Exploring the Boundaries of Gene Regulatory Network Inference

Andreas Tjärnberg

Stockholm
University

Exploring the Boundaries of Gene Regulatory Network Inference

Andreas Tjärnberg

ISBN 978-91-7649-299-4

Print: Holmbergs, Malmö 2015
Distributor: Department of Biochemistry and Biophysics

Abstract

To understand how the components of a complex system like a living cell interact and regulate each other, we need to collect data about how the components respond to system perturbations. Such data can then be used to solve the inverse problem of inferring a network that describes how the pieces influence each other. The work in this thesis concerns modelling of the regulatory system of a cell, often represented as a network. Common tools and concepts in systems biology are used.

The first investigation focuses on network sparsity and algorithmic biases introduced by penalised network inference procedures. Many contemporary network inference methods rely on a sparsity parameter, such as the L_1 penalty term used in the LASSO. However, a poor choice of the sparsity parameter can give highly incorrect network estimates. In order to avoid such poor choices, we devised a method to optimise the sparsity parameter, which maximises the accuracy of the inferred network. We showed that it is effective on *in silico* data sets with a reasonable level of informativeness and demonstrated that accurate prediction of network sparsity is key to elucidate the correct network parameters.

The second investigation focuses on how knowledge from association networks can be used in regulatory network inference procedures. The quality of expression data is typically inadequate for reliable gene regulatory network (GRN) inference. Therefore, we constructed an algorithm to incorporate prior knowledge and demonstrated that it increases the accuracy of network inference when the quality of the data is low.

The third investigation aimed to understand the influence of system and data properties on network inference accuracy. L_1 regularisation methods commonly produce poor network estimates when the data is ill-conditioned, even when the signal to noise ratio is so high that all links in the network can be proven to exist at the common significance level of 0.01. In this study we elucidated general principles about the conditions for which we expect degraded accuracy. Moreover, it allowed us to estimate the expected accuracy from observable properties of simulated data, that can be used to predict the performance of inference algorithms on biological data.

Finally, we built a software package GeneSPIDER for solving problems encountered during previous investigations. The software package supports highly controllable network and data generation, as well as data analysis and exploration in the context of network inference.

To my grandmother Mildred

List of Papers

The following papers, referred to in the text by their Roman numerals, are included in this thesis.

PAPER I: **Optimal sparsity criteria for network inference.**
Tjärnberg A., Nordling T., Studham M., and Sonnhammer EL. *Journal of Computational Biology*, **20(5)**, 398-4089 (2013).
DOI: 10.1089/cmb.2012.0268

PAPER II: **Functional association networks as priors for gene regulatory network inference.**
Studham M., Tjärnberg A., Nordling T., Nelander S., and Sonnhammer EL. *Bioinformatics*, **30(12)**, i130–i138 (2014).
DOI: 10.1093/bioinformatics/btu285

PAPER III: **Avoiding pitfalls in l1-regularised inference of gene networks.**
Tjärnberg A., Nordling T., Studham M., Nelander S., and Sonnhammer EL. *Mol. BioSyst.*, **1(11)**, 287-296 (2015).
DOI: 10.1039/C4MB00419A

PAPER IV: **GeneSPIDER - Generation and Simulation Package for Informative Data ExploRation.**
Tjärnberg A., Nordling T., Morgan D., Studham M., and Sonnhammer EL. *manuscript.*, (2015).

Reprints were made with permission from the publishers.

Contents

Abbreviations

AIC	Akaike information criterion.
AUPR	area under precision recall curve.
AUROC	area under receiver operator curve.
BIC	Bayesian information criterion.
CLS	constrained least squares.
CV	cross validation.
FBL	feedback loop.
FFL	feed forward loop.
FN	false negative.
FP	false positive.
GRN	gene regulatory network.
LASSO	least absolute shrinkage and selection operator.
LS	least squares.
MM	Michaelis-Menten.
ODE	ordinary differential equation.
RNI	robust network inference.
RSS	residual sum of squares.
SNR	signal to noise ratio.
TF	transcription factor.
TLS	total least squares.
TN	true negative.
TP	true positive.

1. Introduction

Biological systems are diverse and complex and not easily described by simple rules. Whether it is the patterns of whole populations and societies, the interactions of individual organisms or the cellular, internal, and external machinery, understanding biological systems is motivated by the need to predict, modify or enhance the behaviour or functions of these systems. Gaining a mechanistic understanding of the system can help prevent outbreak of systemic failure (disease), as well as present the optimal course of action to restore base level functioning. Thus, if cancer for example, is viewed as an unwanted state, or a breakdown of cellular function we would like to intervene and return the system to its natural state. This could be done either by forcing the system to return to its original state or by selectively terminating the faulty cells without damaging healthy cells. Understanding how the parts of the system interact helps to selectively target and steer the system as a whole to a desired state.

Components of these systems are capable of changing over time, responding to internal and external inputs in non-random ways, as well as transferring information among themselves. A key observation of these systems is that their behaviour cannot be trivially understood or predicted by knowing any single component's properties. However, emergent behaviours can be viewed as a result of placing several components in a specific context. A component can be a anything from a gene or cell, to a population of cells or entire organisms. A human cell, for example, cannot easily be grown in isolation without modifying its internal machinery to some state dissimilar from that which it exhibited *in vivo*. That is, the composition of the system is not complete without incorporating the interactions the components exert among themselves and the resultant behaviour that arises (Barabasi and Oltvai, 2004). It should be noted that a wide range of systems can be investigated using similar, contextually relevant concepts, and that knowledge can be derived indirectly from other areas of research, evidenced by comparisons between the structuring of different system interactions that display both common and disparate traits (Milo et al., 2002).

This thesis and the work herein aims to understand the intracellular machinery, specifically what we will call the gene regulatory network (GRN).

The intracellular system cannot be accurately described isolated from its environment, and if it were, we could not assume that the behaviour would reflect that in its natural environment. This observation makes studying these systems non-trivial. Specific changes to the system are not easily induced, isolated or even measured.

Classically, if we want to study some phenomena of nature, we isolate it to the best of our ability and selectively change variables to build a picture of how the phenomena can best be described. For the reasons mentioned above, compounded by the cheer number of components of the system, with around 20 000 protein coding genes giving around 400 000 000 possible interactors within a single cell, it is nearly impossible to isolate a biological system to such a degree yet on a large scale as to be confident that there are no disruptive unknown variables in play. All studies considering more than a few components need to account for this effect and incorporate noise into their conclusions.

One aim of systems biology is to understand the structure and behaviour of biological systems on a specific hierarchical level, where the cell is but one example. A thorough understanding of the boundaries and performance of the tools used and the properties of the experiments carried out is of prime importance. The focus of the work done in this thesis is the study of properties of algorithms and data for constructing reliable models for representing biological systems. To contribute to the possibility of inferring, from data, GRNs with high confidence, that accurately reflects the underlying biology, in order to allow conclusions and knowledge to be derived from these models.

2. Background

2.1 Biological systems

Biological systems cover a wide range of different phenomena. In this section I will go through the specific biological system referred to in this thesis, gene regulation in the cell. This will, in part, motivate the need of the mathematical and computational modelling used in this research area. The vast complexity of the cell is such that to account for all components and environmental factors that interact and regulate response in the cell is an intractable problem. A core phenomenon of the cellular function is expression, *i.e.* the regulation of when and how much of any given biomolecules are expressed.

2.1.1 Gene regulation and gene regulatory networks

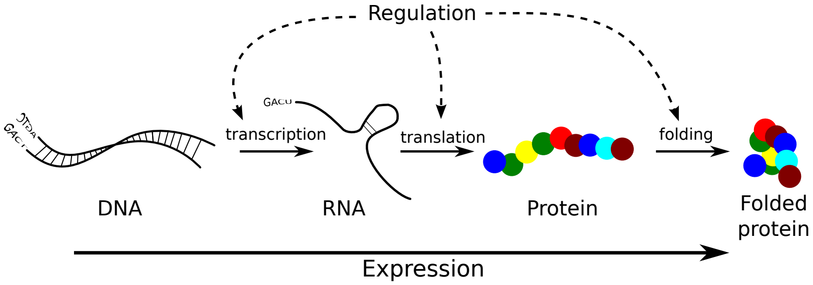

Figure 2.1: The central dogma of molecular biology (Brown, 2002). The flow of expression is shown left to right. Figure inspired by Gardner and Faith (2005)

Regulation in biological systems means the process of how an entity (biomolecule) controls the behaviour of another entity (biomolecule). In the cell this can be the process of a protein binding to DNA to regulate how much of a specific gene becomes transcribed, where the protein is referred to as a transcription factor (TF). When the TF binds to the gene-specific binding site, the interaction activates expression of the gene. If the TF lowers or turns off the expression of a gene, the interaction is suppressing the gene. The TF *regulates* the gene and this then counts as a regulation. Figure 2.1 shows the flow of expression, where gene expression is a multistep process (Brown, 2002). First the gene is transcribed, converting a double stranded DNA nucleotide base sequence into a single stranded RNA molecule one or more times. Second, the RNA molecule's nucleotide sequence is translated to a sequence of amino acids, *i.e.* a protein. The third step folds this amino acid sequence, imparting further functionality to a now fully realised protein. An additional step of the central dogma of molecular biology is *DNA replication*, where the DNA replicates itself during cell division. This perpetuation step is not directly considered here when considering gene expression.

Each of these levels of expression can be regulated by environmental factors in the cell. The concentration of a specific TF, for example, determines how saturated a TF binding site is and in essence how much the regulated gene is affected. Each component of the system has an associated number of parameters that refer to specific rate constants of biochemical reactions taking place or parameters of the model used (see: sections 2.2 and 2.3.1).

External signalling also plays a central role in regulating internal molecular concentrations and responses, as demonstrated by for example, the regulatory interactions of the bacterial flagellum. The bacterial flagellum is an appendage protruding from the bacteria, with the function to move the bacteria in response to external environmental factors. In short, the bacterium senses a concentration gradient through receptors on the cell membrane, if it is moving. If the gradient indicates that the bacteria is moving towards something nutritious the behaviour of the flagellum will change and the bacteria will propel itself towards the higher concentration of nutrients. If no gradient is sensed the behaviour changes and the bacterium tumbles randomly until a new signal appears. The bacteria also responds to damaging chemicals by reversing the response so the direction of motion is away from the higher concentration (Berg, 2000).

The complex function displayed by the bacteria could not be achieved without predictable regulation. The regulatory machinery and behaviour of the flagellum can be modelled accurately, and displays several different emergent systems properties. These include *e.g.* robustness, the function of the regulatory machinery to maintain its function for a wide range of parameters of the system, and exact adaptation, meaning that the bacteria can reset the internal state to be able to respond appropriately to new changes even though the external environment is changed *i.e.* the bacteria counteracts being overwhelmed by chemical stimuli (Alon, 2007).

Reactions taking place in the cell works on several different time scales. For example in *Escherichia coli* the time a TF takes to find and bind to a specific target location takes roughly 1-6 minutes (Elf et al., 2007). This is done through diffusion through the cell. For larger cells or for faster reactions the cell has to rely on different mechanisms for regulation (Alon, 2007). To get an overview of the interactions or regulatory machinery we can display the interactions, of TF bindings or protein to protein interactions that we can infer or observe as links in a graph. This is then a network representation of the interactions in the cell. Including metabolites into the network expands the description beyond the interactions of genes to encompass other cell signalling phenomena. We can also model a cellular regulatory network by quantifying the interactions with a direction of influence, *i.e.* if the interaction is increasing or decreasing the activity or expression of the target. This would then constitute the cellular regulatory network.

Figure 2.2 shows a hierarchical separation of different regulatory networks in the cell. The interconnected nature of these intercellular components does not lend itself to differentiation and thus many of these relationships cannot be well defined in a real cell. Here, however, they are separated by concepts, and in some regards, measuring techniques. In the figure we have the metabolic layer, depicting the path of different metabolites or transformations of metabolites, modelled often by mass action kinetics (Jamshidi and Palsson, 2010). We also have the protein layer that details the protein to protein interaction network as well as protein complexes. The formation of a protein complex would constitute a case where the proteins might not have any regulatory effect on each other, or they might not be influencing the rate or change of any of the proteins involved. However, they are still considered an interaction here. It can be that the complex regulates something else and that all involved proteins need to be present for a regulatory interaction to occur, much like an `AND` operator in a Boolean operation. The third layer is the gene layer where DNA sequence is transcribed to RNA. The RNA themselves can have regulatory effects alone, they can become translated into proteins or both.

The dashed lines on the bottom layer are interactions shown as direct influences between genes themselves. In reality, this can not be the case. Instead this is a representation of the interactions when the gene products and their cumulative effects through different layers are condensed to a description of interactions between genes.

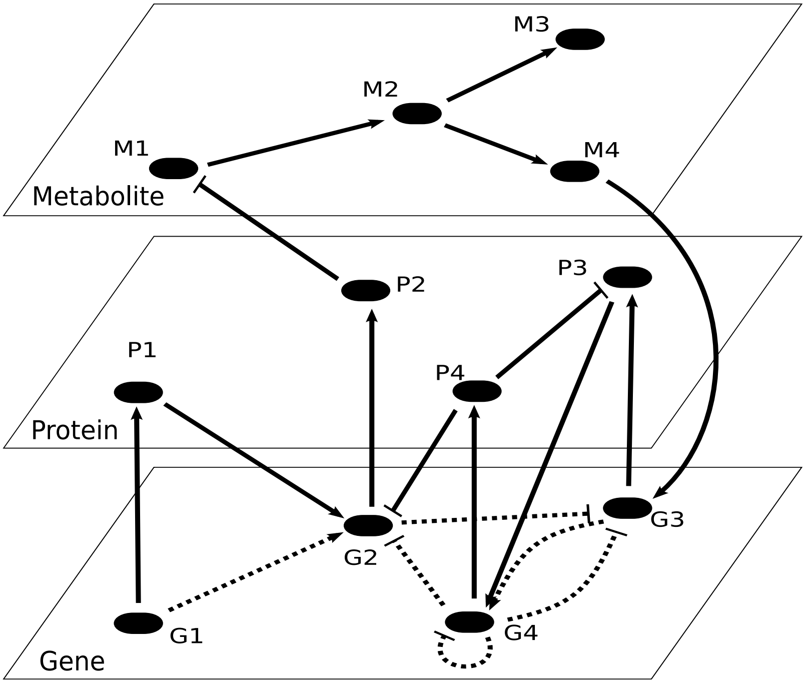

Figure 2.2: Different hierarchical levels of displaying the cellular regulatory network. The arrows indicates direction of regulation; if the head of the link is an arrow it means the interaction is activating and if the head of the link is T shaped it means the interaction is suppressing. Figure inspired by Crampin (2006).

In the following part of this thesis this abstract layer is what is referred to as the gene regulatory networks (GRNs) and *gene interactions* should be interpreted as the cumulative effects of the influences of gene products on other gene products, if not stated otherwise.

Discussing the GRN in these terms is partially made for practical reasons. All nodes of the "true" GRN as depicted in the figure might not be observable under specific experimental setups. For example, the experimental setup for measuring mRNA, protein and metabolites is very different and is not easily combined on a large scale, and in some cases the dynamics of one layer might not be representative of those in another layer (Gygi et al., 1999). The time scales of reactions for different layers or sub-networks may also vary substantially. Some interactions may not be observed if measuring the system over several days or under just a few seconds (Elf et al., 2007). One can not assume that any given collection of cells being observed are synchronised in expressing different properties or processes. One cell might be in the process of differentiating, displaying an expression pattern related specifically to that state, while other cells might not be in that state. A measurement on such a setup reflects an average over the cells in the sample and might not reflect any specific interaction pattern present in any of the individual cells.

It is also common that the different layers of the networks are separated into different databases. For simpler organisms the TF network is constructed from curated data and contains a large number of interactions. *RegulonDB* (Salgado et al., 2013) has a large set of TF binding interactions collected in a regulatory network of *E. coli*, while Yeastract (Teixeira et al., 2013) hosts a similar database corresponding to *Saccharomyces cerevisiae*. These networks aim at mapping direct binding interactions between gene and gene products, specifically TFs and binding sites. It has also been shown that mRNA expression data can be used to construct these networks (Faith et al., 2007), and that it can be used to validate or extract knowledge.

Network medicine

One of the main areas of practical application for network biology is in medicine. Roughly 2000 of human genes are disease associated (Amberger et al., 2009). With the high amount of interactors and interactions it is implied that the effects of the disease associations are not isolated to those genes (Barabasi et al., 2011). The effect of *comorbidity* is an indication that a specific disease is not isolated in its effects. Comorbidity is the ability of a disease to enhance other diseases if some specific disease is already present. By building a network of interactions and influences of cellular components a bigger picture can emerge of disease effects on the regulatory system. By overlaying implicated disease genes on the network one can draw conclusions of other disease associated genes. The more complete this picture the better the conclusions of such a study (Barabasi et al., 2011).

One of the main goals of drug discovery is to find compounds with specific properties that can target and affect pathways with high accuracy with minimal side effects (Schreiber, 2000). Generating reliable models that predict the effect of a specific perturbation from a drug compound will aid in creating more specific and effective drug treatments.

Great interest and funds for drug development are geared towards curing cancer. Cancer treatments are usually highly invasive as cancer affects the regulatory operations of the cell itself, altering the signalling pathways and behaviour (Weinberg, 1996). The effects of cancer are multi-factorial, many times different for each type of cancer, and related to the regulatory system of the cell. An accurate model of healthy cells would serve as a basis for finding alterations in the regulatory system on a very detailed level.

Systems biology approaches for elucidating the context specific regulatory networks of the cell will aid in creating a medical approach that is, predictive, personalised and preventive (Flores et al., 2013).

2.2 System theory

In this section I will give a general description of a system. I will also introduce ordinary differential equations (ODEs) and dynamical systems as a description of how a system is changing over time, and finally in this section I will give a brief description of properties associated with systems in a GRN framework.

2.2.1 System description

The representation of a system is as important as learning about the system itself. Whether mathematical, chemical, graphical or other, an accurate description can help fuel insight into what is being observed. This is especially important because assumptions of the representation have the potential to confer inaccurate or misleading information.

A mathematical description of a system is *e.g.*

$$\Psi(\boldsymbol{\phi}_j, \boldsymbol{\xi}) = 0 \tag{2.1}$$

for a multivariate problem, where $\boldsymbol{\theta}$ is the model parameters of the model and Ψ is the function that connects the independent variables $\boldsymbol{\phi}_j$, to the dependent variables, $\boldsymbol{\xi}$ (Aster et al., 2005). For example, the commonly used linear mapping is of the form

$$\boldsymbol{\Phi\theta} = \boldsymbol{\xi} \tag{2.2}$$

Here independent variables ϕ_{ij} are mapped by the parameters $\boldsymbol{\theta}$ to the dependant variables ξ_i. For $n = 3$ variables and m data points recorded, this becomes

$$\begin{bmatrix} \phi_{11} & \phi_{21} & \phi_{31} \\ \phi_{12} & \phi_{22} & \phi_{32} \\ \vdots & \vdots & \vdots \\ \phi_{1m} & \phi_{2m} & \phi_{3m} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_m \end{bmatrix} \tag{2.3}$$

In the inverse problem (see section 2.3.3) one needs to find a set of parameters $\boldsymbol{\theta}$ that fits the data ($\boldsymbol{\xi}$,$\boldsymbol{\Phi}$).

2.2.2 Dynamical systems

A dynamical system describes a set of variables behaviour over time.

A popular method of describing evolving systems is the ODE model, which relates the state of the system to its rate of instantaneous change

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x},\boldsymbol{p},\boldsymbol{v},t) \tag{2.4}$$

where $\dot{\boldsymbol{x}}$ is the rate of change of the states $\boldsymbol{x}$ representing some measurable quantity, in the context of GRNs this is typically mRNA expression or protein abundance. $\boldsymbol{p}$ is any input to the system, henceforth called perturbation, and $\boldsymbol{v}$ is stochastic effects, or noise affecting the evolution of the system. f may be any function and t the time. Now

$$\boldsymbol{y} = g(\boldsymbol{x}(t),\boldsymbol{\epsilon}) \tag{2.5}$$

describes the output variables $\boldsymbol{y}$ as a function, g, of the states $\boldsymbol{x}$ and the noise term $\boldsymbol{\epsilon}$, the output variables may be the the same as the input variables.

2.2.3 System properties

Network motifs

Some specific network motifs have been shown to be over represented in biological systems, as demonstrated by investigating the transcriptional network of *E. coli* and *S. cerevisiae* (Milo et al., 2002), while others are underrepresented, compared to what would be expected of random networks. Of special note is the feed forward loop (FFL) motif, which is highly over represented. One hypothetical reason for evolution of the under- or over- representation of specific regulatory motifs is that they serve specific functions, such as delayed responses, pulse response, synchronisation clocks, step responses and switches (Alon, 2007). The Feedback loops (FBLs) motif is often considered in system theoretic approaches, capable of causing highly correlated responses, so called interampatte systems (Nordling and Jacobsen, 2009) (see: section 2.2.3 (Interampatte systems)). Motifs may also explain phenotype, when functioning as biological switches (Wolkenhauer et al., 2005). Feedback has been shown to help describe the behaviour of bacterial chemotaxis (Yi et al., 2000). A few examples of modelling the FBL are presented in section 2.2.3 (Linear vs. nonlinear models).

Steady states

Steady states are defined when the rate of change $\dot{\boldsymbol{x}} = 0 \equiv f(\boldsymbol{x}_0, \boldsymbol{p}, T)$ in (2.4). The nature of the steady state can be elucidated by analysing the system in the vicinity of $f(\boldsymbol{x}_0, \boldsymbol{p}, T) = 0$, with T being a time when the system is in steady state and $\boldsymbol{x}_0$ is the state of the system at steady state. The solution to the system of equations for $\boldsymbol{x}_0$ in multivariate analysis, is the steady state. For the system $f(\boldsymbol{x}_0) = 0$ we can calculate the jacobian, J, the partial derivatives of f over the states $\boldsymbol{x}$. The nature of the steady states can then be derived from the eigenvalues of J for linear time invariant systems such as the ones studied here. If the real part of all eigenvalues are negative, then the system trajectories will converge to a stable state if placed in the vicinity of that state. If any real part is positive an unstable trajectory exists for that state and the system will diverge if placed in the vicinity of that state and will not converge to a stable state where $\dot{\boldsymbol{x}} = 0$. For a linear system (2.20) the solution of $f(\boldsymbol{x}_0) = 0$ is always unique, meaning only one steady state exists for any linear system. The eigenvalues of J might reveal that this is an unstable steady state and the system will diverge away from this state (Khalil and Grizzle, 1996).

Non-linear systems might have more complex descriptions of the function $f(\boldsymbol{x}_0) = 0$, with multiple steady state solutions. This means that the system has multiple steady states, where some might correspond to converging states, while others might be unstable steady state. Here, *unstable* means that the system will naturally diverge from its steady state when small perturbations are introduced. This way of determining the behaviour of the steady state does not generalise to all nonlinear systems, but can be used for those that can be linearised around the steady state (Khalil and Grizzle, 1996).

The stability property has been incorporated in algorithms (Zavlanos et al., 2011) and when collecting data (Gardner et al., 2003) for inferring GRN. The assumption is that if biological systems would not be stable, even random variations would eventually accumulate within the system leading to a system collapse (Kremling and Saez-Rodriguez, 2007).

One simple mechanism in GRNs for maintaining stability is degradation. As every entity that regulates something else in the system will degrade or be diluted over time as a function of the concentration, an infinite growth can not be maintained. This is because an equilibrium will be reached depending on the growth rate and degradation rates of the molecules (Alon, 2007).

Linear vs. nonlinear models

Correct system representation is crucial, as different models will directly propagate unique properties and features. The model should capture important features of the underlying system while remaining intelligible, giving insight into how the system is assembled and predictions of behaviour under given conditions. Another practical consideration when choosing a representation is the possibility of evaluating or retrieving a solution, either analytically or computationally, of a model. Added complexity will often result in longer compute time or harder solution evaluation.

The following section will detail an example of two types of systems, a nonlinear representation developed to model enzyme kinetics and a linear representation as a simplified version of the nonlinear.

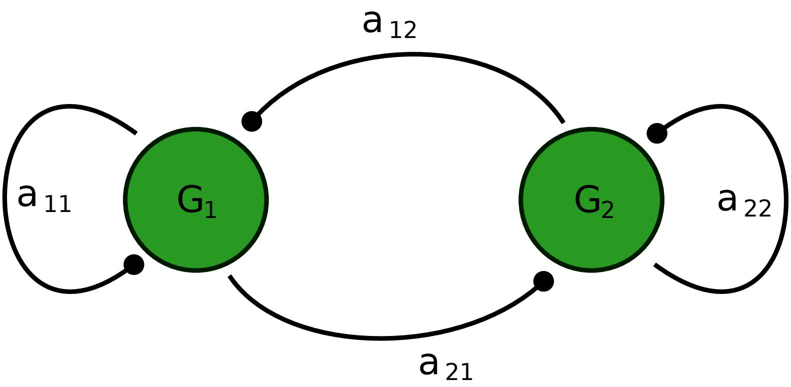

Figure 2.3: An abstract graphical representation of a mutual activating feedback circuit of two genes. The ball at the end of the link is a placeholder for an unspecified interaction, if an arrowhead is put there it means an activating interaction and if a T bar is put at the end it means a repression.

Figure 2.3 is the graphical, or network, representation of a two gene mutually regulating FBL. We can mathematically describe this system as a system of first-order ODEs,

$$\begin{aligned} \dot{x}_1 &= f_{G_1}(a_{11}, a_{12}, \alpha_1, x_1, x_2, \boldsymbol{K}_1) &= g_{x_1} \\ \dot{x}_2 &= f_{G_2}(a_{21}, a_{22}, \alpha_2, x_1, x_2, \boldsymbol{K}_2) &= g_{x_2} \end{aligned} \tag{2.6}$$

f_{G_*} is a function of choice that are chosen based on modelling assumption or purpose and could be different for different interactions. The parameters of model are $a_{11}, a_{12}, a_{21}, a_{22}$, α_1 and α_2. Any other parameters in the functions f are represented by $\boldsymbol{K}_i$. The states of the system is x_1 and x_2 represents some quantity related to the gene G_1 and G_2 respectively.

To simplify somewhat, lets look at an activating FBL with degradation without autoregulation and make $f(G)$ independent of the model parameters a_{ij} and α_i, then

$$\begin{aligned} \dot{x}_1 &= a_{12} f_{G_2}(x_2) - \alpha_1 x_1 &&= g_{x_1} \\ \dot{x}_2 &= a_{21} f_{G_1}(x_1) - \alpha_2 x_2 &&= g_{x_2} \end{aligned} \tag{2.7}$$

The degradation is here explicitly modelled as a linear effect on the measured quantity representing the gene itself. The rate of degradation is considered as decay of x_i and captured in the parameter α_i. If we incorporated autoregulation in the model, meaning that *e.g.* G_1 would regulate its own expression, with its gene products, we would need to incorporate the parameter a_{11}.

Now we can look at some properties of this system. First lets look at steady state. To find the steady states we set the rate $\dot{x}_1$ and $\dot{x}_2 = 0$ and solve for x_1 and x_2. To find the behaviour of this system close to its steady state (see: section 2.2.3 (Steady states)) we find the Jacobian matrix,

$$J = \begin{pmatrix} \frac{\partial g_{x_1}}{\partial x_1} & \frac{\partial g_{x_1}}{\partial x_2} \\ \frac{\partial g_{x_2}}{\partial x_1} & \frac{\partial g_{x_2}}{\partial x_2} \end{pmatrix} = \begin{pmatrix} -\alpha_1 & a_{12} f'_{G_1}(x_2) \\ a_{21} f'_{G_2}(x_1) & -\alpha_2 \end{pmatrix} \tag{2.8}$$

and behaviour of the steady state is described by the eigenvalues of the Jacobian. The eigenvalues are calculated by finding the λ of

$$\begin{gathered} |J - \lambda \boldsymbol{I}| = 0 \\ (-\alpha_1 - \lambda)(-\alpha_2 - \lambda) - (a_{12} f'_{G_1}(x_2))(a_{21} f'_{G_2}(x_1)) = 0 \end{gathered} \tag{2.9}$$

where $|.|$ is the determinant and $\boldsymbol{I}$ is the identity matrix. This will evaluate to a quadratic function with two solutions for λ, one for each eigenvalue. The eigenvalues are evaluated at the steady state, so that $f'_{G_1}(x_2)$ and $f'_{G_2}(x_1)$ are evaluated at the steady state (Hirsch et al., 2004).

Lets consider the case where f_G is the linear function for both G_1 and G_2. Then (2.7) will have four parameters a_{12}, a_{21} and α_1, α_2 and the steady state would look like

$$\begin{aligned} 0 &= a_{12} x_2 - \alpha_1 x_1 \\ 0 &= a_{21} x_1 - \alpha_2 x_2 \end{aligned} \tag{2.10}$$

and the steady state solution is

$$\begin{aligned} x_1 &= 0 \\ x_2 &= 0 \end{aligned}$$

and (2.9) will, depending on the parameters a_{ij} and α_i, be positive, negative or complex. Complex eigenvalues always comes in pairs. The real part of the eigenvalues $\Re(\lambda)$ determines if the system is stable (-) or unstable (+). The imaginary part $\Im(\lambda)$ determines the oscillatory behaviour of the system.

There is the special case when the steady state solution has the following form

$$\frac{\alpha_1\alpha_2}{a_{12}a_{21}} = 1 \tag{2.11}$$

$$\alpha_1\alpha_2 = a_{12}a_{21} \tag{2.12}$$

This is the case when J is singular. This means that infinte number of solutions exist for the steady state under these conditions. This is when the determinant of the jacobian $\det(J) = 0$ and any point in the null space of the system is a steady state (Khalil and Grizzle, 1996).

Now lets look at the nonlinear case when f_G is the Michaelis-Menten (MM) kinetics function. The MM function has been used to model GRNs before (August and Papachristodoulou, 2009). Other alternatives can be chosen as well, *e.g.* Hill kinetics or boolean functions. The MM function is

$$f_{G_i}(x_j) = \frac{x_j}{x_j + K_{ji}} \tag{2.13}$$

for an activator, and

$$f_{G_i}(x_j) = \frac{K_{ji}}{x_j + K_{ji}} \tag{2.14}$$

for a repressor, where j indicate the activator or repressor and i the target. K_{ij} is the activator coefficient which relates to the amount of x_j needed to be present until significant activation or repression is achieved. For MM the amount of x_j needed for 50% activation of its maximum.

To simplify lets look at mutual activation. The steady state equations from (2.6) will now be,

$$\begin{aligned} 0 &= a_{12}\tfrac{x_2}{x_2+K_{21}} - \alpha_1 x_1 \\ 0 &= a_{21}\tfrac{x_1}{x_1+K_{12}} - \alpha_2 x_2 \end{aligned} \tag{2.15}$$

We have a steady state at $[x_1, x_2] = [0, 0]$ however in this case this is not a unique solution, and we also have a solution at

$$\begin{aligned} x_1 &= \frac{S_{x_1} S_{x_2} - K_{12} K_{21}}{S_{x_2} + K_{21}} \\ x_2 &= \frac{S_{x_1} S_{x_2} - K_{12} K_{21}}{S_{x_1} + K_{12}} \end{aligned}$$

where $S_{x_1} = a_{12}/\alpha_1$ and $S_{x_2} = a_{21}/\alpha_2$.

Some notes on these observations. For nonlinear systems like the ones with MM kinetics there could exist more than one steady state. To be able to find the steady state behaviour a set of parameters for the model needs to be chosen.

This particular nonlinear system can not exhibit infinite growth as long as the degradation factor is considered. The growth rate will eventually be balanced out by the degradation factor.

Depending on if a specific combination of parameters in the equation (2.8) fulfils (2.11) the system becomes singular and an infinite number of solutions can be found for the steady state.

The nonlinear system that we explored had 6 parameters while the linear system had 4. Including autoregulation will increase the number of parameters for the nonlinear system to 10. For the linear system there is no differentiation between autoregulation and degradation, which is easily seen by adding autoregulation to equation (2.7). The effects are additive and not independently modelled and no differentiation can be made except that the degradation has a suppressing (-) effect and autoregulation can have an activating effect switching the sign of the interaction to be positive.

As mentioned before, some care needs to be taken when deciding what model to use to represent the system. While some features can not be captured by the linear model, such as bi-stability, the increase in complexity and degrees of freedom for the nonlinear system can risk creating models that do not represent the underlying biology and by extension increase the demand for more data.

Time separated hierarchical systems

Investigating hierarchies in systems helps us understand the behaviour of the system and can simplify further analysis. A dynamical system may work on several different time scales. The time constant τ can be derived from the eigenvalues of the jacobian, J, in essence estimating the scale of the effect of the system changes.

$$\tau_i \equiv \frac{1}{|\Re(\lambda_i)|} \tag{2.16}$$

where $\Re(\lambda_i)$ is the real part of eigenvalue λ for gene i.

Practically, the time constant is calculated for a nonlinear system around its steady state. Fast and slow modes can be separated either by eigenvalue spectral clustering or by imposing a threshold, τ^S on the time constant, so that if $\tau_i > \tau^S$, i belongs to the fast modes and to the slow otherwise (Kremling and Saez-Rodriguez, 2007).

Hierarchical analysis of system dynamics have been used to reduce dimensionality of the system (Zagaris et al., 2003). Time scale separation is implicated as being a cause of an interampatte behaviour of a system (Nordling and Jacobsen, 2009).

Time scale separation is sometimes a motivation for model reduction to facilitate a simpler model representation. When the time constants and associated dynamics can be viewed as the system operating in different time scales faster modes than the ones under observation can be considered as steady state and slower modes can be discarded as they are then independent of any changes in the time window (Kremling and Saez-Rodriguez, 2007).

Interampatte systems

Interampatteness is a property of biochemical networks that can be recognised by a highly correlated response to system perturbations (Nordling and Jacobsen, 2009). The degree of interampatteness can for liner systems be calculated as the condition number of the static gain matrix.

$$\kappa(\boldsymbol{G}) = \frac{\overline{\sigma}(\boldsymbol{G})}{\underline{\sigma}(\boldsymbol{G})} \tag{2.17}$$

where $\overline{\sigma}(\boldsymbol{G})$ is the largest singular value and $\underline{\sigma}(\boldsymbol{G})$ is the smallest singular value of $\boldsymbol{G}$.

Several data sets have been observed to be ill-conditioned. This is also the effect of doing measurements on an interampatte system. The data obtained from perturbing a 10 gene network of the *Snf1* pathway in *S. cerevisiae* (Lorenz et al., 2009) had a condition number, $\kappa = 253$, and a data set from a 9 gene network in *E. coli* (Gardner et al., 2003) had a condition number, $\kappa = 54$. The corresponding estimated interampatteness degree was $\kappa = 215$ and $\kappa = 154$ respectively.

Considering the inverse problem (section 2.3.3) it is known that the smallest signals in the system has the largest effect on the solution when trying to recover the system. The smallest signal is often the one most susceptible to noise corruption and by extension is the weak point of the inference. The perturbation design should counteract the interampatteness of the system under investigation as some responses could be masked by attenuation effects.

2.3 Systems biology

Systems biology aims to find descriptions of biological systems that takes into account the complex interactions that are typically found within *e.g.* the cellular regulatory network. A problem sought to be solved by a systems biology approach is to give a qualitative and quantitative understanding of the biological systems as a whole. Sub-problems concerning finding the behaviour and structure of regulatory networks need to be solved and taken in to account to reach this understanding.

The primary step to achieve this is to infer the structure of the network. This involves what is commonly known as a "top down" approach, contrasting the "bottom up" approach that traditionally means investigating singular regulatory interactions or the specific properties of a biomolecule. When most of the specific details of the biochemical reactions are known then a "bottoms up" approach can be appropriate to build up a view of the system and investigate emergent behaviour not observed or easily inferred from the parts of the system (Kremling and Saez-Rodriguez, 2007). This section will focus on a part of systems biology, namely the inference of causal network models describing gene regulatory networks (GRNs). First a brief overview of different model formalism will be given, second a more focused in depth view of linear dynamical models, and third its application to network inference of GRNs.

2.3.1 Model formalism

As described in section 2.2.1 we can describe a system generally as (2.1). Depending on the transfer function and response we can describe several different types of systems regularly used in systems biology. A whole slew of different approaches have been developed or adapted for network inference of GRNs.

Correlation based methods measure correlation between variables and infer a link between genes if the correlation is high enough. To be able to use correlation based methods to infer a directed regulatory network, and not just an association network, time series data needs to be used.

A similar approach is the information theoretic approach. The information theoretic approach is based on estimating the mutual information of the variation in the expression patterns of measured genes. The expression space could either be discretised to simplify calculations or used as is. This type of model extends to nonlinear relationships as mutual information can describe many types behaviours (Margolin et al., 2006).

Boolean networks links gene expression through boolean operators such as `AND`, `OR` and `NOT` (Albert and Othmer, 2003). Boolean interactions are based on the truth table of the interactors. This means that the expression of each gene needs to be discretised to determine if the gene is `ON` or `OFF` and can be expressed as,

$$\boldsymbol{x}(t+1) = f^B(\boldsymbol{x}(t)) \tag{2.18}$$

where f^B is a boolean function and $\boldsymbol{x}(t+1)$ is the state (`ON` / `OFF`) of the state variables at time $t+1$ as a function of the state, $\boldsymbol{x}$ at time t.

Bayesian models are models based on conditional probabilities. Due to the nature of conditional probabilities the bayesian model can not handle FBLs. To be able to model GRNs with feedback one need to extend the bayesian model to the dynamic bayesian models. The Bayesian network is modelled with conditional probabilities

$$\mathsf{P}(X_i = x_i | X_j = x_j) = f(x_i | x_j) \tag{2.19}$$

where x represent the specific value of the random variable X. For a network, one would evaluate the probability of a structure of relationships. Each network model would then be a product of conditional probabilities based on the structure of the network.

Another class of models is the ODE models (2.4). Several different models fall under this umbrella. An example of a nonlinear ODE is a model using Michaelis-Menten (MM) kinetics. This can be extended to include modelling with the cooperative Hill coefficients. The coefficients in the Hill function determine the steepness of the activation curve. This could also be replaced in the extreme case with a boolean condition, where activation turns on only if the concentration of some activation molecule reaches a certain level (Alon, 2007).

For the linear ODE the rate of change for each gene in the system is the cumulative effect of all other regulators for that gene. The linear system model will be discussed in detail in section 2.3.2.

There are several review articles describing different approaches and model formalism for network inference in systems biology, see *e.g.* (de Jong, 2002; Gardner and Faith, 2005; Hecker et al., 2009; Yaghoobi et al., 2012) for an overview of the main ones.

One should note that some care has to be taken to the choice of model for fitting the data. For a nonlinear model the degrees of freedom might not be well defined. Even for very simple models with few parameters very complex patterns of data can be fitted (Andrae et al., 2010). If any set of data can be fitted with the model there is no way of discriminating between competing models, and there is no test that can exclude a model over another, which should be required for a model to be considered descriptive.

2.3.2 Linear dynamical models

The benefit of using linear models is that they are simple and can describe various complex phenomena observed in biological systems such as *e.g.* feedback and feed forward motifs. Even if the system is nonlinear, as long as the system operates close to steady state a linear model can be approximated to describe the casual interactions.

A mathematical description of the linear system is as follows,

$$\begin{array}{rcl} \dot{x}_i(t) & = & \sum_{j=1}^{N} a_{ij} x_j(t) + p_i(t) - f_i(t) \\ y_i(t) & = & x_i(t) + e_i(t). \end{array} \tag{2.20}$$

If we are using the linear model in a biological systems context then the state vector $\boldsymbol{x}(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^T$ represents mRNA expression changes relative to the initial state we refer to as $t = 0$ of the system The vector $\boldsymbol{p}(t) = [p_1(t), p_2(t), \ldots, p_N(t)]^T$ represents the applied perturbation, which may be corrupted by the noise $\boldsymbol{f}(t)$. The perturbations could be *e.g.* gene knockdowns using siRNA or gene over-expressions using a plasmid with an extra copy of the gene. The response vector $\boldsymbol{y}(t) = [y_1(t), y_2(t), \ldots, y_N(t)]^T$ represents the measured expression changes that differ from the true expression changes by the noise $\boldsymbol{e}(t)$. a_{ij} represents the influence of an expression change of gene j on gene i. If gene j up regulates gene i then a_{ij} is positive and if gene j down regulates gene i then a_{ij} is negative. If gene j and i have no interaction then $a_{ij} = 0$.

Linear ODEs have been used extensively in the context of systems biology. It has been shown that nonlinear models can be linearised around a steady state or log-transformed to be able to make use of the properties associated with linear systems and that near steady state the kinetics are well described by a linear model (Crampin, 2006). However, that means that if we are not operating close to a steady state a linear model might give misleading conclusions. Until the quality of data is such that a clear discrimination between when a simple linear model can explain the data and when it cannot, extra care should be taken when, or if, choosing a more complex model.

Steady state data

For steady state data we can simplify (2.20) to

$$\boldsymbol{Y} = -\boldsymbol{A}^{-1}\boldsymbol{P} + \boldsymbol{A}^{-1}\boldsymbol{F} + \boldsymbol{E} \tag{2.21}$$

in matrix notation, when the set of experiments are considered. $\boldsymbol{Y}$ is the observed steady state response matrix with measurement error $\boldsymbol{E}$, when applying the perturbations $\boldsymbol{P}$. $\boldsymbol{F}$ is the difference between the true perturbations $\check{\boldsymbol{P}}$ and the observed perturbations $\boldsymbol{P}$, and $\boldsymbol{A}$ is the network represented as a matrix where each element represent an interaction. Linear systems with steady state data have been used in several network inference projects (Gardner et al., 2003; Julius et al., 2009; Tegnér et al., 2003).

Least squares estimate and prediction error

To find the ordinary least squares estimate of (2.21) we solve for $\boldsymbol{A}$,

$$\boldsymbol{A}_{ls} = -\boldsymbol{P}\boldsymbol{Y}^{\dagger} \tag{2.22}$$

Here $\dagger$ represents the Moore-Penrose generalised matrix inverse. If the data does not contain any noise we assume we can find an exact solution for $\boldsymbol{A}$. However in general, if we have collected noisy data a solution to the above can not be guaranteed and we need to find the least squares solution $\boldsymbol{A}_{ls}$ (Aster et al., 2005).

To fit the data one wants to find the parameters of the model that minimises the distance to the regression curve that relates the independent and dependent variables (Aster et al., 2005). This can be expressed with the following equation,

$$\hat{\boldsymbol{A}} = \arg\min_{\boldsymbol{A}} ||\boldsymbol{A}\boldsymbol{Y} + \boldsymbol{P}||_{L_2}^2 \tag{2.23}$$

If the noise in $\boldsymbol{F}$ and $\boldsymbol{E}$ are i.i.d. and normally distributed, $\mathcal{N}(\mu, \lambda)$ with mean μ and variance, λ, then the least squares estimate is also the maximum likelihood estimate (Hastie et al., 2009).

Equation (2.23) is sensitive to outliers due to the nature of the 2-norm, $\|.\|_2$ and it might be favourable to introduce the 1-norm instead

$$\hat{\boldsymbol{A}} = \arg\min_{\boldsymbol{A}} ||\boldsymbol{A}\boldsymbol{Y} + \boldsymbol{P}||_{L_1} \tag{2.24}$$

this norm corresponds to fitting to the median rather than the mean as in (2.23). For (2.23) the function is differentiable, but for (2.24) it is not. This problem can be overcome by noting that (2.24) is peace-wise differentiable and convex. Meaning that one can search for the optimal solution by finding the peace-wise optimal solutions (Aster et al., 2005).

2.3.3 Gene Regulatory Network inference

Gardner and Faith (2005) separated two types of network inference types, the first or "physical" approach aims at constructing the transcriptional regulatory network directly, *i.e.* to determine the physical binding of one transcription factor to another. This strategy concerns itself with direct chemical bonding interactions. In some cases however, it may be that an intermediate step is not observed and no direct binding occurs even though a change based on influence can be observed. The second approach is the influence strategy. For this approach the regulatory influences are sought rather than physical bindings.

As one of the primary objectives of network inference is to find the regulatory interactions, network inference is primarily a model selection problem and not a parameter estimation problem. However, this line is sometimes blurred with the introduction of algorithms, such as LASSO (Tibshirani, 1996), which both estimate parameters and return a selection of candidate models.

Several studies have employed a linear dynamical systems framework. Gardner et al. (2003) used a linear model, motivated by linearisation of a nonlinear model around a steady state. Furthermore data was recorded with a steady state assumption on the measured mRNA expression data for 9 genes in the SOS pathway in *E. coli*. A linear regression method was then used to estimate model parameters for an exhaustive search of subsets of interactors for each gene in the network.

A necessary condition to be able to infer a casual influence network from steady state data and a linear dynamical system, such as those in section 2.3.2, is that specific perturbations are made to each gene that is going to be included in the network. This is the case for time series data as well with the difference being that for time series data a single perturbation might be sufficient, and it does not necessarily need to be kept constant until the system relaxes to a steady state (D'haeseleer et al., 1999).

Penalised linear regression

Based on equation (2.23) and (2.24) we can see that the estimate of $\tilde{\boldsymbol{A}}_{ols}$ contains contributions from the noise matrices $\boldsymbol{E}$ and $\boldsymbol{F}$, even when assuming that the independent variable is noise free, $\boldsymbol{F} = 0$, we still have to deal with a noisy expression matrix $\tilde{\boldsymbol{Y}}$. The result of fitting the data with a noisy $\tilde{\boldsymbol{Y}}$, is that the estimated model $\boldsymbol{A}_{ols}$ tends to be overfitted, meaning that the parameters of the model fits the noise. This has the consequence that the model fitted to the data does not generalise well to other data with different noise realisations. For network inference it means that a link can be inferred in the network compensating for the effect of noise. A network like that is hard to interpret as it usually depicts every gene interacting with every other gene (Hastie et al., 2009). An approach to dealing with overfitting is to introduce a penalty term to the model fitting,

$$\hat{\boldsymbol{A}}_{\text{reg}}(\tilde{\zeta}) = \arg\min_{\boldsymbol{A}} ||\boldsymbol{A}\boldsymbol{Y} + \boldsymbol{P}||^2_{L_2} + \zeta||\boldsymbol{A}||_{L_2} \tag{2.25}$$

with ζ corresponding to a parameter that regulates the impact of the penalty term on the ordinary least squares estimate. The penalty term $\zeta||\boldsymbol{A}||_{L_2}$ penalises the model parameters squared size. This has a result that large parameters will be penalised more than smaller. This approach smooths the parameters of the models, however it does not eliminate model parameters well.

LASSO is another penalty method (Tibshirani, 1996). The lasso problem can be written as,

$$\hat{\boldsymbol{A}}_{\text{reg}}(\tilde{\zeta}) = \arg\min_{\boldsymbol{A}} ||\boldsymbol{A}\boldsymbol{Y} + \boldsymbol{P}||^2_{L_2} + \tilde{\zeta}||\boldsymbol{A}||_{L_1}. \tag{2.26}$$

The LASSO penalises the absolute size of model parameters. The difference from the ridge-regression is that LASSO produces different models depending on the penalty parameter ζ (Ng, 2004). The LASSO has the property that it combines model selection with parameter estimation. Due to this property LASSO has become very popular and a lot of work has been done on investigating the performance, such as its weakness on ill-conditioned data (Candès and Plan, 2009; Fan and Li, 2001; Jia and Rohe, 2012; Zhao and Yu, 2006).

As ridge-regression does not suffer the same weaknesses as LASSO, an effort to combine both of these penalties called *elastic-net* has been made. The Elastic-net (Zou and Hastie, 2005) method combines the L_1 penalty from LASSO and the L_2 penalty from ridge regression. The influence of the penalties are then weighted by a parameter α such that,

$$\hat{\boldsymbol{A}}_{\text{reg}}(\zeta) = \arg\min_{\boldsymbol{A}} C + \tilde{\zeta}\left(\alpha||\boldsymbol{A}||_{L_1} + (1-\alpha)||\boldsymbol{A}||^2_{L_2}\right), \tag{2.27}$$

where $C = ||\boldsymbol{AY} + \boldsymbol{P}||^2_{L_2}$. The elasic-net has been shown to be beneficial when compared to other algorithms to infer GRNs (Gustafsson and Hörnquist, 2010).

Zou (2006) extended the LASSO with the adaptive LASSO algorithm, which introduce a weighting term for each model parameter. These weights should be picked carefully and based on properties of the data, and should in theory be able to overcome the shortcomings of LASSO.

In (Julius et al., 2009) a structural constraint was introduced to the LASSO penalty derived from *a priori* knowledge where a link could be specified as being present or not present, positive, negative or uncertain. An additional constraint was introduced by Zavlanos et al. (2011) where the stability of the inferred network was ensured. In both cases a model similar to the one introduced in section 2.3.2 was used, with a steady state assumption.

Model selection

To choose a "good" model when inferring networks is not trivial. LASSO produces a range of different models depending on the regularisation parameter ζ.

As mentioned in section 2.3.3, overfitting is an issue when the data is noisy. The predictive performance of a network estimate can be calculated with the weighted residual sum of squares (RSS),

$$\chi^2(df) \sim \mathrm{W\,RSS}(\boldsymbol{A}_f) = (\boldsymbol{y} - \boldsymbol{A}_f^{-1}\boldsymbol{p})^T W^{-1} (\boldsymbol{y} - \boldsymbol{A}_f^{-1}\boldsymbol{p}) \tag{2.28}$$

where $\boldsymbol{A}_f$ denotes any network arrived at by any function, with covariance matrix W of the measurement errors. If the errors in $\boldsymbol{Y}$ are i.i.d. and normally distributed, $\mathcal{N}(\mu, \lambda)$ with mean μ and variance, λ, then the weighted RSS follows a χ^2 distribution with df degrees of freedom (Andrae et al., 2010; Aster et al., 2005). It is also possible to compare models to determine if one model is significantly better than another. The ratio of two reduced χ^2 distributions with degrees of freedom, df_1 and df_2,

$$R = \frac{\chi_1^2/df_1}{\chi_2^2/df_2} = \frac{\chi_1^2 df_2}{\chi_2^2 df_1} \tag{2.29}$$

will follow an F distribution with parameters df_1 and df_2. And a statistical test can be made to determine how much better one model is over the other (Aster et al., 2005).

To circumvent the over-fitting problem, one might employ a cross validation (CV) approach. CV means leaving out a part of the data, fitting the model to the remaining data and calculate (2.28) or simply the RSS on the left out data. This procedure is repeated for different portions of the data and the error is calculated each time.

Due to the statistical properties of the weighted RSS it is suitable for goodness of fit testing. If the error is significantly larger than expected the model is discarded.

The prediction error approach is used in the Inferelator (Bonneau et al., 2006), a network inference framework, together with a CV scheme to select a model with sufficiently good performance. The common assumption that GRNs are sparse is used and motivates a selection of a prediction error one standard deviation above the minimum prediction error for selecting the network that is more sparse.

Two other approaches for model selection are the Bayesian information criterion (BIC) and the Akaike information criterion (AIC) (Akaike, 1973). Both approaches is based on the likelihood function, the BIC can be written as

$$\mathrm{BIC} = m \ln\left(\frac{\mathrm{RSS}}{m}\right) + k\ln(m) \tag{2.30}$$

where m is the number of data points, and k the number of free parameters to be estimated.

Both BIC and AIC makes a trade off between a models prediction capability on the training data and model complexity, and both methods have in some cases been shown to perform worse than CV (Thorsson and Michael, 2005).

Inverse problems

Aster et al. (2005) describes the nature of the inverse problem, which arises when one tries to estimate model parameters based on measured data or observations related to some independent variables. This includes the network inference problem and relates to the inference problem's sensitivity to noise.

Looking at equation (2.22) we can decompose the matrix $\boldsymbol{Y} = \boldsymbol{U\Sigma V}^T$, which is just a linear combination of the singular values σ_k and the singular vectors, $\boldsymbol{v}_k\boldsymbol{u}_k^T$, where k is the specific singular value. Now the inverse of $\boldsymbol{Y}$, can be written as another linear combination of these entities,

$$\boldsymbol{Y}^\dagger \equiv \sum_{k=1}^{n} \frac{1}{\sigma_k}\boldsymbol{v}_k\boldsymbol{u}_k^T \tag{2.31}$$

which means that the singular value that has the largest effects on the estimate of (2.22) is the smallest singular value of $\boldsymbol{Y}$. The smallest singular value represents the direction in the data with the least variation and least information, meaning that the influence of the noise $\boldsymbol{E}$ is potentially substantial as the noise corrupts the smallest variation easier.

From equation (2.31) we can derive a definition for an upper bound on the global signal to noise ratio (SNR), where

$$\text{SNR} \equiv \frac{\underline{\sigma}(\boldsymbol{Y})}{\overline{\sigma}(\boldsymbol{E})} \tag{2.32}$$

and the variables are defined as in (2.21) with $\overline{\sigma}$ representing the largest singular values and $\underline{\sigma}$ representing the smallest non zero singular value. This can be understood as the largest possible effect the noise can have on the smallest singular value of the measurements. In practise we do not have access to $\boldsymbol{E}$ and we then define the SNR based one the estimated variance of the noise,

$$\text{SNR} \equiv \frac{\underline{\sigma}(\boldsymbol{Y})}{\sqrt{\chi^{-2}(\alpha, df)\lambda(\boldsymbol{Y})}} \tag{2.33}$$

$\chi^{-2}(\alpha, df)$ is the inverse of the χ^2 distribution at α significance level and df degrees of freedom. $\lambda(\boldsymbol{Y})$ is the variance of the noise or measurement error of $\boldsymbol{Y}$.

2.4 Network inference – community efforts

The field of network inference has amassed a collection of tools from various scientific disciplines and a scientifically diverse group of individuals constitutes the network inference community.

In this section I will describe some of the efforts, resources and approaches that has been built around this research field and how they are connected, as well as giving a reference list of different tools that have been developed in the systems biology field.

2.4.1 Benchmarks

Benchmarking can be used as a tool for evaluating the performance of algorithms or methods trying to solve specific problems. Usually, introducing a new algorithm demands that the claims made of its usefulness is accompanied by a benchmark, a test against other competing methods or algorithms or some test of performance on data that can be compared to previous estimates (Friedman et al., 2010; Lauria et al., 2009; Margolin et al., 2006). However, it might be the case that new information or better data becomes available at a later date or that the application for the method is expanded. For this reason larger benchmarks are often conducted with a larger scope than provided by the original analysis (Bansal et al., 2007; Penfold and Wild, 2011). These benchmarks have the aim of exploring the performance of methods tested under both a realistic and wide range of conditions as well as against methods of different types and requirements.

Two classes of data are often collected in relation to GRN inference, steady state data and time series data. Different assumptions follow these different data types. For steady state data one needs to measure and perturb every gene to be included in the inferred network, see (2.20). For time series data not all genes needs to be perturbed but enough data needs to capture the regulatory effects in short and long term (Hecker et al., 2009).

One can focus on one of these data types when benchmarking algorithms *e.g.* time series data (Narendra et al., 2011; Ward et al., 2009) or mix different approaches that use both types of data (Bansal et al., 2007; Penfold and Wild, 2011). The advantage of mixing data is that one can evaluate if any data approach is more informative and if any method approach are to be preferred. The advantage of not mixing data types is that one can more easily isolate specific factors that can influence a specific algorithms performance for a specific data type.

Another feature of the data is the underlying model assumptions. To make the data more realistic, a model based more closely on the underlying theory of how the system operates might be used. Different model assumptions demand different types of data, whether it is to simulate *in silico* data or to decide what data needs to be collected from an *in vivo* setup (Gardner and Faith, 2005). For example, when considering Boolean networks, if the regulatory structure of the network is such that a gene can not be "turned on" one can not collect all different combinations of inputs required to make a truth table for the inference. The more regulators the more risk that not all combinations can be realised trivially, and the more data needs to be collected.

The DREAM challenge is a community effort and competition that aims at combining the previously mentioned features of benchmarking in addition to including a large contributing community (Marbach et al., 2012). The challenges go back to 2007 and has evolved over time. The DREAM challenge is split into several different challenges where one or more are focused on network inference, or identifying unknown regulatory interactions with the help of data and a partly complete network. The challenges present a mix of *in silico* and *in vivo* data and with some exceptions makes the data available for use when the challenge has finished for use in other works (Folch-Fortuny et al., 2015).

Another core part of any benchmark is how to evaluate the performance of an algorithm being tested and evaluating strengths and weaknesses of methods and approaches. As a primary aim of network inference is to find the regulatory structure of the GRN one usually tests true positive (TP), false positive (FP), true negative (TN) and false negative (FN), where positive represents a link and negative the absence of a link. True and false represent the classification an inference method has made if a link should be present or not given that it exist in the gold standard network. These measures are usually summarised in a more easily interpretable form, such as a fraction of the measures that range between 0 and 1, *e.g.* sensitivity $= \frac{TP}{TP+FN}$, precision $= \frac{TP}{TP+FP}$, specificity $= \frac{TN}{TN+FP}$ and negative prediction value $= \frac{TN}{TN+FN}$ (Bansal et al., 2007). What one would like is a single number that represents the performance and is easily compared and understood. The area under receiver operator curve (AUROC) and the area under precision recall curve (AUPR) are used in many benchmarks, see for example, (Marbach et al., 2012, 2010; Narendra et al., 2011). Some examples of incorporating the sign of the link has been made (Hache et al., 2009a). This means extending the binary classification into a more complex structure where you take in to account a link, which is inferred but with the wrong sign.

Cantone et al. (2009) generated an *in vivo* data set from an engineered network. The network was tuned so that the interactions would be known and the network was perturbed and the response was measured both for steady state and time series data. The purpose of this data set was to be able to benchmark methods on a realistic true model with actual measured data. Even during these conditions it is shown that inferring the true network is difficult (Penfold and Wild, 2011).

2.4.2 Data and experiments, *in silico* vs. *in vivo* for benchmarking

A large collection of toolboxes has been developed aimed at systems biology research. which focuses mainly on creating simulated GRNs see for example:(Hache et al., 2009b; Schaffter et al., 2011; Van den Bulcke et al., 2006).

This is a response to the fact that regulatory networks in biology are generally lacking in information and are one of the least available networks types (Barabasi et al., 2011). This has to be paired with available data suitable for network inference under stable enough conditions so that the change in the states observed in the data is a consequence of regulatory effects and not for example the network being in a specific mode or that a part of the network is missing, which can happen if genes are deleted. Toy models and *in silico* generated data have been shown to be a good proxy for estimating performance of network inference algorithms (Bansal et al., 2007). *In silico* models have been used to predict and tune optimal evolutionary growth through the metabolic network (Ibarra et al., 2002). It is also beneficial if one can prepare or extend experimental procedures by first running simulations on a computer and many times it is necessary to be able to maximise the usefulness of the *in vivo* experimental output (Nordling, 2013).

Another benefit of being able to use simulated data is that it is easier to explore and examine a wider range of properties of both network and data. Networks with different structure and different number of motifs can be generated and methods can be tested on how they perform during specific conditions (Marbach et al., 2012).

If some knowledge exists, even partial knowledge, one can incorporate this information to get more realistic data sets, such as known regulatory networks (Schaffter et al., 2011).

For *in vivo* generated data the concern about ”realistic” models or experimental conditions, such as realistic noise models, system response patterns or network structure are taken care of. Therefore it is desired to generate data in living systems even when testing methods. For these systems a gold standard network might not exist to estimate network inference performance. There has been several successful attempts of both data generation and inference including *in vivo* data and proposed true GRN (Cantone et al., 2009; Gardner et al., 2003; Lorenz et al., 2009). However, even these data sets can be shown to have low quality, such as having low SNR and being ill-conditioned, indicating that there is still work to be done for generating *in vivo* data sets suitable for GRN inference.

2.4.3 Tools of systems biology

In a research field that rely heavily on computation it’s unavoidable that a huge number of lines of code and data is generated. In addition to the scientific knowledge generated with these tools, they are themselves a valuable contribution to the body of scientific knowledge. In this section I will try to collect a number of different tools used in systems biology with the aim of helping with GRN inference. The tools cover mainly three different areas. (i) Algorithms and methods for inferring networks, which is the main area of tool development. Without them the goals of systems biology could not be reached. (ii) Data formats and communications. To be able to share data and communicate results and information, common data formats should be developed. (iii) Simulation and benchmarking. These tools should accompany any inference method so that it can easily be evaluated.

Table 2.1 gives an overview of inference methods. The list is not meant to be exhaustive but instead to give a wide overview of the different approaches available. For each method the short and long names are given, if available. The goal of the algorithm together with the modelling scheme is also listed.

Table 2.2 lists a number of tools used for *in silico* simulation and modelling. As detailed in section 2.4.2, the demand for testing the array of network inference methods is facilitated by tools that can generate simulated data and networks.

Table 2.3 lists tools and formats for sharing and communicating systems biology data and knowledge.

Table 2.1: List of network inference methods. Short name is the name usually used to refer to the method.

Reference	Short Name	Description	Model Scheme	Goal
di Bernardo et al. (2005)	MNI	Mode-of-action by network identification		Determine drug targets
Julius et al. (2009)		LASSO based convex programming implementation with prior constraints	ODEs	GRN
Greenfield et al. (2010)	MCZ	Median Corrected Z-Scores	Information-theoretical	GRN
Pinna et al. (2010)		Graph-based method	Z-score-based	GRN
Grimaldi et al. (2011)	RegnANN	Reverse engineered gene networks with artificial neural networks	neural networks	GRN
Zavlanos et al. (2011)		Inferring stable genetic networks from steady-state data	linear dynamical systems	GRN
Xiong and Zhou (2012)		Method with regression and correlation	Info-theoretic / LDS	GRN
Gardner et al. (2003)	NIR	Network identification by multiple regression	ODEs	GRN & identify drug targets
Friedman et al. (2010)	Glmnet	Lasso (L1) and elastic-net regularized generalised linear models		Linear regression
	LSCO	least squares with cutoff		
Faith et al. (2007)	CLR	Context likelihood of relatedness	Information-theoretical	GRN
Jörnsten et al. (2011)	EPoC	Endogenous perturbation analysis of cancer		GRN

Continued on next page

Continued from previous page

Reference	Short Name	Description	Model Scheme	Goal
Shih and Parthasarathy (2012)		Single source k-shortest paths algorithm	graph theory	GRN
Menéndez et al. (2010)	GMRF	Graphical lasso with Gaussian Markov Random Fields	relevance based	GRN
Zou (2006)		Adaptive lasso		
Fan et al. (2009)		SCAD penalty		
Nordling and Jacobsen (2011)		Rank Reduction	linear ODE	GRN
Wang et al. (2012)		Inference with kalmar filter	combined linear and logistic	GRN
Nordling (2013)	RNI	Confidence based Robust Network Inference		GRN
Wu and Lange (2008)	CCD	Cyclic coordinate descent Lasso solver		
Cosgrove et al. (2008)	SSEM-Lasso	Sparse simultaneous equation model – Lasso regression		Determine drug targets
Oates et al. (2012)		Bayesian network using Goldbeter Koshland kinetics	Bayesian	Protein-signalling network
Lauria et al. (2009)	NIRest	NIR with perturbation estimate	ODEs	estimate P, identify GRN
Margolin et al. (2006)	ARACNE	Algorithm for the reconstruction of accurate cellular networks	Information-theoretical	GRN
Küffner et al. (2012)	ANOVA	ANOVA	ANOVA	GRN
Huynh-Thu et al. (2010)	GENIE3	Tree-based method	Tree-based	GRN
Castelo and Roverato (2009)	Qp-graphs	Q-order partial correlation graphs	graph theory	GRN

Continued on next page

Continued from previous page

Reference	Short Name	Description	Model Scheme	Goal
Ambroise et al. (2012)	TNIFSED	Supervised transcriptional network inference from functional similarity and expression data	supervised	Assign probability of being target of each TF
Mordelet and Vert (2008)	SIRENE	Supervised inference of regulatory networks	supervised	Assign targets to TFs
Sun et al. (2007)	TRND	Transcriptional regulatory network discovery	Bayesian	Assign targets to TFs
de Matos Simoes and Emmert-Streib (2012)	BC3NET	Bootstrap aggregation ensemble C3NET	Information-theoretical	GRN
Altay and Emmert-Streib (2011)	C3NET	Conservative causal core network inference	Information-theoretical	GRN
Friedman et al. (2008)		Graphical lasso		Sparse inverse covariance estimation
Bonneau et al. (2006)	Inferelator	the Inferelator	ODEs	GRN
Gevaert et al. (2007)		Bayesian network inference with prior data	Bayesian	GRN
Lähdesmäki et al. (2008)	RJMCMC	Reversible jump Markov chain Monte Carlo	Bayesian	GRN
Nelander et al. (2008)	CoPIA	Combinatorial Perturbation-based Interaction Analysis	ODEs	GRN
Yip et al. (2010)		Integration of knockout and perturbation data	ODEs	GRN
Yu et al. (2004)	BANJO	Dynamic Bayesian network inference	Bayesian	GRN
Djebbari and Quackenbush (2008)		Seeded Bayesian networks	Bayesian	GRN

Continued on next page

Continued from previous page

Reference	Short Name	Description	Model Scheme	Goal
Äijö and Lähdesmäki (2009)		Dynamic Bayesian network inference with Guassian processes	Bayesian	GRN
Chai et al. (2013)		Dynamic Bayesian network inference with imputed missing values	Bayesian	GRN
Wang et al. (2010)		[Boolean] Process-based network decomposition	Boolean	GRN or motifs
Schulz et al. (2012)	DREM	Dynamic Regulatory Events Miner		More TF-target and timing than GRN
Hache et al. (2007)	GNRevealer	Reconstructing GNRs with neural networks	neural networks	GRN
Kabir et al. (2010)		Linear time-variant method using self-adaptive differential evolution		GRN
Küffner et al. (2010)	PNFL	Petri net with fuzzy logic	petri net	GRN
Grzegorczyk and Husmeier (2013)		Non-homogeneous dynamic Bayesian network	Bayesian	GRN
Wu et al. (2011)	SSM	State space model w/hidden variables	state space model	GRN
Penfold et al. (2012)		Hierarchical non-parametric Bayesian	Bayesian	GRN
Böck et al. (2012)		Hub-centered GRN inference using automatic relevance	Bayesian	GRN or hubs
Layek et al. (2011)		Boolean networks represented by Karnaugh maps	Boolean	GRN

Continued on next page

Continued from previous page

Reference	Short Name	Description	Model Scheme	Goal
Kimura et al. (2012)	LPM	Linear program machine-based S-system GRN inference method	S-system	GRN
Alakwaa et al. (2011)	BicAT-Plus	Bi-clustering with Bayesian for GRN inference	Bayesian	GRN
Li et al. (2011)	DELDBN	Differential Equation-based Local Dynamic Bayesian Network	Dynamic Bayesian	GRN
August and Papachristodoulou (2009)		Linear convex solver program for biochemical nonlinear network inference	ODE	GRN
Yuan et al. (2011)		Robust network structure reconstruction	ODE's/LDS	GRN
Zhang et al. (2012)	NARROMI	Noise and redundancy reduction technique using recursive optimisation and mutual information	Info-theoretic and ODEs	GRN

Table 2.2: Simulation and benchmark data generation tools used for network inference.

Reference	tool	modelling
Schaffter et al. (2011)	GeneNetWeaver	Non-linear regulatory networks
Villaverde et al. (2015)	BioPreDyn-bench	Ready to run benchmarks
Hache et al. (2009b)	GeNGe	Non-linear regulatory networks
Van den Bulcke et al. (2006)	SynTReN	Non-linear regulatory networks
Di Camillo et al. (2009)	netsim	Non-linear regulatory networks

Table 2.3: Tools for used in systems biology to facilitate communication and results.

Reference	tool	usage
Almeida et al. (2003)	SBML	data format
Miller et al. (2010)	CellML	data format with related simulation tools
MATLAB (2014)	SimBiology	simulation and programming
Schmidt and Jirstrand (2006)	SBToolbox	simulation and programming
Hoops et al. (2006)	Copasi	Dynamic model exploration
Bellot et al. (2015)	NetBenchmark	Collection of benchmarking tools

3. Present investigations

3.1 Model sparsity selection based on minimum prediction error (PAPER I)

Optimal model selection is an open problem. How to properly choose a specific set of parameters for the network inference algorithms to determine the optimal sparsity has not been solved.

Some classical alternatives proposed are the BIC and AIC, which both trade-off prediction and complexity to find an optimal model, as well as CV and selection based on minimisation of the RSS.

All these methods for model selection are motivated by the fact that data is recorded with noise and that over-fitting the model is always a risk. The selection methods have been shown to perform well asymptotically with *e.g.* the number of samples (Stoica and Selen, 2004)

In this paper we studied the effects on model selection when the data had a varying degree of information and few samples, typically no higher than twice the number of variables. Information in the data was defined based on the optimal performance of the inference method robust network inference (RNI) on the data when compared to the true network. If the performance matched the true network for the best model produced by the method, the data set would be considered informative. When the performance was nonoptimal, but better than random the data set was deemed partly informative, and if the performance was no better than random the data was labelled uninformative. The informativeness was varied based on two factors, (i) the properties of the network and experimental design, (ii) the SNR.

The data used was generated *in silico* as this had been utilised with success previously and been shown to be an good indication of how a method would perform on other data (Bansal et al., 2007; Menéndez et al., 2010).

We determined two additional steps that should be utilised when solving a network inference and model selection problem. First, we showed that to be able to utilise a leave out cross validation approach, or as we employ it here, a leave *one* out cross optimisation (LOOCO), one needs to test for dependence of the sample on the rest of the data and only include the sample in the left out group if it is sufficiently described by the data that is going to be used to infer a network. The reason for this is that a network inferred from data with no information of a left out sample cannot make any predictions about that sample. Secondly we introduced a step of re-estimating the parameters returned from an inference algorithm. Here we argued that because of the penalty used in many inference methods, to combine model selection and data fitting, the parameters of the model are not the maximum likelihood estimates, which may skew the RSS for the predictions. The algorithm for re-estimating the parameters are a constrained least squares (CLS) algorithm. CLS preserves the structure of the network while refitting the parameters. We showed that if the data was uninformative we could not make a useful reliable model selection. If the data was partly informative or informative, the model selection based on the RSS would find the model that minimised the FPs conditioned on the TPs being maximised. In practice, giving our selection method a boundery where the minimum RSS could only be achieved when all TPs were present.

3.1.1 Future perspective

We showed that conceptually our approach worked. However, we did not investigate the performance in general and what the behaviour of our approach would be for a wide variety of data properties. Several technical additions to a new study would greatly benefit this investigation.

We did not test the BIC and AIC selection methods. Both of these methods are dependent on the likelihood function and should therefore also have their performance influenced by our additional steps.

The RSS was calculated as the mean RSS over all the selected leave out samples. A new study would greatly benefit from utilising the statistical properties of the RSS, such as if the error of the measurements are assumed to be normal, the RSS will follow a χ^2 distribution. With some care when estimating the degrees of freedom for each model (Andrae et al., 2010) an exclusion step could then be made where all models not passing a goodness of fit test would be excluded as candidate networks. The result would be a set of candidate networks in which we could in theory pick any of them. We would expect, though, that we would pick the sparsest candidate with the argument that GRNs are, in general, sparse.

3.2 Including prior information to enhance network inference accuracy (PAPER II)

In this paper we investigated if one could improve inference methods with the help of including prior information.

It is often the case that when trying to solve a network inference problem within biology, the data is under-determined. This means that a unique solution can not be found for regression models. It is also usually the case when dealing with biological data that the SNR is low, or that very few replicates have been recorded.

In both these situations it may be beneficial to include prior information. In the first case, if we include prior structural knowledge of the regulatory interactions, we can constrain the problem to a subset of interactions so that it no longer becomes under-determined. In the second case we might have knowledge that we are confident about of which interactions are more likely to exist and that can help guide an inference method when the data is of poor quality. In this paper we investigated the latter case.

Available on-line there are a number of databases containing functional associations between genes, collected from a wealth of sources with a number of different evidence types (Schmitt et al., 2014; Szklarczyk et al., 2011).

Incorporating a prior in the network inference pipeline can be done in a number of ways. In this study we focused on incorporating functional associations, which are usually represented by a number of the confidence that is associated with a link. These associations are by their nature undirected. It is often unknown if they are representing direct or indirect links, and if they are parallel or serial. Therefore, we opted for including the confidence of links as weights inversely proportional to the confidence, meaning that links that have a high confidence give a low weight to the associated penalty term, giving the link a higher chance of being selected. For example, if the confidence is low but the data indicates a strong link, both effects are traded against each other. By incorporating the associations as weights it gives the possibility of the data to speak as well.

To test the performance of using a prior in the network inference pipeline, a number of different networks and *in silico* data sets were generated. Two different models of system and data were used, a linear system model and a nonlinear system model (Schaffter et al., 2011).

Prior incorporation performance was tested by changing the prior accuracy. Accuracy of the prior was controlling how true links were drawn from distributions of low and high confidence associations.

When the data was uninformative an improvement with the prior could be observed if the prior was more correct than not. For data generated with the linear model the prior needed on average to be more correct than for a nonlinear model. This also scaled with the SNR of the data sets which in general was higher for the linear system vs nonlinear.

We also wanted to test the prior incorporation on real data and used a data collected from *S. cerevisiae* with the gold standard network collected from the Yeastract database (Teixeira et al., 2013). To estimate the performance, we checked the overall performance for all models generated by the inference method. We did this to remove the factor of trying to pick the correct sparsity for the network inference method. An improvement with the prior could be observed over almost all sparsity levels with an emphasis on the sparser range of the spectrum where we would assume that the optimal network should be found.

3.2.1 Future perspective

One question that was not answered in this paper was, at what quality of the data is it useful to include a prior? While the accuracy of the prior was investigated, the range of SNR was not. This could prove useful when the accuracy of the prior or the nature of the prior *e.g.* being undirected, might obstruct the inference algorithm.

Due to the evidence types of the prior, the associations might be indirect. A modified algorithm could make use of this information and instead of inserting a confidence as a weight of an interaction, the association could be incorporated in a way so that the association is preserved in the inferred network even though no direct link would exist, reflecting the nature of the association.

3.3 Practical workarounds for the pitfalls of L1 penalised regression methods (PAPER III)

It is known that the performance of penalised regression methods, specifically the L_1, penalised *e.g.* least absolute shrinkage and selection operator (LASSO), algorithm perform poorly under some conditions (Zhao and Yu, 2006). Sometimes referred to as the predictors having a high co-linearity or the data being ill-conditioned. In systems theoretic terms this can be quantified by calculating the condition number of the data set. An ill-conditioned matrix has a high degree of co-linearity.

The observation here is that even when the data is informative, defined as in PAPER I, section 3.1, the L_1 penalised methods perform as if the data were only partly informative even when we act as if we had expert knowledge when selecting the optimal network produced by the inference method. The performance of these types of inference method have been investigated and been shown to be a function of the data and network (Marbach et al., 2012; Zhao and Yu, 2006). The issue with these results is that they are impractical in reality as we do not know the network structure beforehand and in some cases we would arrive at the wrong conclusions if we used the wrong network structure to calculate them.

We show that a proxy for predicting the performance of an inference method is to investigate the properties of the data, specifically the condition number and the SNR.

We use synthetic data to vary the properties of both network and *in silico* expression data. We constructed the data so that the properties ranged over known values of properties for real biological data sets. The properties of the expression data is highly dependant on the network properties but they can be tuned depending on the experimental design (Nordling and Jacobsen, 2009). This is demonstrated with 3 different experimental designs. Two of the approaches could easily be employed in practise and show specifically that these designs made the data properties highly dependent on the network properties. The third approach would be more complex and costly to implement in practise and is aimed at minimising the condition number for the expression matrix. It demonstrated clearly that de-coupling the data and network properties and tuning the input so that the data properties would approach more desired states greatly enhanced the performance of the inference and network construction.

While few real data set exists with sufficient data to quantify the properties used in this work and simultaneously have a reference regulatory network, we picked one data set derived from over expression experiments with three proposed regulatory networks derived experimentally. From the properties of the data we could reasonably well predict the performance of the inference methods, by comparing to the performance on the *in silico* data.

3.3.1 Future perspective

One aspect that is rarely incorporated in GRN inference algorithms is the errors-in-variables aspect. Errors-in-variable models consider measurement errors perturbations, as well as in the measurements. It is easy to imagine that not only does a perturbation experiment contain noise in the measurement, but in the state of the system when the perturbation is applied as well. The effect of not considering measurement errors in the independent variables when an error exists has, as far as I know, not been studied within systems biology and GRN inference.

Methods that incorporate total least squares (TLS), which considers errors in variables, opposed to least squares (LS) methods, are few and rarely used.

A study on the effect of this could give insight on how to approach this issue and optimise performance on inference with these considerations.

3.4 GeneSPIDER, a software package for a simplified network inference pipeline (PAPER IV)

GeneSPIDER is a software package developed in the computer language and environment MATLAB (2014). The goal of GeneSPIDER is to provide a simple interface for testing algorithms for network inference of GRNs, as well as being able to analyse data acquired from experiments to gain insight into how to proceed with an investigation.

GeneSPIDER provides functionality for benchmarking network inference methods by generating artificial networks and simulating perturbation experiments on those networks and for measuring performance. GeneSPIDER also provides functionality to analyse real world data and guide experimental design. These two concepts are tightly connected. Previous benchmark packages have often focused on generating complex models aimed at being as realistic as possible while simulating standard perturbation experiments, like single gene knockout or knockdown. However, it has been shown that network inference algorithms perform poorly on data generated from simple models with noise levels similar to those found in real data. This problem is related to experimental design and can be investigated by using simpler models.

GeneSPIDER takes the approach that it is as important to find out why network inference methods fail as it is creating realistic models. Models aimed at being realistic are usually very complex, meaning that it can be hard to elucidate or isolate variables that have a direct effect on the performance of the inference. It is also unclear if a more complex model gives qualitatively better simulations, where simpler models do not give any insight. In the lab the researcher often has very little control over the network and network properties. However, the experimental design is under the researchers control to a larger extent than the hidden system under investigation. Therefore, it makes sense to also investigate what experiments give the most informative data. This has been done to a large extent in the systems theory field, but it has not been extensively incorporated in benchmarking toolboxes related to GRNs.

GeneSPIDER aims to provide a platform to bridge this gap, with the possibility of investigating optimal perturbation design being as accessible as model simulation. It is built on previous work and as such provides functionality to solve the problems encountered therein.

3.4.1 Future perspective

GeneSPIDER could be extended to incorporate more variations of expression data *e.g.* time series data experiments. This kind of data is also available to the network inference community and suffer many of the same shortcomings as steady state data when considering experimental design toolboxes.

Many functions of GeneSPIDER are under development, such as optimal perturbation design, and therefore programmatically not optimally implemented. This is partly because the problem formulation is not finalised. Further work on how to formulate and implement different details of experimental designs and error estimation of both input and output variables and incorporating that in to GeneSPIDER is on the TODO list for the software package.

References

Akaike, H., 1973. Maximum likelihood identification of gaussian autoregressive moving average models. *Biometrika*, 60(2):255–265.

Alakwaa, F. M., Solouma, N. H., and Kadah, Y. M., 2011. Construction of gene regulatory networks using biclustering and bayesian networks. *Theoretical Biology & Medical Modelling*, 8:39–39. 1742-4682-8-39[PII].

Albert, R. and Othmer, H. G., 2003. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster. *Journal of Theoretical Biology*, 223(1):1 – 18.

Almeida, J. S., Wu, S., and Voit, E. O., 2003. Xml4mat: Inter-conversion between matlab tm structured variables and the markup language mbml.

Alon, U., 2007. *An introduction to systems biology: design principles of biological circuits*, vol. 10 of *Chapman & Hall/CRC mathematical and computational biology series*. Chapman & Hall/CRC, 1 edn.

Altay, G. and Emmert-Streib, F., 2011. Structural influence of gene networks on their inference: analysis of c3net. *Biology Direct*, 6(1):31.

Amberger, J., Bocchini, C. A., Scott, A. F., et al., 2009. Mckusick's online mendelian inheritance in man (omimâ?). *Nucleic Acids Res*, 37(Database issue):D793–D796. Gkn665[PII].

Ambroise, J., Robert, A., Macq, B., et al., 2012. Transcriptional network inference from functional similarity and expression data: a global supervised approach. *Statistical Applications in Genetics and Molecular Biology*, 11(1):1–24.

Andrae, R., Schulze-Hartung, T., and Melchior, P., 2010. Dos and don'ts of reduced chi-squared. Tech. Rep. arXiv:1012.3754.

Aster, R., Borchers, B., and Thurber, C., 2005. *Parameter Estimation and Inverse Problems*. IG/International Geophysics Series. Elsevier Science.

August, E. and Papachristodoulou, A., 2009. Efficient, sparse biological network determination. *BMC Systems Biology*, 3(1):25.

Bansal, M., Belcastro, V., Ambesi-Impiombato, A., et al., 2007. How to infer gene networks from expression profiles. *Molecular systems biology*, 3(78):78.

Barabasi, A.-L., Gulbahce, N., and Loscalzo, J., 2011. Network medicine: a network-based approach to human disease. *Nat Rev Genet*, 12(1):56–68.

Barabasi, A.-L. and Oltvai, Z. N., 2004. Network biology: understanding the cell's functional organization. *Nat Rev Genet*, 5(2):101–113.

Bellot, P., Olsen, C., Salembier, P., et al., 2015. Netbenchmark: a bioconductor package for reproducible benchmarks of gene regulatory network inference. *BMC Bioinformatics*, 16(1):312.

Berg, H. C., 2000. Motile behavior of bacteria. *Physics Today*, 53(1):24–30.

Bonneau, R., Reiss, D. J., Shannon, P., et al., 2006. The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome biology*, 7(5):R36.

Brown, T. A., 2002. *Genomes*. Oxford : BIOS, 2nd edn. Previous ed.: 1999.

Böck, M., Ogishima, S., Tanaka, H., et al., 2012. Hub-centered gene network reconstruction using automatic relevance determination. *PLoS ONE*, 7(5):e35077.

Candès, E. J. and Plan, Y., 2009. Near-ideal model selection by ℓ_1 minimization. *The Annals of Statistics*, 37(5A):2145–2177.

Cantone, I., Marucci, L., Iorio, F., et al., 2009. A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. *Cell*, 137(1):172–181.

Castelo, R. and Roverato, A., 2009. Reverse engineering molecular regulatory networks from microarray data with qp-graphs. *Journal of Computational Biology*, 16(2):213–227.

Chai, L. E., Mohamad, M. S., Deris, S., et al., 2013. Modelling gene networks by a dynamic bayesian network-based model with time lag estimation. In *Revised Selected Papers of PAKDD 2013 International Workshops on Trends and Applications in Knowledge Discovery and Data Mining - Volume 7867*, pp. 214–222. Springer-Verlag New York, Inc., New York, NY, USA.

Cosgrove, E. J., Zhou, Y., Gardner, T. S., et al., 2008. Predicting gene targets of perturbations via network-based filtering of mrna expression compendia. *Bioinformatics (Oxford, England)*, 24(21):2482–90.

Crampin, E. J., 2006. System identification challenges from systems biology. In *System Identification*, vol. 14 of *1*, pp. 81–93.

de Jong, H., 2002. Modeling and simulation of genetic regulatory systems: a literature review. *Journal of computational biology : a journal of computational molecular cell biology*, 9(1):67–103.

de Matos Simoes, R. and Emmert-Streib, F., 2012. Bagging statistical network inference from large-scale gene expression data. *PLoS ONE*, 7(3):e33624.

D'haeseleer, P., Wen, X., Fuhrman, S., et al., 1999. Linear modeling of mrna expression levels during cns development and injury. In *Pacific symposium on biocomputing*, vol. 4, pp. 41–52. World Scientific.

di Bernardo, D., Thompson, M. J., Gardner, T. S., et al., 2005. Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. *Nature biotechnology*, 23(3):377–383.

Di Camillo, B., Toffolo, G., and Cobelli, C., 2009. A gene network simulator to assess reverse engineering algorithms. *Annals of the New York Academy of Sciences*, 1158(1):125–142.

Djebbari, A. and Quackenbush, J., 2008. Seeded bayesian networks: Constructing genetic networks from microarray data. *BMC Systems Biology*, 2(1):57.

Elf, J., Li, G.-W., and Xie, X. S., 2007. Probing transcription factor dynamics at the single-molecule level in a living cell. *Science*, 316(5828):1191—1194.

Faith, J. J., Hayete, B., Thaden, J. T., et al., 2007. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biol*, 5(1):e8.

Fan, J., Feng, Y., and Wu, Y., 2009. Network exploration via the adaptive lasso and scad penalties. *The Annals of Applied Statistics*, 3(2):521–541.

Fan, J. and Li, R., 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Flores, M., Glusman, G., Brogaard, K., et al., 2013. P4 medicine: how systems medicine will transform the healthcare sector and society. *Per Med*, 10(6):565–576.

Folch-Fortuny, A., Villaverde, A., Ferrer, A., et al., 2015. Enabling network inference methods to handle missing data and outliers. *BMC Bioinformatics*, 16(1):283.

Friedman, J., Hastie, T., and Tibshirani, R., 2008. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics (Oxford, England)*, 9(3):432–441.

Friedman, J., Hastie, T., and Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1–22.

Gardner, T. S., Bernardo, D., Lorenz, D., et al., 2003. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(7):102–105.

Gardner, T. S. and Faith, J. J., 2005. Reverse-engineering transcription control networks. *Physics of Life Reviews*, 2(1):65 – 88.

Gevaert, O., Van Vooren, S., and De Moor, B., 2007. A framework for elucidating regulatory networks based on prior information and expression data. *Annals Of The New York Academy Of Sciences*, 1115(1):240–248.

Greenfield, A., Madar, A., Ostrer, H., et al., 2010. Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models. *PloS one*, 5(10):e13397.

Grimaldi, M., Visintainer, R., and Jurman, G., 2011. Regnann: Reverse engineering gene networks using artificial neural networks. *PLoS ONE*, 6(12):e28646.

Grzegorczyk, M. and Husmeier, D., 2013. Regularization of non-homogeneous dynamic bayesian networks with global information-coupling based on hierarchical bayesian models. *Machine Learning*, 91(1):105–154.

Gustafsson, M. and Hörnquist, M., 2010. Gene expression prediction by soft integration and the elastic net-best performance of the dream3 gene expression challenge. *PloS one*, 5(2):e9134.

Gygi, S. P., Rochon, Y., Franza, B. R., et al., 1999. Correlation between protein and mrna abundance in yeast. *Molecular and Cellular Biology*, 19(3):1720–1730.

Hache, H., Lehrach, H., and Herwig, R., 2009a. Reverse engineering of gene regulatory networks: A comparative study. *EURASIP Journal on Bioinformatics and Systems Biology*, 2009(1):617281.

Hache, H., Wierling, C., Lehrach, H., et al., 2007. Reconstruction and validation of gene regulatory networks with neural networks. pp. 319–324. Universität Stuttgart, Stuttgart.

Hache, H., Wierling, C., Lehrach, H., et al., 2009b. Genge: systematic generation of gene regulatory networks. *Bioinformatics*, 25(9):1205–1207.

Hastie, T., Tibshirani, R., and Friedman, J., 2009. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edn.

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

Hirsch, M. W., Devaney, R. L., and Smale, S., 2004. Differential equations, dynamical systems, and an introduction to chaos. In M. W. Hirsch, R. L. Devaney, and S. Smale, eds., *Differential Equations, Dynamical Systems, and an Introduction to Chaos*, pp. i –. Academic Press, Boston, 2nd edn.

Hoops, S., Sahle, S., Gauges, R., et al., 2006. Copasi—a complex pathway simulator. *Bioinformatics*, 22(24):3067–3074.

Huynh-Thu, V. A., Irrthum, A., Wehenkel, L., et al., 2010. Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE*, 5(9):e12776.

Ibarra, R. U., Edwards, J. S., and Palsson, B. O., 2002. *Escherichia coli* k-12 undergoes adaptive evolution to achieve in silico predicted optimal growth. *Nature*, 420(6912):186–189.

Jamshidi, N. and Palsson, B. Ø., 2010. Mass action stoichiometric simulation models: Incorporating kinetics and regulation into stoichiometric models. *Biophysical Journal*, 98(2):175 – 185.

Jia, J. and Rohe, K., 2012. Preconditioning to comply with the irrepresentable condition. *arXiv preprint arXiv:1208.5584*.

Jörnsten, R., Abenius, T., Kling, T., et al., 2011. Network modeling of the transcriptional effects of copy number aberrations in glioblastoma. *Molecular systems biology*, 7(486):486.

Julius, a., Zavlanos, M., Boyd, S., et al., 2009. Genetic network identification using convex programming. *IET systems biology*, 3(3):155–166.

Kabir, M., Noman, N., and Iba, H., 2010. Reverse engineering gene regulatory network from microarray data using linear time-variant model. *BMC Bioinformatics*, 11(Suppl 1):S56.

Khalil, H. K. and Grizzle, J., 1996. *Nonlinear systems*, vol. 3. Prentice hall New Jersey.

Kimura, S., Matsumura, K., and Okada-Hatakeyama, M., 2012. Inference of s-system models of genetic networks by solving linear programming problems and sets of linear algebraic equations. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pp. 1–8.

Kremling, A. and Saez-Rodriguez, J., 2007. Systems biology–an engineering perspective. *Journal of Biotechnology*, 129(2):329 – 351.

Küffner, R., Petri, T., Tavakkolkhah, P., et al., 2012. Inferring gene regulatory networks by anova. *Bioinformatics*, 28(10):1376–1382.

Küffner, R., Petri, T., Windhager, L., et al., 2010. Petri nets with fuzzy logic (pnfl): Reverse engineering and parametrization. *PLoS ONE*, 5(9):e12807.

Lauria, M., Iorio, F., and Di Bernardo, D., 2009. Nirest: A tool for gene network and mode of action inference. *Annals of the New York Academy of Sciences*, 1158(1):257–264.

Layek, R. K., Datta, A., and Dougherty, E. R., 2011. From biological pathways to regulatory networks. *Molecular Bio Systems*, 7(3):843–851.

Li, Z., Li, P., Krishnan, A., et al., 2011. Large-scale dynamic gene regulatory network inference combining differential equation models with local dynamic bayesian network analysis. *Bioinformatics*, 27(19):2686–2691.

Lorenz, D. R., Cantor, C. R., and Collins, J. J., 2009. A network biology approach to aging in yeast. *Proceedings of the National Academy of Sciences*, 106(4):1145–1150.

Lähdesmäki, H., Rust, A. G., and Shmulevich, I., 2008. Probabilistic inference of transcription factor binding from multiple data sources. *PLoS ONE*, 3(3):e1820.

Marbach, D., Costello, J. C., Küffner, R., et al., 2012. Wisdom of crowds for robust gene network inference. *Nature Methods*, 9(8):796–804.

Marbach, D., Prill, R. J., Schaffter, T., et al., 2010. Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences of the United States of America*, 107(14):6286–6291.

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

MATLAB, 2014. 8.3.0.532 (r2014a).

Menéndez, P., Kourmpetis, Y. a. I., ter Braak, C. J. F., et al., 2010. Gene regulatory networks from multifactorial perturbations using graphical lasso: application to the dream4 challenge. *PloS one*, 5(12):e14147.

Miller, A. K., Marsh, J., Reeve, A., et al., 2010. An overview of the CellML API and its implementation. *BMC Bioinformatics*, 11(1):1–12.

Milo, R., Shen-Orr, S., Itzkovitz, S., et al., 2002. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827.

Mordelet, F. and Vert, J.-P., 2008. Sirene: supervised inference of regulatory networks. *Bioinformatics*, 24(16):i76–i82.

Narendra, V., Lytkin, N., Aliferis, C., et al., 2011. A comprehensive assessment of methods for de-novo reverse-engineering of genome-scale regulatory networks. *Genomics*, 97(1):7–18.

Nelander, S., Wang, W., Nilsson, B., et al., 2008. Models from experiments: combinatorial drug perturbations of cancer cells. *Molecular Systems Biology*, 4(1):1–11.

Ng, A. Y., 2004. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pp. 78–. ACM, New York, NY, USA.

Nordling, T. E. M., 2013. *Robust inference of gene regulatory networks*. Ph.D. thesis, KTH School of Electrical Engineering, Automatic Control Lab.

Nordling, T. E. M. and Jacobsen, E. W., 2009. Interampatteness - a generic property of biochemical networks. *IET systems biology*, 3(5):388–403.

Nordling, T. E. M. and Jacobsen, E. W., 2011. On sparsity as a criterion in reconstructing biochemical networks. In *Accepted for publication at the 18th International Federation of Automatic Control (IFAC) World Congress, 2011.*

Oates, C. J., Hennessy, B. T., Lu, Y., et al., 2012. Network inference using steady-state data and goldbeter-koshland kinetics. *Bioinformatics (Oxford, England)*, 28(18):2342–2348.

Penfold, C. A., Buchanan-Wollaston, V., Denby, K. J., et al., 2012. Nonparametric bayesian inference for perturbed and orthologous gene regulatory networks. *Bioinformatics*, 28(12):i233–i241.

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

Pinna, A., Soranzo, N., and de la Fuente, A., 2010. From knockouts to networks: establishing direct cause-effect relationships through graph analysis. *PloS one*, 5(10):e12912.

Salgado, H., Peralta-Gil, M., Gama-Castro, S., et al., 2013. RegulonDB v8.0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more. *Nucleic Acids Research*, 41(D1):D203–D213.

Schaffter, T., Marbach, D., and Floreano, D., 2011. GeneNetWeaver: In silico benchmark generation and performance profiling of network inference methods. *Bioinformatics (Oxford, England)*, pp. btr373–.

Schmidt, H. and Jirstrand, M., 2006. Systems biology toolbox for matlab: a computational platform for research in systems biology. *Bioinformatics*, 22(4):514–515.

Schmitt, T., Ogris, C., and Sonnhammer, E. L. L., 2014. FunCoup 3.0: database of genome-wide functional coupling networks. *Nucleic acids research*, 42(Database issue):D380–D388.

Schreiber, S. L., 2000. Target-oriented and diversity-oriented organic synthesis in drug discovery. *Science*, 287(5460):1964–1969.

Schulz, M. H., Devanny, W. E., Gitter, A., et al., 2012. Drem 2.0: Improved reconstruction of dynamic regulatory networks from time-series expression data. *BMC Systems Biology*, 6:104.

Shih, Y.-K. and Parthasarathy, S., 2012. A single source k-shortest paths algorithm to infer regulatory pathways in a gene network. *Bioinformatics (Oxford, England)*, 28(12):i49–i58.

Stoica, P. and Selen, Y., 2004. Model-order selection: a review of information criterion rules. *Signal Processing Magazine, IEEE*, 21(4):36–47.

Sun, J., Tuncay, K., Haidar, A., et al., 2007. Transcriptional regulatory network discovery via multiple method integration: application to e. coli k12. *Algorithms for Molecular Biology*, 2(1):2.

Szklarczyk, D., Franceschini, A., Kuhn, M., et al., 2011. The string database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic acids research*, 39(Database issue):D561–D568.

Tegnér, J., Yeung, M. K. S., Hasty, J., et al., 2003. Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *Proceedings of the National Academy of Sciences*, 100(10):5944–5949.

Teixeira, M. C., Monteiro, P. T., Guerreiro, J. F., et al., 2013. The yeastract database: an upgraded information system for the analysis of gene and genomic transcription regulation in saccharomyces cerevisiae. *Nucleic Acids Research*.

Thorsson, V. and Michael, H., 2005. Reverse engineering galactose regulation in yeast through model selection reverse engineering galactose regulation in yeast through model selection. *Statistical Applications in Genetics and Molecular Biology*, 4(1).

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B Methodological*, 58(1):267–288.

Van den Bulcke, T., Van Leemput, K., Naudts, B., et al., 2006. Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, 7(1):1–12.

Villaverde, A. F., Henriques, D., Smallbone, K., et al., 2015. Biopredyn-bench: a suite of benchmark problems for dynamic modelling in systems biology. *BMC Systems Biology*, 9(1):8.

Wang, G., Du, C., Chen, H., et al., 2010. Process-based network decomposition reveals backbone motif structure. *Proceedings of the National Academy of Sciences*, 107(23):10478–10483.

Wang, L., Wang, X., Samoilov, M. S., et al., 2012. Inference of gene regulatory networks from knockout fitness data. *Bioinformatics*, pp. bts634–.

Ward, C., Yeung, E., Brown, T., et al., 2009. A comparison of network reconstruction methods for chemical reaction networks. In *Proceedings of the Foundations for Systems Biology and Engineering Conference*, pp. 197–200.

Weinberg, R. A., 1996. How cancer arises. *Scientific American*, 275(3):62–71.

Wolkenhauer, O., Ullah, M., Wellstead, P., et al., 2005. The dynamic systems approach to control and regulation of intracellular networks. *FEBS letters*, 579(8):1846–1853.

Wu, T. T. and Lange, K., 2008. Coordinate descent algorithms for lasso penalized regression. *Annals of Applied Statistics*, 2(1):224–244.

Wu, X., Li, P., Wang, N., et al., 2011. State space model with hidden variables for reconstruction of gene regulatory networks. *BMC Systems Biology*, 5 Suppl 3:S3.

Xiong, J. and Zhou, T., 2012. Gene regulatory network inference from multifactorial perturbation data using both regression and correlation analyses. *PloS one*, 7(9):e43819.

Yaghoobi, H., Haghipour, S., Hamzeiy, H., et al., 2012. A review of modeling techniques for genetic regulatory networks. *Journal of medical signals and sensors*, 2(1):61—70.

Yi, T.-M., Huang, Y., Simon, M. I., et al., 2000. Robust perfect adaptation in bacterial chemotaxis through integral feedback control. *Proceedings of the National Academy of Sciences*, 97(9):4649–4653.

Yip, K. Y., Alexander, R. P., Yan, K.-K., et al., 2010. Improved reconstruction of in silico gene regulatory networks by integrating knockout and perturbation data. *PLoS ONE*, 5(1):9.

Yu, J., Smith, V. A., Wang, P. P., et al., 2004. Advances to bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, 20(18):3594–3603.

Yuan, Y., Stan, G.-B., Warnick, S., et al., 2011. Robust dynamical network structure reconstruction. *Automatica*, 47(6):1230–1235.

Zagaris, A., Kaper, H. G., and Kaper, T. J., 2003. Analysis of the computational singular perturbation reduction method for chemical kinetics. *Nonlinear Science*, pp. 59–91.

Zavlanos, M. M., Julius, a. A., Boyd, S. P., et al., 2011. Inferring stable genetic networks from steady-state data. *Automatica*, 47(6):1113–1122.

Zhang, X., Liu, K., Liu, Z.-P., et al., 2012. Narromi: a noise and redundancy reduction technique improves accuracy of gene regulatory network inference. *Bioinformatics (Oxford, England)*, pp. 1–8.

Zhao, P. and Yu, B., 2006. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563.

Zou, H., 2006. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Zou, H. and Hastie, T., 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

Äijö, T. and Lähdesmäki, H., 2009. Learning gene regulatory networks from gene expression measurements using non-parametric molecular kinetics. *Bioinformatics*, 25(22):2937–2944.

Acknowledgements

I am not really the person that spreads his net wide when it comes to social interactions. Even so, I find myself struggling to fit all the people in these few pages that I owe my thanks to for supporting me on this journey.

First I would like to thank my family, we are truly a big family and while I don't meet most of you often enough you are always in my thoughts. I especially would like to thank my mother for always fighting to provide me with the best, and for being a rock where I could always find support.

I would also like to take this opportunity to thank my past self for not giving up on me. I will keep in touch, just out of reach.

Mari, I could not have done this without you. Your constant support and positive attitude made this process much simpler than it would have been without you; Jimmy and Britt-Mari, thank you for all the good times during my holidays.

Pierre, our history require no words. We never did put Abstras in his final resting place though…

David and family, for all the laughs and for always treating me like a member of the family.

Patrik, I'm not sure I would be where I am today if it wasn't for you introducing me to programming and all things computer related.

Julien and Emma, crazy, smart, deep. No ideas are to wild to be examined in your presence. My time back in good old Göteborg have been much more exciting thanks to you. I look forward to every time we can sit down over a fika and chat for hours.

Stockholms Marie, Göte and Evy, who welcomed me with open arms when I needed a place to live when arriving in Stockholm.

I would also like to give *cred* to the members of the Sonnhammer group, both past and present – Christoph for always being a friend. I will always be ready for another Zombie raid! Lizzy make sure to be nice to

Christoph when it's his turn to graduate; Dimitri for your interesting conversations, and for always inviting me when there was an event where we could share our common nerd interests; Mateusz, I'm glad that you are on your path to becoming a true Emacs guru. At least I got to convert one person!; Daniel, don't doubt yourself, just rewrite the code!; Stefanie, for our little talks. We have a saying in Sweden "Simma lugnt"; Gabe, for some reason I always felt that you where the only other swede in the group, even though your "northern" Swedish words was many times nothing I could understand; Matt, thanks for removing all those "asses";
To all of you, I hope that we can still find the time to continue with our researchathons, our retro-gaming evenings and all the other goodness that we have enjoyed together.

I would also like to thank my co-supervisor Torbjörn for introducing me to the world of signal processing, but foremost I would like to thank you for always providing support and insight to life as a person and as a scientist, and for being a person that I can look up to.

Finally, I would like to thank my supervisor Erik Sonnhammer for giving me this opportunity and for guiding me through my education, for having patience with all my missteps, and for giving me the freedom to explore my own interests.

Sammanfattning

Systembiologi studerar biologi ur ett systemperspektiv. Detta innebär att förutom att undersöka detaljer, så som en specifik gen, en cell eller en individs beteende, i biologiska system undersöker man hur de här komponenterna interagerar med varandra i ett större sammanhang. När det gäller individer i en population försöker vi skapa oss en bild av hur populationen fungerar utifrån hur individerna interagerar i populationen och inte utifrån individens specifika egenskaper fungerar när denne är isolerad. Studerar vi gener tittar vi på hur uttrycket av generna, i kombination med uttrycket av andra gener, reglarar hur cellen fungerar och vilka funktioner som cellen kan utföra och använda sig av beroende på hur generana påverkar varandra.

Målet är att kunna beskriva hur systemet, i det här fallet cellen, fungerar och att kunna förutsäga vad som kommer hända när olika förändringar sker i och runt cellen. Det klassiska tillvägagångsättet som använts för att vetenskapligt studera fenomen, att bryta ner ett system i dess beståndsdelar och studera dem ingående för att sedan kunna dra slutsatser om hela systemet har visat sig svårt när man studerar komplexa levande system. Egenskaper hos systemet som är svåra att förutsäga genom att studera enskilda komponenters egenskaper, har kunnat förklaras när man placerat komponenterna i en specifik kontext. För gener är denna kontext den biologiska cellen.

Den mänskliga cellen har omkring 20 000 proteinkodande gener. Antalet möjliga interaktioner mellan dem är 400 000 000. Inkluderar vi molekyler som kommer utifrån cellen och de olika steg som en gen kan uttryckas i och därför regleras på, ökar snabbt detta tal i storlek. Att studera alla komponenter i detalj är inte enkelt och att sedan kunna ge en informativ bild av hur systemet fungerar när komponenterna interagerar med varandra är än mer svårforcerat.

Verktygen och teorierna från systembiologin är framtagna och utvecklade för att kunna generera kunskap under de här förutsättningarna. Både att kunna rekonstruera och skapa modeller av interaktionerna, så kallade genreglernätverk, och att kunna dra hållbara slutsatser samt få en mekanistisk förståelse om det underliggande biologiska systemet, fall-

er inom systembiologins forskningsfält.

För att kunna skapa genreglernätverk krävs experiment designade för att kunna generera modeller av genreglernätverk. Vi måste kunna mäta koncentrationen av flera specifika biologiska molekyler, såsom mRNA och protein, samtidig och med hög precision. Beroende på vad vi mäter får vi olika sätt att se hur mycket en gen är uttryckt. För att kunna generera nätverk där vi vet vilken gen som påverkar vilken annan gen krävs att vi utför specifika störningar på cellen som cellen måste svara på. Hur cellen svarar är en konsekvens av hur cellens reglernätverk är uppbyggt. Att skapa sådan data är inte trivialt och medför flera problem när målet är att kunna rekonstruera ett nätverk som kan beskriva de underliggande funktionerna och interaktionerna som cellen utför. Data av den här typen är ofta brusigt vilket gör att de förändringar som cellen gör i svar på våra störningar inte går att observera helt tydligt. Systemets uppbyggnad och brus är exempel på egenskaper som datan och även systemet vi tittar på har. Det vi vill veta är till exempel hur mycket brus och mätfel vi ska ta hänsyn till och hur cellens genreglernätverk påverkar våra mätningar. Den här typen av data är den typ som behövs för att kunna konstruera modeller för genreglernätverk i celler.

Arbetet som är underlag för denna avhandling fokuserar på de begränsningar som egenskaperna hos data och hos systemet vi observerar, ger när vi försöker ta fram en modell av cellens genreglernätverk. Dessutom tittar vi på hur egenskaperna hos datan påverkar hur bra vi kan förvänta oss att olika metoder kommer vara på att återskapa interaktioner mellan generna.

Vi tittar också på vad vi måste ta hänsyn till för att olika metoder ska prestera så bra som möjligt och vilka egenskaper som avgör detta. För att kunna avgöra hur egenskaperna påverkar prestandan hos metoderna som används, behöver vi också veta vilka egenskaper som är viktiga att studera. Detta är det andra viktiga bidraget som behandlas i underlaget till denna avhandling. Vi visar att specifika egenskaper som vi kan mäta kan leda till slutsatser om hur vi ska analysera den tillgängliga datan.

Det tredje bidraget försöker svara på frågan om det går att inkludera information och evidens som framtagits med annan data för att öka effektiviteten hos metoderna som används. Här demonstrerar vi att för data av låg kvallite kan inkludering av evidens av annat slag öka effektiviteten hos metoderna.

Slutligen delar vi med oss av de lösningar som tagits fram under tiden vi studerat och löst olika problem genom ett mjukvarupaket i förhoppningen att det kan vara användbart för andra och visar på specifika analysmetoder som borde ingå när man försöker modellera gennätverk.

Glossary

in silico From latin meaning “in silicon”, meaning run on a computer. 25, 27, 35, 37, 39

in vivo From latin meaning “in the living”, meaning experimentation done on living systems. 25–27

χ^2 the chi-square distribution. 22, 24, 36

DREAM Dialogue on Reverse Engineering Assessment and Methods. 25

condition number describes how ill-conditioned a system is.. 15, 38, 39

singular value represent the variation along the principal vectors.. 15, 23

steady state The state of the system where it does not experience any changing in time. Where the state variables $\dot{\boldsymbol{x}} = 0$.. 10, 12–15, 18–20, 22, 26

steady state data Data collected when a system is in steady state. Typically after an a change have been induced and the system has converged.. 19, 20, 25, 41, *see* steady state

time series data Data collected at different time points during the evolution of a dynamic system.. 16, 20, 25, 26, 41