05323	Rheumatoid arthritis
05320	Autoimmune thyroid disease
05321	Inflammatory bowel disease (IBD)
05330	Allograft rejection
05332	Graft-versus-host disease
05340	Primary immunodeficiency
05010	Alzheimer's disease
05012	Parkinson's disease
05014	Amyotrophic lateral sclerosis (ALS)
05016	Huntington's disease
05020	Prion diseases
04940	Type I diabetes mellitus
04930	Type II diabetes mellitus
04932	Non-alcoholic fatty liver disease (NAFLD)
04950	Maturity onset diabetes of the young
05030	Cocaine addiction
05031	Amphetamine addiction
05032	Morphine addiction
05033	Nicotine addiction
05034	Alcoholism
05410	Hypertrophic cardiomyopathy (HCM)
05412	Arrhythmogenic right ventricular cardiomyopathy (ARVC)
05414	Dilated cardiomyopathy
05416	Viral myocarditis
05221	Acute myeloid leukemia
05220	Chronic myeloid leukemia
05217	Basal cell carcinoma
05218	Melanoma
05211	Renal cell carcinoma
05219	Bladder cancer
05215	Prostate cancer
05213	Endometrial cancer
05222	Small cell lung cancer
05223	Non-small cell lung cancer

Table H.3.: Merged KEGG pathways used for NEM simulation-contd)

ID	Pathway Name
04080	Neuroactive ligand-receptor interaction
04060	Cytokine-cytokine receptor interaction
04512	ECM-receptor interaction
04514	Cell adhesion molecules (CAMs)
05166	HTLV-I infection
05162	Measles
05164	Influenza A
05161	Hepatitis B
05160	Hepatitis C
05168	Herpes simplex infection
05169	Epstein-Barr virus infection
05146	Amoebiasis
05144	Malaria
05145	Toxoplasmosis
05140	Leishmaniasis
05142	Chagas disease (American trypanosomiasis)
05143	African trypanosomiasis

Appendix I
Literature network used for comparing NSCLC network

The literature based network used to compare and validate the NEM inferred network for NSCLC data was extracted from the database tool MetaCore ™. To retrieve the network the list of input gene was same as the S-genes for the used NSCLC data i.e the perturbed genes (see table I.1)

Table I.1.: List of input genes to retrieve network from MetaCore ™

STK11	PIK3C3	TRUB2	LEPR	TSC2
TSC1	GSK3B	GSK3A	RAF1	ITGB4
PRKAB1	BCL10	EGFR	ESPL1	SRC
WDR3	PRKAA1	RPS6KA1	MTOR	RPSKB1

Since it was not always possible to detect a direct relation between two genes of interest, we used a shortest path algorithm implemented in the MetaCore ™tool. This algorithm takes the nodes of interest and tries to identify all the paths available between every pair of input genes. We set a maximum path length of 2. This means if there was a path between a pair of input genes, with a path legth equal to 2 it was included in our network. Thus we could figure out direct and indirect relations among the genes. Doing so also retrieved some genes which were not a part of our list but were connecting the input genes. The small molecules in the network thus retrieved were ommitted if they did not break the network. The entire network computed has been shown in figure I.1. In the shown network, the genes that were the part of the input list are shown in red and grey nodes are the connecting genes.

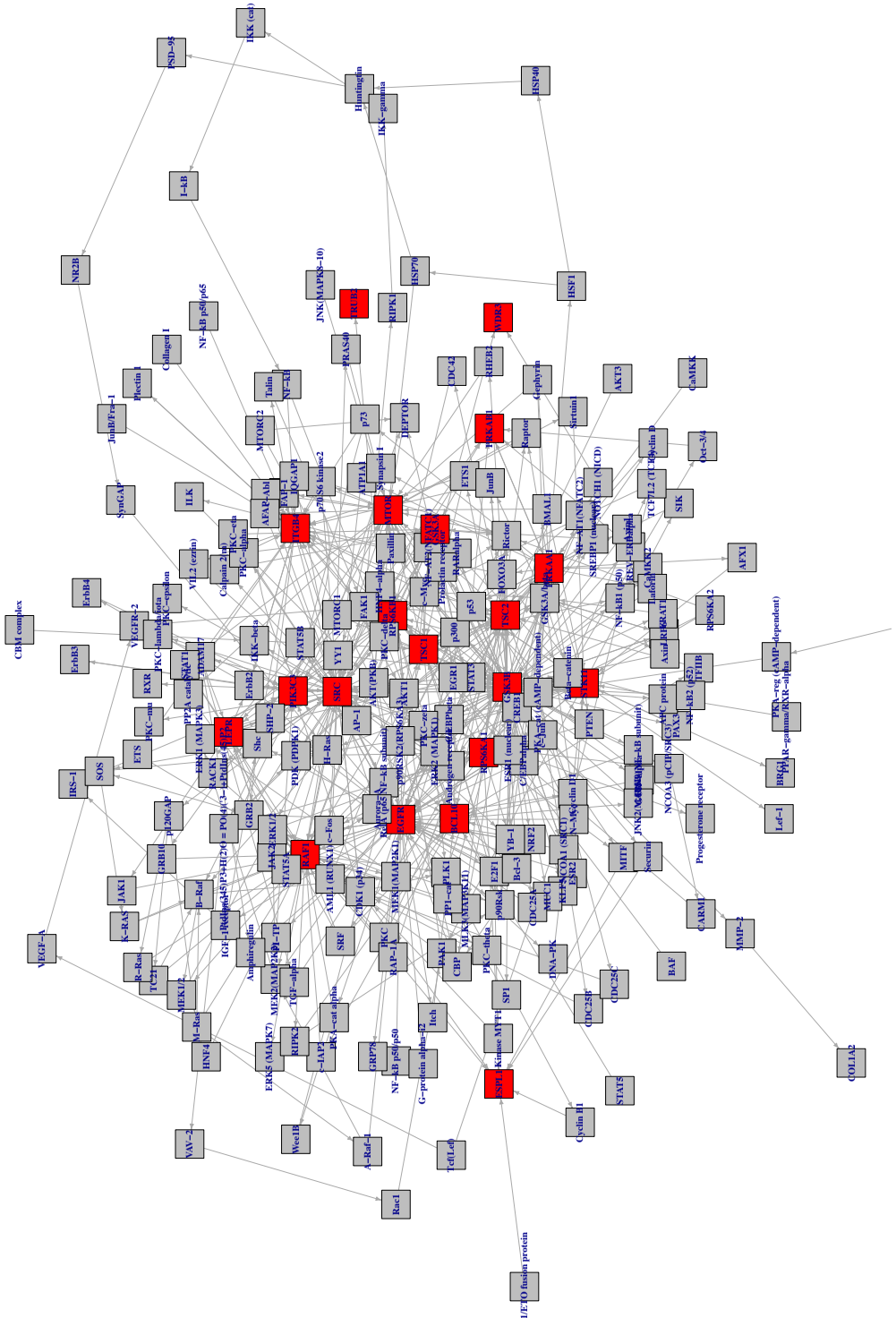

Figure I.1.: Literature network from MetaCore ™used for comparing and validating NSCLC networks. The network containg additional nodes as the network was derived by a shortest pathlength agorithm for path length 2.

Appendix J
NSCLC network explained by literature

Table J.1.: The edges in retrieved network and their explanation by MetaCore ™

Inferred Edge	Explained By
BCL10 →EGFR	BCL10 →JNK2(MAPK9) →GSK3B →MUC1 →EGFR
EGFR →GSK3A	EGFR →AKT1 →GSK3A
EGFR →PIK3C3	EGFR →PIK3C3
ESPL1 →ITGB4	none
GSK3A →TRUB2	GSK3A →EGR1 →SRC →HNF4-alpha →TRUB2
GSK3B →GSK3A	GSK3B →ETS1 →GSK3A
ITGB4 →PIK3C3	ITGB4 →VIL2 (ezrin) →SRC →PIK3C3
LEPR →GSK3A	LEPR →SRC →AKT(PKB) →GSK3A
STK11 →GSK3A	STK11 →GSK3B →SREBP1 (nuclear) →GSK3A
MTOR →PIK3C3	MTOR →c-Myc →PIK3C3
RPS6KB1 →PRKAB1	RPS6KB1 →GSK3B →REV-ERBalpha →PRKAB1
RPS6KA1 →RPS6KB1	RPS6KA1 →GSK3B →RPS6KB1
PIK3C3 →BCL10	PIK3C3 → PtdIns(45)P2 →AKT(PKB) →BCL10
PIK3C3 →TSC1	PIK3C3 →MTOR →c-Myc →TSC1
PRKAA1 →WDR3	PRKAA1 →p53 →GSK3B →NOTCH1 (NICD) →WDR3
PRKAB1 →RPS6KB1	PRKAB1 →PRKAA1 →PKC-zeta →RPS6KB1
RAF1 →MTOR	RAF1 →Aurora-A →SRC →c-Abl →MTOR
RAF1 →SRC	RAF1 →Aurora-A →SRC
SRC →RAF1	SRC →RAF1
TRUB2 →GSK3B	none
TRUB2 →LEPR	none
TRUB2 →RPS6KA1	none
TRUB2 →TSC2	none
TSC1 →GSK3A	TSC1 →MTOR →AKT(PKB) →GSK3A
TSC2 →GSK3A	TSC2 →CDC42 →RPS6KB1 →GSK3A
WDR3 →RAF1	none

Appendix K
Selected GO terms for S-genes

To look at the functional role of S-gene under analysis we did a GO term annotation for thse S-genes. We focussed on the Biological Process (BP) category. The GO terms of our interest are terms associated with the following terms or the related terms:

1. Apoptosis
2. Cell proliferation
3. Cell cycle
4. Mitosis
5. Cell division

Most of the S-genes under consideration were found to be annotated with such terms. These associted terms have been listed together with the corresponding genes on the nex page. Please note that the list presented here contains only the annotations of our interest. The complete list is not been mentioned here.

BCL10			
8915	biological_process	GO:0042981	regulation of apoptotic process
8915	biological_process	GO:0008219	cell death
8915	biological_process	GO:0006921	cellular component disassembly involved in execution phase of apoptosis
8915	biological_process	GO:0002906	negative regulation of mature B cell apoptotic process
8915	biological_process	GO:0043280	positive regulation of cysteine-type endopeptidase activity involved in apoptotic process
8915	biological_process	GO:2001238	positive regulation of extrinsic apoptotic signaling pathway
8915	biological_process	GO:0001783	B cell apoptotic process
8915	biological_process	GO:0070231	T cell apoptotic process
EGFR			
1956	biological_process	GO:0008283	cell proliferation
1956	biological_process	GO:0050679	positive regulation of epithelial cell proliferation
1956	biological_process	GO:0043066	negative regulation of apoptotic process
1956	biological_process	GO:0008284	positive regulation of cell proliferation
1956	biological_process	GO:0031659	positive regulation of cyclin-dependent protein serine/threonine kinase activity involved in G1/S transition of mitotic cell cycle
1956	biological_process	GO:0042127	regulation of cell proliferation
1956	biological_process	GO:0045930	negative regulation of mitotic cell cycle
1956	biological_process	GO:0048661	positive regulation of smooth muscle cell proliferation
ESPL1			
9700	biological_process	GO:0000212	meiotic spindle organization
9700	biological_process	GO:0000070	mitotic sister chromatid segregation
9700	biological_process	GO:0045143	homologous chromosome segregation
9700	biological_process	GO:0006915	apoptotic process
9700	biological_process	GO:0000910	cytokinesis
9700	biological_process	GO:0000278	mitotic cell cycle
9700	biological_process	GO:0045842	positive regulation of mitotic metaphase/anaphase transition
9700	biological_process	GO:0040001	establishment of mitotic spindle localization
9700	biological_process	GO:0045875	negative regulation of sister chromatid cohesion
9700	biological_process	GO:0007059	chromosome segregation
9700	biological_process	GO:0007127	meiosis I
GSK3B			
2932	biological_process	GO:0000320	re-entry into mitotic cell cycle
2932	biological_process	GO:0043066	negative regulation of apoptotic process
2932	biological_process	GO:2000738	positive regulation of stem cell differentiation

Figure K.1.: Selected GO terms for S-genes.

PIK3C3			
5289	biological_process	GO:0000910	cytokinesis
PRKAA1			
5562	biological_process	GO:0043066	negative regulation of apoptotic process
5562	biological_process	GO:0008284	positive regulation of cell proliferation
5562	biological_process	GO:0007050	cell cycle arrest
PRKAB1			
5564	biological_process	GO:0007050	cell cycle arrest
RAF1			
5894	biological_process	GO:0008283	cell proliferation
5894	biological_process	GO:0006915	apoptotic process
5894	biological_process	GO:0008285	negative regulation of cell proliferation
5894	biological_process	GO:0042981	regulation of apoptotic process
5894	biological_process	GO:0043066	negative regulation of apoptotic process
5894	biological_process	GO:0045595	regulation of cell differentiation
5894	biological_process	GO:0043154	negative regulation of cysteine-type endopeptidase activity involved in apoptotic process
5894	biological_process	GO:0071550	death-inducing signaling complex assembly
RPS6KA1			
6195	biological_process	GO:0043066	negative regulation of apoptotic process
6195	biological_process	GO:0007049	cell cycle
6195	biological_process	GO:0043154	negative regulation of cysteine-type endopeptidase activity involved in apoptotic process
6195	biological_process	GO:0045597	positive regulation of cell differentiation
RPS6KB1			
6198	biological_process	GO:0048661	positive regulation of smooth muscle cell proliferation
6198	biological_process	GO:0006915	apoptotic process
6198	biological_process	GO:0043066	negative regulation of apoptotic process
6198	biological_process	GO:0045931	positive regulation of mitotic cell cycle
6198	biological_process	GO:0000082	G1/S transition of mitotic cell cycle
6198	biological_process	GO:2001237	negative regulation of extrinsic apoptotic signaling pathway
SRC			
6714	biological_process	GO:2001243	negative regulation of intrinsic apoptotic signaling pathway
6714	biological_process	GO:0007049	cell cycle
6714	biological_process	GO:2001237	negative regulation of extrinsic apoptotic signaling pathway
STK11			
6794	biological_process	GO:0008285	negative regulation of cell proliferation
6794	biological_process	GO:0007050	cell cycle arrest
6794	biological_process	GO:0072332	intrinsic apoptotic signaling pathway by p53 class mediator
TSC1			
7248	biological_process	GO:0008285	negative regulation of cell proliferation
7248	biological_process	GO:0007050	cell cycle arrest
TSC2			
7249	biological_process	GO:0008285	negative regulation of cell proliferation
7249	biological_process	GO:0007050	cell cycle arrest
7249	biological_process	GO:0051726	regulation of cell cycle
7249	biological_process	GO:0050680	negative regulation of epithelial cell proliferation

Figure K.2.: Selected GO terms for S-genes (Continued from last page)

Appendix L
Log fold changes in RPPA data

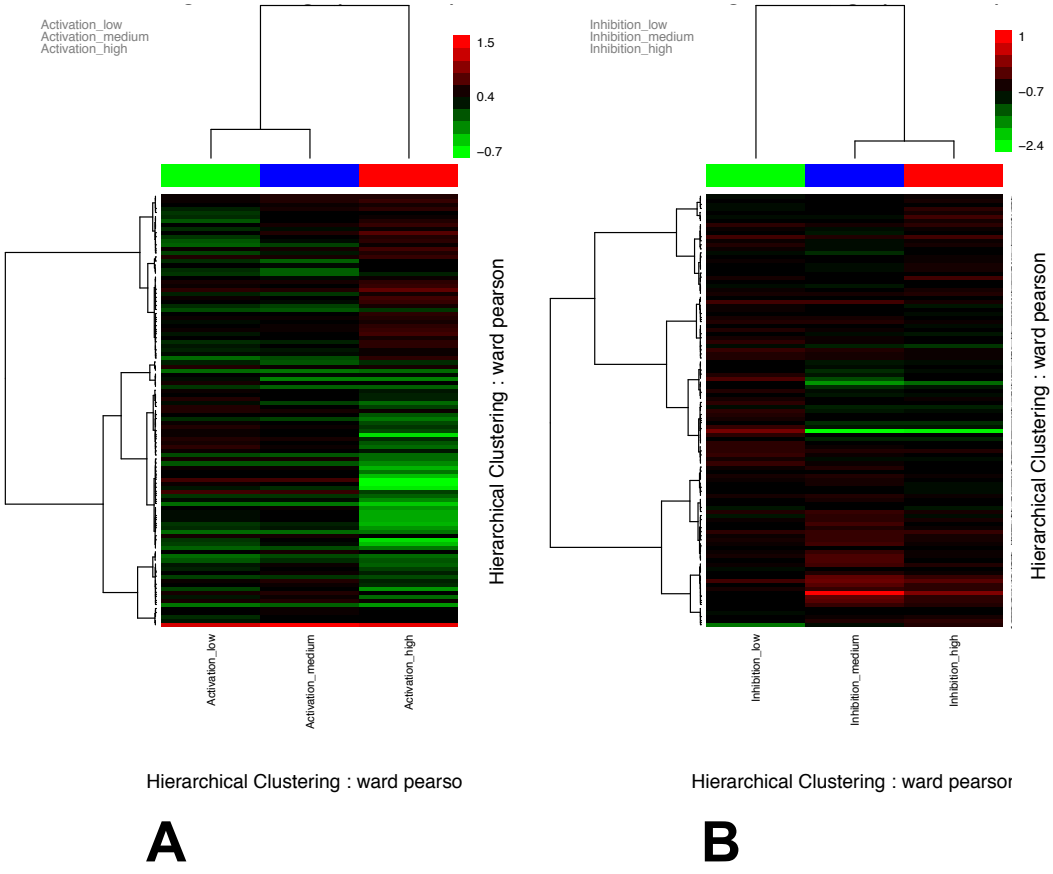

Figure L.1.: Heatmap: log fold changes for all proteins measured in RPPA for drug effect (A) Activation (B) Inhibition.

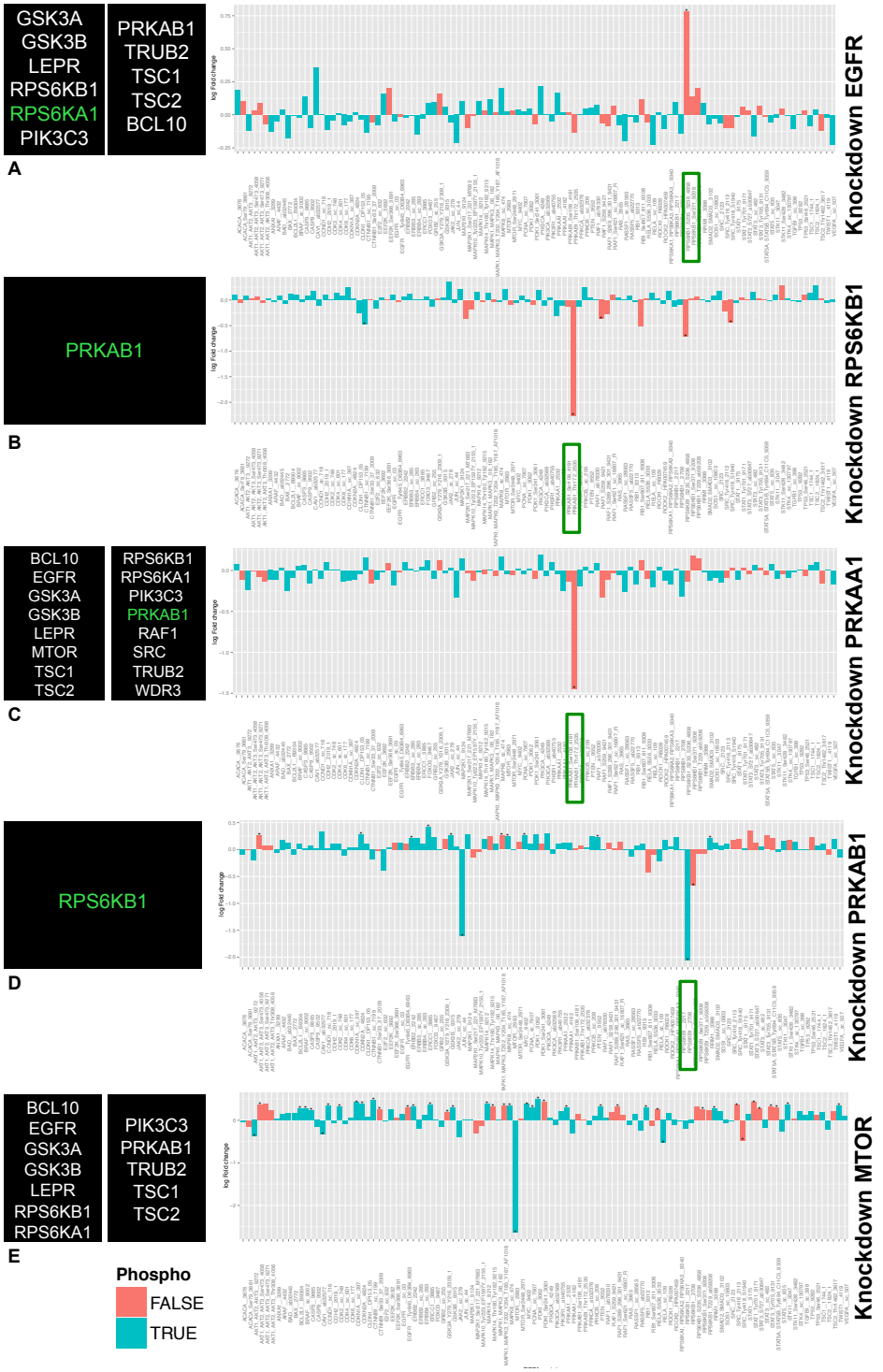

Figure L.2.: Plot showing effect of knock down for five genes (EGFR, RPS6KB1, PRKAA1 PRKAB1 and mTOR) in terms of log fold changes at protein level obtained via RPPA. The downstream S-genes computed via a $breadthfirstsearch$ are provided in the box on the left hand side. The measured protein located at immediate downstream neighborhood have been highlighted in green.

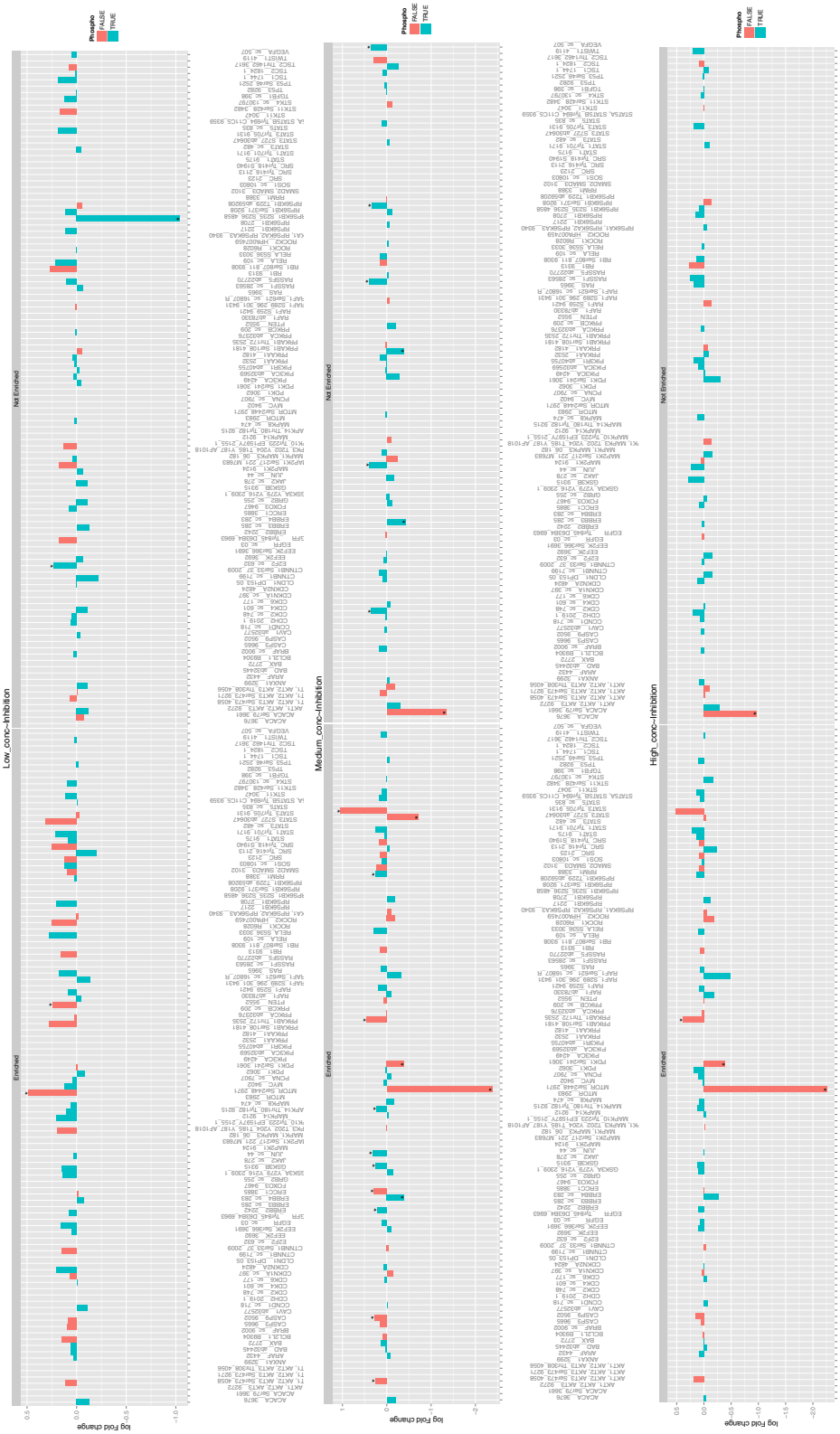

Figure L.3.: log fold changes drug treatment (inhibitory) for genes enriches with interesting go categories. Similar plot for activation control can be found in appendix L.4.

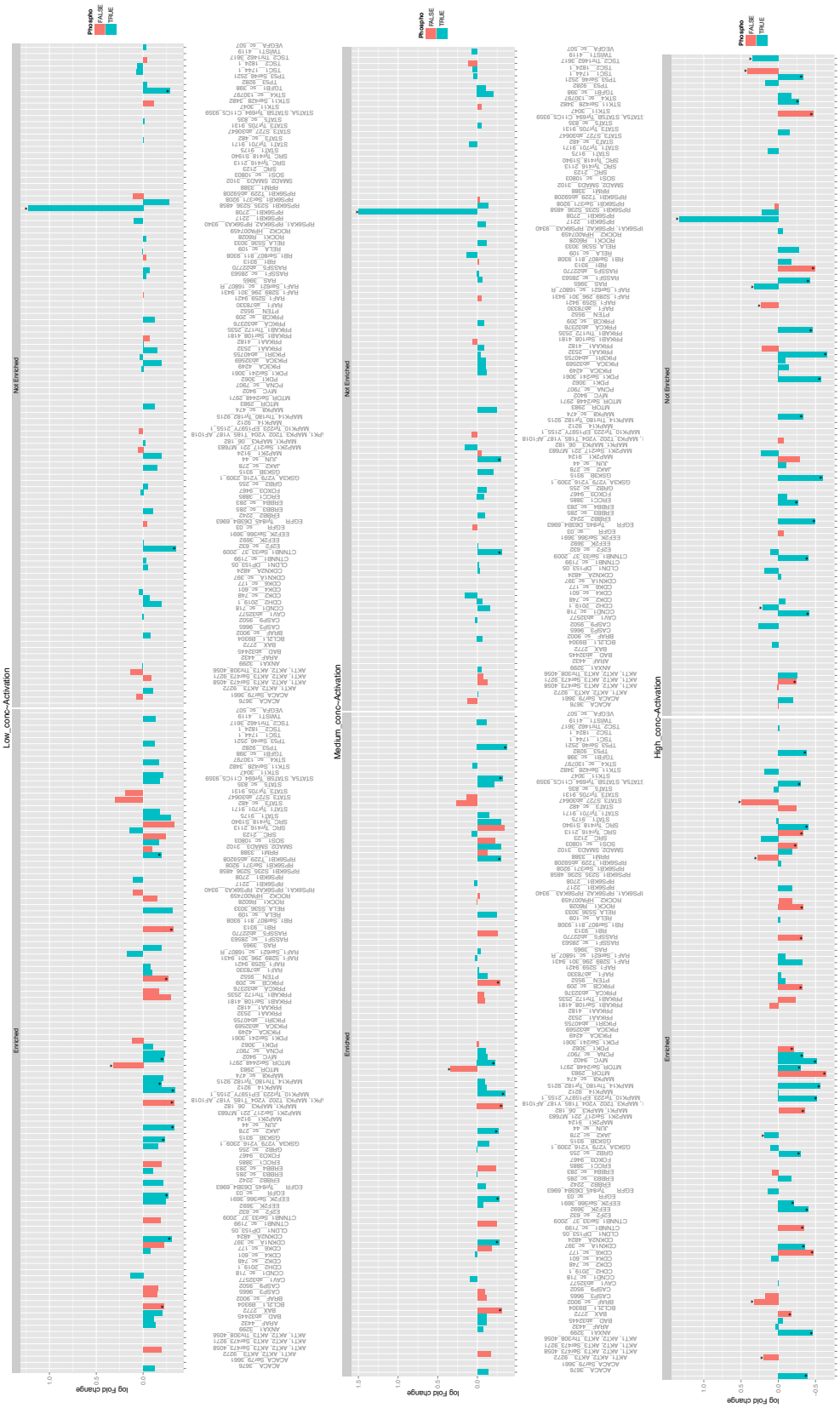

Figure L.4.: log fold changes after activation by drug targetting AMPKt

Appendix M
List of publications

Peer reviewed journals

1. PriorNet: R package to compute consensus probabilistic prior for networks from multiple knowledge sources, *Bioinformatics* (To be submitted)*

2. Learning optimal EGFR/AMPK signaling network from perturbation data integrated with knowledge to identify therapeutic targets, *Nature Molecular Systems Biology* (Submitted)*

3. Boosting probabilistic graphical model inference by incorporating prior knowledge from multiple sources. *PLoS ONE*, 8(6):e67410, 06 2013. doi: 10.1371/journal.pone.0067410. *

4. Learning gene network structure from time laps cell imaging in RNAi knockdowns. *Bioinformatics*, 2013. doi:10.1093/bioinformatics/btt179. *

5. Fast and efficient dynamic nested effects models. *Bioinformatics*, 27(2):238244, 2011.

Conference

1. Learning Gene Network Structure from Time Laps Cell Imaging in RNAi Knock-Downs, **Highlight Paper**, *German Conference on Bioinformatics (GCB)*, Göttingen (Germany) 2013 *

2. Constructing informative prior from multiple knowledge sources to improve network inference, *Rocky Mountain Bioinformatics Conference, Aspen, CO (USA)* 2012. *#

3. Boosting Statistical Network Inference by Incorporating Prior Knowledge from Multiple Sources, *European Conference on Computational Biology*, and in satellite Conferences: *Machine Learning in Systems Biology and ESCS* 2012 (Basel, SWITZERLAND) 2012. *#

4. H. Fröhlich, P. Praveen , A. Tresch; Fast and Efficient Dynamic Nested Effects Models, *Intelligent Systems for Molecular Biology (ISMB)* 2011, **Highlight Paper**, Viena (AUSTRIA), July 2011.

Posters (selected)

1. P. Praveen , A. Tresch, H. Fröhlich; Fast and Efficient Dynamic Nested Effects Models, ISMB/ECCB European Conference on Computational Biology 2011, Viena (AUSTRIA), July 2011

2. P. Praveen , A. Tresch, H. Fröhlich; Efficient learning of signaling networks from perturbation time series via dynamic nested effects models, European Conference on Computational Biology 2010, Gent (BELGIUM), September 2010 *

* First author
# Award winning presentations

List of Figures

Vitae

Paurush Praveen

Rovereto, Italy

Academics

- **2010-2013: Bonn-Aachen International Center for Information Technology (B-IT), University of Bonn (Bonn, Germany)** Doctoral Studies in Comutaional Life Science
- **2007-2009: Bonn-Aachen International Center for Information Technology (B-IT), University of Bonn (Bonn, Germany)** Master of Science in Life Science Informatics
- **2007: Indian Institute of Technology (IIT Kharagpur, India)** STC in Bioinformatics for Genomics and Proteomoics
- **2002-2006: Acharya Institute of Technology (Bangalore (India))** Bachelor of Engineering in Biotechnology

Experience

- **2013-Till date: The Microsoft Research-COSBI** Research Scientist
- **2008-2010: Fraunhofer Institute for Scientific Computing and Algorithms (SCAI)** Research Assistant
- **2006: Defense Research and Development Organization (Defence Bioengineering and Electromedical Laboratory (DRDO DEBEL), Bangalore, India)** Trainee