# Inferring Biological Networks from Time-Course Observations Using a Non-linear Vector Autoregressive Model

Afshin Mashadi-Hossein

A thesis
submitted in partial fulfillment
of the requirements for the degree of
Master of Science

Department of Biostatistics
University of Washington
2015

Reading Committee:
Dr. Ali Shojaie, Chair
Dr. Noah Simon

# Contents

# Chapter 1

# Seeing Biology through Networks

With the advances in our ability to collect molecular information at a cellular level, biomedical research is increasingly focused on defining diseases and tailoring treatments based on patients' molecular profile. Central to this effort is the understanding of the molecular interactions that regulate cellular activities. Early research viewed these molecular interactions to occur in a serial fashion, where biological signals were believed to sequentially travel from one molecular component to the next [39]. However, cellular functions are increasingly viewed to be regulated through networks of molecules working in parallel [65]. This network-based perspective on cellular regulations has important ramifications for how diseases are characterized and treated at the molecular level. For example, a disease could be defined as the abnormalities in interactions of a molecular network, even when no single molecular culprit to the disease can be identified [62]. The network view of diseases can also help guide the development of more effective treatments based on network structures as opposed to individual biomarkers [86].

Transitioning to a network-based approach in diagnosis and treatment of diseases, however, requires advances in our capabilities to observe (the components of) networks and to characterize their dynamics. To this end, new experimental and computational tools need to be developed. Experimentally, the tools of molecular biology are rapidly growing in capacity. Multiomic platforms, currently in development [54], are expected to reduce the complexities involved in observing the components of biological networks. Meanwhile, collaborative initiatives such as The Cancer Genome Atlas (TCGA) [35] and the Encode project [9] have substantially increased our collective capacity to study and record diseases at the molecular level.

Computationally, while the tools for studying networks have been rapidly advancing, it is generally observed that the rate of data generation has outpaced the computational capabilities to interpret them [39]. This adds urgency to the need for new computational methods that help bridge the gap between the growing body of 'omics'data and the much needed information on the role of networks in human diseases. This need motivated the work presented in the next two chapters.

This chapter is focused on providing a broad overview of how biological networks are observed experimentally and recovered computationally. The remainder of the chapter is organized as follows: In Section 1.1 a brief review of some of the experimental approaches used in the study of biological networks is provided. Section 1.2 will discuss methods developed for inferring the topology of networks. Specifically, Section 1.2 discusses the early methods developed for understanding patterns of

co-regulations in biological networks, as well as the more recent developments in application of Bayesian networks and regularized estimators for inferring networks of conditional independence. The section concludes with a brief review of some of the approaches employed for establishing causality in inferred networks.

## 1.1 Collecting Data on Biological Networks

Understanding the capabilities and the limitations of the experimental methods for collecting data on biological networks can help establish a context in which to evaluate computational methods used in network inference. Experimentally, collecting data that could inform reliable reconstruction of a molecular network is complex and costly. The interactions within biological networks span a diverse range of molecule types (e.g. protein-protein, protein-DNA, and protein-RNA interactions as well as interaction of each of these molecule types with various metabolites [20]). As no single method in molecular biology can quantify all these interaction types, collecting data on a network can require application of a diverse range of experimental methods. These complexities in observing molecular interactions combined with the large number of possible interactions highlight the need for efficient computational methods suitable for use in high-dimensional [1] settings.

Furthermore, to establish causal dependencies, experiments need to go beyond collecting passive observations of network components. This might be done by altering the state of network components and observing the propagation of the perturbation effects through the network. In studying biological networks, these perturbations can be in the form of eliminating a gene (knock out), reducing the expression of a gene (knock down) or use of chemicals that stimulate the network or alter interactions of network components. Alternatively, genetic variations across populations can be used as a naturally occurring perturbation to the population expression levels [63].

A knock out (KO) perturbation permanently alters the organism by effectively eliminating a gene from its genome [27]. Generating a KO organism ensures that the targeted gene is completely excluded from the network. However, KO organisms are costly to produce, and for viability of the organism, only non-essential genes can be targeted. In contrast, a knock down (KD) perturbation uses small interfering RNAs (siRNA), which, as the names implies, interfere with the expression of their targeted gene often drastically reducing their expression levels [51]. This method is efficient, and as it does not completely eliminate the expression of a targeted gene, can be used to directly perturb even the essential genes. Both knock out and knock down perturbations are common in studying biological networks. To efficiently capture inter-dependencies of the network components, often networks are observed under multiple perturbations applied simultaneously according to multifactorial experimental designs.

Collecting time point observations following perturbations can help characterize the network dynamics. Such experiments are considerably more challenging to conduct, but when such complexities are justified/manageable, time-course observations can provide inference methods with data based on which temporal relations

[1] High dimensionality is due to the small number of observations, compared to the large number of possible interactions.

can be inferred. Temporal relations, although do not prove causality, can provide supporting evidence for causal relations. In contrast, when time-independent interdependencies of network components are of primary interest, experiments might be designed to capture the steady state of the network ignoring any transient variability in the network.

## 1.2 Inferring the Topology of Biological Networks

The topology of a network formally defines the 'relation' among the elements that comprise the network. For example, the topology of a gene regulatory network (GRN) defines the relation between a set of genes, proteins, and metabolites involved in regulating the transcription of a gene. In this context, the exact definition of 'relation' specified by a given network topology could vary based on the method used for inferring the topology. For example, association, direct association and causal dependence can each be used as the 'relation' specified by a network topology. As we will discuss in the remainder of this chapter, depending on the type of relation defined by a network, the task of inferring a network topology (hereafter referred to as network inference) could vary substantially in complexity. To better understand this task, it is helpful to formalize the representation of a network topology.

Graphs are often used to represent a network topology, where each element of a network is referred to as a node (or vertex) and the relationship between nodes are represented by edges. When edges are directional, they typically represent causal [2] dependence between two nodes. In contrast, undirected edges suggest conditional dependence (or in some contexts only pairwise associations) between the elements of the network.

Equivalently, the structure of a network can be represented by an adjacency matrix. For a causal network with $p$ elements, the corresponding adjacency matrix is a $p \times p$ matrix of zeros and ones [3], where the $i^{\text{th}}$ row and $j^{\text{th}}$ column being 1(0) represents the $j^{\text{th}}$ node being causally dependent on (independent from) the $i^{\text{th}}$ node. Figure 1.1 shows a directed graph, as well as the adjacency matrix representation of the structure of a network with 4 elements.

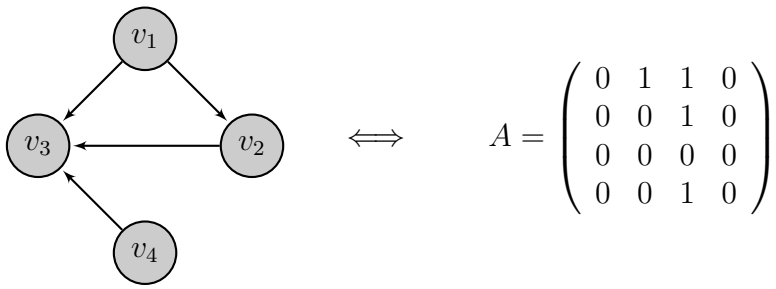



Figure 1.1: A simple directed graph with 4 nodes and 4 edges and the corresponding adjacency matrix A

This formal representation of networks helps us define network inference as estimation of the elements of an adjacency matrix from observations made on the

[2] In some contexts, directional edges could represent temporal relations.

[3] In some settings, instead of the binary matrix, the elements of an adjacency matrix could be real values quantifying the nature of interactions, time lags, etc.

state of the network components. This immediately reveals one of the characteristic challenges facing any network inference method: As the number of parameters to be estimated grows quadratically relative to the size of the network and given the experimental costs discussed in Section 1.1, network inference problems are often underdetermined.

### 1.2.1 Establishing Patterns of Co-regulations

Not to be hampered by the problem of being underdetermined, some of the early work in this area restricted their aim to evaluation of pairwise similarities in the profile of network components. For example, Langfelder *et al.* used exponentiated absolute value of Spearman correlation to build a network of co-expressions [43]. Likewise, Balasubramaniyan *et al.* used Spearman rank correlation as a measure of similarity between the expression of two genes observed over time [2]. Given that their data contained time-lag information, they then hypothesized causal effect links between genes with similar expression profiles.

Besides correlations, other similarity metrics have been used to capture pair-wise associations between network components. In developing RELNET (RELevance NETworks), Butte and Kohane used mutual information (MI) between probability distribution of discretized expression values for pairs of genes. Given the discrete expression vectors X and Y, MI could be computed as in Eq. (1.1)

$$\mathrm{MI}(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left( \frac{p(x,y)}{p(x)p(y)} \right) \tag{1.1}$$

Here, $p(x,y)$ is the joint (discrete) probability of $X$ and $Y$ while $p(x)$ and $p(y)$ are the corresponding marginal (discrete) probabilities. Following computation of MI for all pairs, the authors used a permutation test to set the threshold for values of MIs considered significant. They "hypothesize[d] that the higher mutual information is between two genes, the more likely it is they have biological relationship" [6].

Other methods have expanded on RELNET algorithm. One such method is ARACNE (algorithm for reconstruction of accurate cellular network) [47]. Relative to RELNET, ARACNE uses Data Processing Inequality (DPI) to avoid similarities between the profiles of two network components being inferred as a direct association when that association can be explained by an intermediate component. Following the computation of MI for all pairs, ARACNE evaluates all triplet nodes X, Y and Z and excludes any edge between X and Z if $I(X,Z) \leq \min\left(I(X,Y), I(Y,Z)\right)$. In addition to RELNET and ARACNE, a number of other algorithms such as CLR (Context Likelihood of Relatedness) [21] use MI and/or correlation scores to infer network topology.

Methods using pairwise similarity scores for network inference are intuitive and computationally inexpensive. Additionally, despite their simplicity, some have shown to perform fairly well relative to some of the more sophisticated approaches. In a recent review article by Maetschke *et al.* [46], the aforementioned methods as well as a number of other more sophisticated methods were evaluated on simulated datasets. Among the unsupervised methods, null mutant Z-score [61] which also operates based on a simple pairwise scoring scheme, succeeded to outperform all other methods. In the same analysis, correlation based methods performed surprisingly well.

### 1.2.2 Establishing Conditional Independence

Despite their advantages, network inference based on pairwise scoring schemes are inadequate for identifying conditional independence among elements of a network. Employing strategies such as the DPI filter in ARACNE is aimed at mitigating this shortcoming. However, generally the networks inferred form pairwise scoring schemes are interpreted as showing patterns of co-regulations among the elements of the true network [49].

Relative to the patterns of co-regulations, it is much more informative for an inferred network to elucidate direct dependencies between the elements of the true (unknown) network. This however would require simultaneous (as opposed to pairwise) estimation of all inter-dependencies such that any dependency between two elements is estimated after adjustment for all other effects which could explain the variations in the state of those elements. Formally, this would amount to establishing conditional independence between the elements of the network. Specifically, given the set of all nodes V, we would like the connection between nodes X and Y to imply [49]

$$X \not\perp\!\!\!\perp Y | Z \ \forall Z \subseteq V \setminus \{X, Y\} \tag{1.2}$$

Many methods have been developed for inferring networks that encode conditional independence relationships. A considerable proportion of these methods follow the framework Bayesian networks (BN). Regularized graphical models constitute another growing family of methods for inferring networks of conditional independence.

#### Bayesian Networks

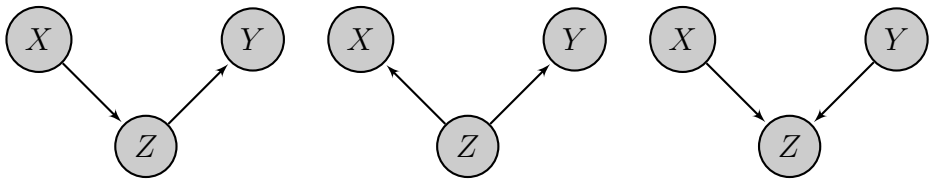
"Directed graphs $[\cdots]$ used to represent causal or temporal relationships $[\cdots]$ [have come] to be known as Bayesian networks" [59, p. 14]. Probabilistically, the structure of a Bayesian network defines a unique set of conditional probability distributions (CPD). In network inference, where this structure is unknown, data can be used to infer CPDs which can in turn be used to define the graphs [4] that are compatible with the inferred CPDs.

Bayesian network inference methods can be broadly categorized into constraint-based or score-based algorithms [68]. Constraint-based algorithms infer network structure through an elimination process. Starting from a fully connected graph, these methods eliminate edges connecting nodes that are deemed (conditionally) independent based on some hypothesis test of independence. In contrast, score based methods search the space of candidate graphs, score each candidate and select the graph(s) with the highest score.

Compared with the score-based algorithms, constraint based methods, such as the PC algorithm [73], generally offer better computational efficiency. However, the performance of the constraint-based methods are often too sensitive to the failure of hypothesis tests in determining true conditional independence. For this reason, score-based algorithms have come to be favored in computational biology [49] and other applied fields [44].

[4] It is noteworthy that while a BN specifies unique CPDs, a set of CPDs can be compatible with more than one BN. Therefore, inferring BNs from observational data often leads to a group of equivalent graphs rather than a single graph.

Learning the structure of BNs using a score-based algorithm can be generally broken down into two sub-tasks: 1) applying a scoring scheme that allows comparison of candidate graphs, and 2) searching the space of possible graphs for good graph candidates to score. To understand how the first task (scoring graphs) is approached, it is helpful to view edges of a graph as pathways for flow of information (or influence) between the nodes. This intuitive view can then be used to identify conditional (in)dependencies among the elements of the graph. Knowing conditional dependencies, in turn enables decomposition of the graph score to an aggregate of its sub-component scores, thereby making graph scoring manageable. But to define how the notion of information flow relates to conditional (in)dependencies we need to review the concept of d-separation.




(a) Chain (b) Outflow (c) Collider

Figure 1.2: d-separation: in 1.2a and 1.2b Z is in the *active* path of information whereas in 1.2c information flow is blocked in Z and therefore Z does not d-separate X and Y in 1.2c

The concept of d-separation due to Pearl [82] may be understood by the simple graphs in Figure 1.2. First, we define a path to be 'active' when there is no 'collision' of information flow in any of the elements of that path. In Figure 1.2c, information flow collides in Z. Therefore, the path through Z is not in an active path and Z does not d-separate X and Y. The utility of recognizing d-separation is in its relation to conditional independence. For sets of nodes, X, Y and Z, when Z d-separates X and Y, X and Y are independent conditional on Z. When it does not, X and Y may (or may not) be dependent conditional on Z. For example, in Figures 1.2a and 1.2b $X \perp\!\!\!\perp Y|Z$ whereas for 1.2c that may or may not be true. Using d-separation and the conditional dependency it establishes, for Figure 1.2a we can express the joint probability (score) of X,Y and Z as $p(X, Y, Z) = p(Y|Z)p(Z|X)p(X)$ which is entirely composed of marginal and conditional distributions (as opposed to joint distributions). Similarly for Figure 1.2b, $p(X, Y, Z) = p(X, Y|Z)p(Z) = p(Y|Z)p(X|Z)p(Z)$. Lastly, for Figure 1.2c the joint probability can be expressed as $p(X, Y, Z) = p(Z|X, Y)p(X, Y) = p(Z|X, Y)p(X)p(Y)$. This type of factorization allows BNs to define conditional probability relations which facilitate scoring each candidate graph in a manageable recursive fashion based on the observed data.

One natural option for scoring a candidate graph is the maximum likelihood of the graph. To allow for such scoring scheme, one could make parametric assumptions about the probability distribution of the data. Following such assumptions, the task of network scoring is converted into finding the maximum likelihood estimate (MLE) of the parameters of the assumed distribution given the observed data and subsequently using MLE to estimate the maximum likelihood (ML) score. When nodes are measured as continuous random variables, Gaussian distribution is a common choice for modeling the data. In the case of discrete data, Multinomial model of likelihood is often used [23].

While intuitive, reliance on ML scores for network inference results in overfitting (i.e. edges are inferred between nodes that are in reality conditionally independent). This behavior is not unexpected when considering the relationship between likelihood scores and the mutual-information-based techniques discussed earlier. This relationship can be shown using the following simple but illustrative example from [42]. Imagine we have observed data on two boolean nodes (i.e. nodes that can take values in $\{0, 1\}$) X and Y. Consider the two structures in Figures 1.3a and 1.3b as the two candidate graphs being compared.


(a) $\mathcal{G}_0$ (b) $\mathcal{G}_1$

Figure 1.3: Given to nodes X and Y, graph 1.3b almost always has a higher MI score than graph 1.3a

When the difference in ML scores is used to decide between graphs $\mathcal{G}_0$ and $\mathcal{G}_1$, the connected graph, $\mathcal{G}_1$, almost always has a higher score. This is because, as shown in Eq. (1.3), the difference in scores amounts to $n \times \text{MI}(X, Y)$, where $n$ is the number of observations. While, theoretically $0 \leq \text{MI}$, in practice MI is almost always strictly positive, which leads to the connected nodes being always favored over the disjoint nodes regardless of the data. This is the reason why MI-based methods discussed earlier had to rely on permutation-based null.

$$
\begin{aligned}
&Score(\mathcal{G}_1|Data) - Score(\mathcal{G}_0|Data) \\
&= \sum_{i=1}^{n} \log \hat{\theta}_{X=x_i} + \log \hat{\theta}_{Y=y_i|X=x_i} - \sum_{i=1}^{n} \log \hat{\theta}_{X=x_i} + \log \hat{\theta}_{Y=y_i} \\
&= n \left( \sum_{(x,y)\in Data} \hat{p}(x, y) \log \hat{p}(y|x) - \sum_{y\in Data} \hat{p}(y) \log \hat{p}(y) \right) \\
&= n \left( \sum_{(x,y)\in Data} \hat{p}(x, y) \log \hat{p}(y|x) - \sum_{(x,y)\in Data} \hat{p}(x, y) \log \hat{p}(y) \right) \\
&= n \left( \sum_{(x,y)\in Data} \hat{p}(x, y) \log \frac{\hat{p}(y, x)}{\hat{p}(x)\hat{p}(y)} \right) \\
&= n \ \times \ \text{MI}(X, Y)
\end{aligned}
\tag{1.3}
$$

where $\hat{\theta}_{Z=z_i} \equiv p_{\text{MLE}}(Z = z_i) = \frac{\sum_{i=1}^{n} \mathbf{1}_{[Z=z_i]}}{n} \equiv \hat{p}(Z = z_i)$

In the context of BNs, proper use of likelihood-based scores requires controlling the complexity of graphs. A possible mitigation is to enforce low complexity by

placing scientifically sensible restrictions on implausible edges. A less restrictive approach is using Schwarz's Bayesian Information Criteria (SBIC) score [67] given in Eq. (1.4) for scoring candidate graphs. This latter approach seeks to balance maximizing the likelihood with maintaining a low model complexity.

$$\text{SBIC}(\mathcal{G}_g|\boldsymbol{D}) = \log\ L(\hat{\boldsymbol{\theta}}_g|\boldsymbol{D}, \mathcal{G}_g) - \frac{\|\hat{\boldsymbol{\theta}}_g\|_0}{2} \log\ n \tag{1.4}$$

Here, $n$ is the observation number, $\boldsymbol{\theta}_g$ is the vector of model parameters, $\hat{\boldsymbol{\theta}}_g$ is the maximum likelihood estimate of $\boldsymbol{\theta}_g$ for the data, and $L(\hat{\boldsymbol{\theta}}_g|\boldsymbol{D}, \mathcal{G}_g)$ is the maximum likelihood of the data, $\boldsymbol{D}$, under model $\mathcal{G}_g$.

In SBIC, model complexity is represented by $\|\boldsymbol{\theta}_g\|_0$, the number of non-zero parameters (edges). Therefore, for any edge to be added, the resultant gain in the likelihood score should outweigh the cost of addition of a new parameter. In this way, graph selection based on SBIC scores imposes a penalty on inferring edges to mitigate overfitting.

It is well-known that SBIC is asymptotically consistent. However, in practice (due to limited observations as well deviations from distributional assumptions) it is at best an approximation to the log likelihood of the data. A more rigorous approach to scoring candidate networks is to compute the probability of a candidate graph $\mathcal{G}$ given the data $\boldsymbol{D}$: $p(\mathcal{G}|\boldsymbol{D})$. This approach employs the Bayesian scheme in Eq. (1.5) to estimate the mentioned probability score.

$$p(\mathcal{G}|\boldsymbol{D}) = \frac{p(\mathcal{G})\, p(\boldsymbol{D}|\mathcal{G})}{p(\boldsymbol{D})} \propto p(\mathcal{G}) \int_{\Theta} p(\boldsymbol{D}|\mathcal{G}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{G}) d\Theta \tag{1.5}$$

where $\Theta$ is the support for all parameter estimates $\boldsymbol{\theta}$.

An uninformative (i.e. uniform) prior can be chosen for $p(\mathcal{G})$ when no prior information on the graph is desired to influence the inference [10]. Alternatively one might design a prior that penalizes the number of edges if overfitting is a concern. Evaluating the integral corresponding to $p(\boldsymbol{D}|\mathcal{G})$ is the most involved step for this scoring scheme. An assumption that simplifies this task is to use conjugate pairs of distributions for the conditional likelihood of the data, $p(\boldsymbol{D}|\mathcal{G}, \boldsymbol{\theta})$, and the conditional prior on the network parameters, $p(\boldsymbol{\theta}|\mathcal{G})$. For discrete data a Multinomial model with a Dirichlet prior on the parameters is commonly used [33]. When data is continuous, a Normal distribution and a Normal-Wishart prior for the parameters are common [26]. When conjugate pairs are assumed, computing the score of a graph benefits form closed form solutions for $p(\boldsymbol{D}|\mathcal{G})$. When such assumption however is not justified, approximation techniques (e.g. sampling or Laplace integration [40]) are the only options.

Once a scoring scheme is selected, the second task in network inference in the framework of BNs is finding candidate graphs for scoring. This is of course trivial when there are only a handful of elements in a network, as in that case, the method can score all the possible graphs and pick the graph(s) with the highest score. This however can not be done for larger networks as the number of possible graphs grow super-exponentially. Therefore, the main concern for the second task is how to most efficiently search the network space. This encompasses addressing how to perform

the search as well as how to reduce the computational cost of scoring each candidate graph (i.e. avoiding redundant computations).

For searching among candidate graphs many different heuristic search algorithms have been used [7, 10, 33, 79]. These methods are relatively simple and often fast. However, they can only identify locally optimal points and could be easily trapped in local plateau. Mitigations such as repeated search with different initializations, or maintaining a 'tabu' list of recently surveyed regions of the space [79] are often needed to achieve an acceptable performance. Other methods have used Markov Chain Monte Carlo (MCMC) sampling schemes [17, 45] for searching the graph space. MCMC samplers are also susceptible to being trapped by local optima. Additionally the 'step size' needs to be tuned to allow for the chain to traverse at an appropriate 'pace' to find and converge to the high-scoring graphs.

In the recent years, BNs have made significant contributions in establishing a framework for network inference in biological settings [36,64]. BNs are intuitive and when inferred through the Bayesian approach, they readily facilitate integration of existing biological knowledge into new models. Additional developments in this area such as Dynamic Bayesian Networks (DBNs) have shown promise in helping unravel the temporal dynamic of molecular interactions within biological networks. [34, 86, 95].

Despite these advances, many challenges remain with using BNs for network inference. From selecting appropriate priors to setting the proper search parameters, there are many factors that affect the performance of the BN methods. For example, setting the Markov chain to have low variability, not only slows the convergence rate, but also results in higher correlation between samples. Set the variability too high and it might never converge.

Even with ideal tuning parameters, the computational costs of inferring BNs can be prohibitively expensive. Dynamic programming has been proposed to reduce this time [41] at a substantial cost increase to memory usage. Despite that, estimation of large networks (i.e. networks with $> 100$ nodes) with BNs remains impractical [17, 58].

While advances in computing and statistics are expected to continue to expand the capabilities of BNs, research in network inference can greatly benefit form complimentary approaches. In a recent meta-analysis for network inference methods, Vignes and colleagues found regularized regression approaches to network inference to be complementary to those of BNs [84]. The regularized inference methods have been the subject of extensive studies in recent years and network inference is positioned to benefit from the growing body of knowledge in this area.

### Regularized Graphical Models

Regularized estimators have been extensively used to solve high-dimensional problems that are believed to have sparse solutions. By penalizing the complexity (i.e. the number of parameters) of the model, these estimators find unique spares solutions for otherwise underdetermined systems. L1-regularization (LASSO) [80] is commonly applied to estimate sparse solutions to underdetermined linear regression problems. Similarly, regularized graphical models have been proposed for network inference [22, 53, 88].

Regularized graphical models were first studied in the context of Gaussian Graphical Models (GGM) [53]. For these models, the task of establishing conditional de-

pendence relationships between the nodes amounts to estimation of the precision (or concentration) matrix of the joint distribution of the network elements. In the context of a GGM with a precision matrix $K = \Sigma^{-1}$, node $a_1$ and node $a_2$ are conditionally independent if and only if $K_{a_1a_2} = K_{a_2a_1} = 0$.

To see this, we need to evaluate the joint distribution of pair of nodes $a_1$ and $a_2$ conditional on all the other nodes. To do so, it is more convenient to represent the multivariate normal distribution in its 'information' form, $\mathcal{N}(h, K)$, defined in Eq.(1.6). Consider $X_{i.} \in \mathbb{R}^p$ to be the $i^{\text{th}}$ observation made on a p-node network. Additionally, assume that $X_{i.} \sim N(\mu, \Sigma)$. We can represent this distribution as in Eq. (1.6)

$$\begin{aligned} X_{i.} &\sim (2\pi)^{-p/2}|\Sigma|^{-1/2}e^{-\frac{1}{2}(X_{i.}-\mu)'\Sigma^{-1}(X_{i.}-\mu)} \\ &= (2\pi)^{-p/2}|\Sigma|^{-1/2}e^{-\frac{1}{2}X_{i.}'\Sigma^{-1}X_{i.}+X_{i.}'\Sigma^{-1}\mu-\frac{1}{2}\mu'\Sigma^{-1}\mu} \\ &= C \ * \ e^{-\frac{1}{2}X_{i.}'\boldsymbol{K}X_{i.}+X'h} \equiv \mathcal{N}(h, K) \end{aligned} \tag{1.6}$$

where $K = \Sigma^{-1}$, $h = \Sigma^{-1}\mu$ and $C$ is a constant

Given the information form of the joint distribution, we can easily derive the joint distribution of any pair of nodes $a_1$, and $a_2$ conditional on all the other nodes. To do so, we partition the nodes in $X_{i.}$ into $X_{i,A}$ and $X_{i,B}$, where $X_{i,A} = [X_{i,a_1}X_{i,a_2}]'$ and $X_{i,B}$ contains the observation on all the other nodes. Treating $X_{i,B}$ as constant, the conditional distribution can be represented as follows:

$$\begin{aligned} &p(X_{i,A}|X_{i,B}) \propto p(X_{i,A}, X_{i,B}) \\ &\propto \exp\Big( -\frac{1}{2}[X_{i,A}'X_{i,B}'] \begin{bmatrix} K_{AA} & K_{AB} \\ K_{BA} & K_{BB} \end{bmatrix} \begin{bmatrix} X_A \\ X_B \end{bmatrix} + [X_{i,A}'X_{i,B}'] \begin{bmatrix} h_A \\ h_B \end{bmatrix} \Big) \\ &\propto \exp\Big( -\frac{1}{2}X_{i,A}'K_{AA}X_{i,A} - X_{i,A}'K_{AB}X_{i,B} + X_{i,A}'h_A \Big) \\ &\propto \exp\Big( -\frac{1}{2}X_{i,A}'K_{AA}X_{i,A} + X_{i,A}'(h_A - K_{AB}X_{i,B}) \Big) \\ &\text{which has the form } \mathcal{N}(h - K_{AB}X_iB, K_{AA}) \end{aligned} \tag{1.7}$$

Given that $(X_{i,A}|X_{i,B}) \sim \mathcal{N}(h - K_{AB}X_iB, K_{AA})$, we have

$$\begin{aligned} \text{cov}(X_{i,a_1}X_{i,a_2}|X_{i,B}) &= \begin{bmatrix} K_{a_1a_1} & K_{a_1a_2} \\ K_{a_2a_1} & K_{a_2a_2} \end{bmatrix}^{-1} \\ \therefore \ (X_{i,a_1} \perp\!\!\!\perp X_{i,a_2})|X_{i,B} &\Leftrightarrow K_{a_1a_2} = K_{a_2a_1} = 0 \end{aligned} \tag{1.8}$$

The inverse of the sample covariance matrix, $S^{-1} = (XX'/n)^{-1}$, might naturally be considered as an estimator of the inverse covariance matrix. However, given the often high dimensional nature of the problem, $S$ is typically rank-deficient and therefore not invertable. Furthermore, in cases where $S$ is invertible, its inverse might not have the sparsity expected of the sparse networks typically seen in biology. For these two reasons, regularized estimators have been proposed for direct estimation of the inverse covariance matrix, $\hat{K}$.

Meinshausen and Bühlmann [53] proposed estimation of $K$ by regressing each node on all the other nodes iteratively using an L1-regularized regression. In this

method, using the coefficients estimated by the mentioned regression, they defined a 'neighborhood' set of nodes for each node such that conditional on its neighborhood set, each node is independent of other nodes. Specifically, they defined the neighborhood set for any node $a$, as the set of nodes corresponding to the non-zero coefficients from regressing $X_a$ on $X_{\setminus a}$. As $\hat{K}_{.a}$ is proportional to the coefficient $\hat{\beta}_{.a}$, the defined neighborhood for node a corresponds to the non-zero elements of $\hat{K}$ which are used to define the edge-sets between node $a$ and the other nodes. Therefore, through regularized linear regression, Meinshausen and Bühlmann's method allows identification of zero elements of $\hat{K}$ matrix as a direct estimation of $K$ would.

Later, a number of authors including Yuan and Lin [92] and Friedman *et al.* [22] proposed direct estimation of the inverse covariance matrix, $K$, via a LASSO-type regularization of the $K$ matrix. In this approach, which is known as graphical LASSO, penalized likelihood of the GGM is used as the objective function to be optimized over $K$. Specifically, for a mean-centered dataset, the optimization problem in Eq. (1.9) is solved.

$$\begin{aligned} &\arg\max_{K} \mathcal{L}(K) - \lambda\|K\|_1 \\ &\text{where } \mathcal{L}(K) \equiv \log\ \det K - \text{trace}(SK) \text{ and } S \equiv XX'/n \end{aligned} \tag{1.9}$$

using coordinate descent, Friedman *et al.* showed that graphical LASSO, Eq. (1.9), can perform network inference for GGMs with hundreds of nodes in seconds. Witten *et al.* [88] and Mazumder and Hastie [50] further improved the computational efficiency of this approach. They noted that when (the features can be ordered such that) the sparse estimate of $K$ matrix is block diagonal, the optimization for each block can be performed separately thereby gaining substantial reduction in processing time. The authors also showed that the sample covariance matrix, $S$, and the optimization penalty, $\lambda$, can be used to identify the sub-networks corresponding to each block. This algorithm is particularly relevant to biological settings where biological molecules are often more strongly inter-dependent within their immediate sub-networks while (conditional on the members of their sub-network) they might have weak inter-dependencies to other outside of their sub-network molecules.

In recent years, regularized GGMs have attracted great interest in statistics, computer science and computational biology leading to many additional extensions to the mentioned methods. For example, Guo *et al.* suggested joint estimation of different networks that share common structures [32]. Similarly, Danaher *et al.* [12] jointly estimated precision matrices across similar but different classes of networks (e.g. healthy vs cancer-affected network) by imposing a fused LASSO penalty (in addition to the sparsity inducing penalty) which encouraged similarity across classes. Compared to separate estimation across classes, these methods more efficiently use all the data available to estimate common structures across networks, while also allow for differences across the classes to be inferred. As another example, Voorman *et al.* [85] extended the neighborhood selection of Meinshausen and Bühlmann [53] to accommodate non-linearity in data that might not be best fit using (penalized) linear regression. They showed that their method outperforms graphical LASSO (in positive predictive value) when relationships between nodes are non-linear while in the case of linearly dependent nodes, the method performs on-par with graphical LASSO. These are but a small sampling of the recent developments in regularized

graphical models, which are collectively expected to narrow the gap between the computational capabilities of these methods and their utility in answering medical and biological questions.

### 1.2.3 Establishing Causal Dependence

As discussed earlier, a graphical model capable of establishing conditional dependencies is far more informative than one that only shows patterns of co-regulations. Similarly, establishing causal dependence between the nodes provides a much clearer scientific picture of the underlying interactions than the knowledge of conditional dependence relationships. In the context of networks, causal dependencies can be naturally thought of as the direction in the flow of information (or influence) and is symbolized by directional edges (arrows) in graphical models.

To establish causal dependencies however one needs more information than what is contained within the data from a passive observation of the state a network (e.g. concentration levels of the nodes). Specifically, establishing causal dependence requires the data to contain information on the direction of the flow of influence through the network. This information can be acquired by perturbing the network as described in Section 1.1 and recording the propagation of the perturbation effect through the network.

For BNs, an approach proposed by Cooper and Yoo [11] can be used to infer causality from a mixture of observational and experimental data. Similarly for regularized graphical models, Fu and Zhou [25] described an approach which allows for using perturbation information to infer the directed graphs from experimental data. Both these methods rely on the fact that the observed value of a perturbed node could mostly (if not entirely) reflect the experimental intervention rather than the influence of its parent nodes. Based on that, when estimating the edges between any node $c$ and its parents, these two methods simply exclude any observations in which the value of node $c$ was directly altered (e.g. set to zero by KO) by a perturbation.

Order of events, although not sufficient for establishing causal relationships, might be used as an evidence in support of the hypothesis of causality. Therefore in the absence of perturbation data, incorporation of temporal information into the network inference task can aid the discovery of causal relationships.

Dynamic Bayesian networks (DBN) are among the methods that can incorporate temporal information into network inference [17]. DBNs are essentially BNs where each node observed at a given time point is represented by a unique node and edges are only allowed from earlier to later time points. As DBNs are computationally indistinguishable from BNs, they are inferred similar to any other BN. However given that each node is represented once for each time point observed, they are often computationally too expensive for use in all but those networks with relatively few nodes observed over a small number of time points.

Regularized graphical models can also be used to infer networks based on time-course observations. For example, similar to DBN, the graphical model can be expanded such that each observed node at a given time is represented by a unique node. Then, an approach akin to that of Meinshausen and Bühlmann [53] can be employed but with edges only allowed (on this expanded network) from nodes at earlier time points to those on the latest time point. The resultant approach follows

closely the framework of Granger causality [28] and is studied in the context of graphical models by a number authors [1, 19, 71].

Data are most informative about the dynamics of networks when they capture temporal changes following experimental perturbations to the network. As a result, methods that can effectively integrate both temporal and interventional information into network inference (when such data are available) are more likely to be able to establish an accurate picture of causal dependencies. Using the approach by Cooper and Yoo [11], Eaton and Murphy implemented a DBN for network inference based on a mixture of observational and experimental data [17]. Also noteworthy in this area is the extension of 'nested effect models' (NEM) [48] method to temporal data by Fröhlich *et al.* [24], which they called dynamic nested effect model (dynoNEM).

DynoNEM closely follows the framework of DBNs with two main differences. First, it follows the NEM paradigm, in which networks are assumed to be comprised of nodes that are directly perturbed (but not observed) and observed nodes that are each affected by one (and only one) perturbed gene (and are not directly perturbed). Second, DynoNEM relaxes the first order Markov assumption. Specifically, time-lagged effects (i.e. effects across multiple time slices) are allowed in DynoNEM. Despite these differences, DynoNEM closely resembles DBNs and are subject to some of the same limitations of BNs discussed earlier. In the next chapter, an alternative approach to network inference based on time course data on perturbed networks is discussed, which follows the paradigm of regularized graphical models and can model larger networks than DBNs generally can.

# Chapter 2

# A Non-linear VAR model for Network Inference from Time-Course Data

As discussed in the previous chapter, time-course measurements on perturbed networks provide rich data for inferring the dynamic relations between the elements of a network. The method proposed here was inspired by one such dataset. Specifically, this method seeks to infer a static (as opposed to time varying) network from a mixture of interventional and observational time course data collected on a network. This method draws upon the work by Shojaie *et al.* [71]. Similar to that work, the method employs a penalized vector autoregressive (VAR) framework for modeling the time course data. Following the estimation of the VAR model, a thresholding approach is taken to further eliminate the edges that capture weak interactions and are likely to be spurious. This work extends the earlier work by Shojaie *et al.* by generalizing the regression to better capture non-linear relationships between the elements of the network. It furthermore accommodates both observational and interventional data following the approach by Fu and Zhou [25]. Lastly, the method allows for incorporation of prior knowledge to restrict the set of edges from which the network structure is to be constructed. This chapter provides a background on the methods employed in this work.

## 2.1 Vector Autoregressive Model of a Network

Vector autroregressive (VAR) models have been extensively used for studying the evolution of multiple inter-related components over time [76]. In the context of network inference, a VAR model seeks to explain the latest observed state of the network based on its observed states over the previous $d$ time points. Formally, for a network observed over $T$ time points, a VAR model with Gaussian noise can be written as:

$$X^T = X^{T-1}A^1 + ... + X^{T-d}A^d + \epsilon^T,\ \epsilon^T \sim \mathcal{MN}(\mathbf{0}_{\mathrm{n}\times\mathrm{p}}, \sigma^2 I_n, \sigma^2 I_p) \tag{2.1}$$

where $X^t \in \mathbb{R}^{\mathrm{n}\times\mathrm{p}}$ is a matrix of $n$ observations of the $p$ nodes of the network at time $t$ and $I_d \in \mathbb{R}^{\mathrm{d}\times\mathrm{d}}$ signifies a d $\times$ d identity matrix. $\mathcal{MN}(\boldsymbol{\mu}, \Sigma_n, \Sigma_p)$ is the n $\times$ p

matrix normal distribution [13]. The matrix normal distribution here specifies a noise matrix $\epsilon^T \in \mathbb{R}^{\mathrm{n}\times\mathrm{p}}$ with all rows distributed *iid* as $\epsilon^T_{i\cdot} \sim N(0, \sigma^2 I_p)$ and all columns distributed *iid* as $\epsilon^T_{\cdot j} \sim N(0, \sigma^2 I_n)$ the two covariance matrices in this notation specify respectively the rows and columns covariance structure of the random matrix(for additional information see [13]).

Here $A^t_{ij}$ captures the conditional dependence of $node_j$ at time $T$ on $node_i$ at time $t$. In other words, $A^t$ can be thought of as an estimated adjacency matrix capturing the conditional dependence of the $p$ nodes of the network at time $T$ on the $p$ nodes of the network at time $t$. Therefore, by estimating $A^t$ matrices we can attempt to capture inter-relation of the nodes of a network over time. Note that since the captured relations are directional (i.e. always from the earlier time points to the latest one), it conveys sequence of events and the resultant inference is a directed graph (in a temporal sense). Solving for the $A$ matrices of Eq. (2.1) amounts to regressing each column of $X^T$ on $\mathcal{X}$.

$$X^T = \mathcal{X}\mathcal{A} + \epsilon^T,\ \epsilon^T \sim \mathcal{MN}(\mathbf{0}_{\mathrm{n}\times\mathrm{p}}, \sigma^2 I_n, \sigma^2 I_p) \tag{2.2}$$

where $\mathcal{X} = [X^{T-1}, ..., X^{T-d}] \in \mathbb{R}^{\mathrm{n}\times\mathrm{q}}$, $\mathcal{A} = [(A^1)', ..., (A^d)']' \in \mathbb{R}^{\mathrm{q}\times\mathrm{p}}$, $q = d \times p$ and $A'$ is the transpose of matrix $A$.

In other words, the model described thus far involves regressing the state of each node at the last observed time point, $T$, on all $d$ previously observed states of all $p$ nodes.

## 2.2 Thresholded LASSO

Let's define $node_j(t)$ to represent the state of the $j^{\mathrm{th}}$ node at time $t$. Considering $\mathscr{A} = \{A^t : t \in \{1, \cdots, T\}\}$ as the set of adjacency matrices of a network, we could define the parent set of $node_j(T)$ as $\pi_j \equiv \{node_i(t) : A^t_{ij} \neq 0, \mathrm{for} i \in \{1, ..., p\}, t < T\}$. Given this parents set, our network of conditional independence allows us to define:

$$(node_j(T)|\pi_j) \perp\!\!\!\perp node_k(t),\ \forall\, node_k(t) \notin \pi_j \text{ and } \forall t < T \tag{2.3}$$

Assuming sparsity of bio-molecular networks, $\pi_j$ is often a very small subset of the node-set $\mathscr{V} = \{node_m(t) : m \in \{1, \cdots, p\}, t < T\}$. In other words, $|\pi_j| \lll |\mathscr{V}|\ \forall\, j \in \{1, ...p\}$. In this setting, when we regress $X^T_{\cdot j}$ (i.e. the $j^{\mathrm{th}}$ column of $X^T$ corresponding to $node_j(T)$) on $\mathcal{X}$, only the small subset of columns of $\mathcal{X}$ corresponding to $\pi_j$ reflect true conditional dependencies, while in truth $X^T_{\cdot j}$ is conditionally independent of the great majority of the columns of $\mathcal{X}$. Given the large number of predictors in this regression that the response variable is conditionally independent of, least square solutions are prone to reflect spurious correlations that exist between the response variable and the conditionally independent predictors.

penalized regression approaches like LASSO [80] have been used effectively to reduce the effect of spurious correlations on regression. This benefit however comes at the cost of biasing the estimated coefficients. As the task of inferring the structure of a network aims to distinguish true conditional dependencies from spurious associations, quantifying the exact strength of correlations are not as critical, which makes penalized regression appealing for network structure learning.

Additionally, given the the experimental constraints, the system of equations for solving Eq. (2.2) can be often underdetermined ($n \ll q$). Regressing $X^T$ on $\mathcal{X}$ in this setting does not lead to a unique least square solution. This further makes the case for use of penalized regression to capture the important inter-relations that comprise the structure of the underlying network.

Shojaie *et al.* [71] estimated $\mathcal{A}$ in Eq. (2.2) via a three-step procedure that involved two tuning parameters, $\lambda_n$ and $\tau$ and can be summarized as follows:

1. Using LASSO, fit Eq. (2.2) and find the estimated adjacency matrices, $\bar{A}^t$, for $t \in \{1, \cdots, T-1\}$. This is equivalent to solving the optimization problem (2.4) for $\forall\, j \in \{1, \cdots, p\}$. As the true order of VAR is unknown, here d = T-1 (i.e. regression is performed over all but the last observed time point).

$$\underset{\alpha_0 \in \mathbb{R}^1, \mathcal{A}_{.j} \in \mathbb{R}^q}{\arg\min} \; n^{-1} \|X^T_{.j} - \alpha_0 - \mathcal{X}\mathcal{A}_{.j}\|_2^2 + \lambda \|\mathcal{A}_{.j}\|_1 \tag{2.4}$$

2. Acquire the thresholded estimate of the adjacency matrices in Eq. (2.1), $\hat{A}^t$, by thresholding every element of $\bar{A}^t$ for $\forall\; t \in \{1, ..., T-1\}$ as follows:

$$\hat{A^t_{ij}} = \bar{A^t_{ij}} 1_{\{\|\bar{A}^t\|_0 < \frac{p^2\beta}{T-1} \text{ and } |\bar{A^t_{ij}}| < \tau\}} \tag{2.5}$$

   where $\beta$ is the desired rate of type II error (e.g. 0.1) for detecting an edge between two nodes.

3. Estimate the order of the VAR model, $\hat{d}$, as follows:

$$\hat{d} = \max_t \{t : \|\hat{A}^t\|_0 < \frac{p^2\beta}{T-1}\} \tag{2.6}$$

For the two tuning parameters, $\lambda$ and $\tau$, they suggested the following:

$$\begin{aligned} \lambda_n &= c_1 \sigma \lambda_0 \\ \tau_n &= c_2 \sigma \lambda_0 \end{aligned} \tag{2.7}$$

where following Zhou *et.al* [93] $\lambda_0 = \sqrt{2\log((T-1)p)/n}$. The constant values $c_1$ and $c_2$ were set based on empirical evaluations. They further suggested the following relation $c_2 = 4c_1$ to simplify tuning. Alternatively, they suggested using general tuning procedures such as cross validation (CV) for finding the optimal values of $\lambda$ and $\tau$.

## 2.3 Non-linear VAR Model of Network

As discussed in Section 2.2, the task of inferring the structure of a network can be broadly thought of as distinguishing (rather than quantifying) true conditional dependencies between nodes from the spurious associations due to noise. In this context, the ability to detect non-linear relations between nodes is expected to enhance our ability to detect conditional dependencies when the inter-relations might not necessarily be linear [85].

The approach to network inference outlined in Section 2.1 can be easily modified to better accommodate any non-linearity in relations between nodes. Specifically, Eq. (2.1) is a especial case of additive models (AMs) [5]. Considering the VAR model in the context AMs, we can expand the observations using a smoother (e.g. B-spline) to better capture higher order relationships between the network components. This expanded model can be represented as:

$$X^T = Y^{T-1}B^1 + \cdots + Y^{T-d}B^d + \epsilon^T, \epsilon^T \sim \mathcal{MN}(\mathbf{0}_{\mathrm{n}\times\mathrm{p}}, \sigma^2 I_n, \sigma^2 I_p) \tag{2.8}$$

Here, $Y^t = [f(X^t_{.1}), \cdots, f(X^t_{.p})] \in \mathbb{R}^{\mathrm{n}\times\mathrm{u}}$ where $X^t_{.i}$ is the $i^{\text{th}}$ column of $X^t$ and $u = p \times k$. $f(.)$ is the smoother of choice mapping any n-vector to an $n \times k$ matrix. $k$ is the order of the smoother, for example, $k = 3$ when $f(.)$ is a cubic B-spline. In this transformation, $B^t \in \mathbb{R}^{\mathrm{u}\times\mathrm{p}}$ are the estimated coefficients relating the expanded observations at time $T-t$ to the state of the $p$ nodes at time $T$. $B^t$ could be related to a generalized [1] concept of adjacency matrix, $\breve{A}^t$, defined by the following identity:

$$\breve{A}^t_{ij} \equiv \|B^t_{G_{ij}}\|_2 \text{ for } i, j \in \{1, ..., p\} \tag{2.9}$$

where $B^t_{G_{ij}} \in \mathbb{R}^k$ are the k coefficient relating the expanded $X^t_{.i}$ to the $X^T_{.j}$.

We could express Eq. (2.8) in a more compact form:

$$X^T = \mathcal{Y}\mathcal{B} + \epsilon^T, \epsilon^T \sim \mathcal{MN}(\mathbf{0}_{\mathrm{n}\times\mathrm{p}}, \sigma^2 I_n, \sigma^2 I_p) \tag{2.10}$$

where $\mathcal{Y} = [f(\mathcal{X}_{.1}), \cdots, f(\mathcal{X}_{.q})] \in \mathbb{R}^{\mathrm{n}\times\mathrm{s}}$, $\mathcal{B} = [(B^1)', ..., (B^d)']' \in \mathbb{R}^{\mathrm{s}\times\mathrm{p}}$, and $s = q \times k$.

Transforming the problem from Eq. (2.2) to Eq. (2.10) increases the number of parameters to be estimated by k folds, thereby exasperating the challenges relating to the (typically) high-dimensional nature of the task at hand (see sec 2.2). However, given that each edge in the network corresponds to groups of size k within $\mathcal{B}$, penalizing these groups together is a natural approach that also alleviates the mentioned challenge of high-dimentionality. Specifically, the three steps procedure of Shojaie *et al.* outlined in Section 2.2 can be generalized as follows:

1. Using group LASSO [91], fit Eq. (2.10) and find the estimated adjacency matrices $\breve{A}^t$ (using the identity (2.9)) for $t \in \{1, ..., T-1\}$. This is equivalent to solving the following optimization problem for $\forall\, j \in \{1, ..., p\}$

$$\underset{\beta_0\in\mathbb{R}, \mathcal{B}_{.j}\in\mathbb{R}^s}{\arg\min} \frac{1}{2n}\|X^T_{.j} - \beta_0 - \mathcal{Y}\mathcal{B}_{.j}\|_2^2 + \lambda \sum_{i=1}^{p} \sqrt{k}\|\mathcal{B}_{g_{ij}}\|_1 \tag{2.11}$$

[1] This generalized adjacency matrix encodes higher order interactions.

Here $\mathcal{B}_{g_ij}$ is the group of k coefficients corresponding to the expanded $\mathcal{X}_{.)}$. Using the definition of $\mathcal{B}$ and the identity (2.9), $\breve{A}^t$ for $t \in \{1, ..., T-1\}$ is acquired.

2. Acquire the thresholded estimate of the adjacency matrices in Eq. (2.1), $\tilde{A}^t$, by thresholding every element of $\breve{A}^t$ for $\forall\ t \in \{1, ..., T-1\}$ as follows:

$$\tilde{A^t_{ij}} = \breve{A^t_{ij}} 1_{\{\|\breve{A}^t\|_0 < \frac{p^2\beta}{T-1} \text{ and } |\breve{A^t_{ij}}| < \tau\}} \tag{2.12}$$

where $\beta$ is the desired rate of type II error for detecting an edge between two nodes.

3. Estimate the order of the VAR model, $\hat{d}$ as follows:

$$\hat{d} = \max_t \{t : \|\tilde{A}^t\|_0 < \frac{p^2\beta}{T-1}\} \tag{2.13}$$

## 2.4 Tuning the Model

As outlined in Section 2.2 (and its generalized version of Section 2.3) the three-step procedure involves two tuning parameters: $\lambda$ and $\tau$. For setting these parameters one could use the Eq. (2.7). However, this would require estimation of $\sigma$ and also setting $c_1$ based on empirical observations. Cross validation (CV) is another approach for tuning. However, CV, being based on predictive error, is known to select a regression penalty, $\lambda$, that tends to under-shrink the coefficient estimates [69]. An attractive alternative to CV is tuning based on Schwarz Bayesian Information Criteria (SBIC). Schwarz introduced SBIC as a criterion for "selecting one of a number of models of different dimensions" [67]. Schwarz showed that when data, $\boldsymbol{x}$, is generated from an exponential family of distributions and assuming that different levels of model complexity (i.e. different number of parameters) are all equally likely, SBIC is asymptotically related to marginal probability of the data for a candidate model, $M_i$ according to Eq. (2.14).

$$\log P(\boldsymbol{x}|M_i) \approx \text{SBIC} = \log\ L(\hat{\boldsymbol{\theta}}_i|\boldsymbol{x}) - \frac{|\boldsymbol{\theta}_i|}{2} \log\ n \tag{2.14}$$

where $\boldsymbol{\theta}_i$ is the vector of model parameters, $\hat{\boldsymbol{\theta}}_i$ is the MLE of $\boldsymbol{\theta}_i$ given the data and $L(\hat{\boldsymbol{\theta}}_i|\boldsymbol{x})$ is the maximum likelihood of the data under model $M_i$. This criterion is also referred to as BIC and is often (scaled by -2 and) represented as

$$\text{BIC} = -2\ \log\ L(\hat{\boldsymbol{\theta}}_i|\boldsymbol{x}) + \text{DOF}\ \log\ n \tag{2.15}$$

where DOF is the degrees of freedom which for simple linear regression is equivalent to $|\boldsymbol{\theta}_i|$.

In the context of linear regression, when stochastic variations in the data are assumed to be normally distributed, Eq. (2.14) is simplified (up to an added constant) into

$$\text{BIC} = n \log \frac{\text{RSS}}{n} + \text{DOF} \log n \tag{2.16}$$

where RSS is the residual sum of squares from the regression.

Degrees of freedom in ordinary linear regression is well defined as the number of random variables that can vary independently within the constraint imposed by the value of estimated parameters. However, penalized regressions are often applied in high-dimensional settings with sparse solutions where a large number of variables included in the model are believed to be 'irrelevant'. In such settings (where the number of observations is far smaller than the number of variables) we clearly cannot rely on the intuitive concept of DOF defined for simple linear regression, but rather as the number of parameters of the model itself is an estimate, the true value of DOF can only be estimated. In the case of LASSO Zou *et al.* [94] showed that the number of non-zero coefficients is an unbiased estimator for DOF of the regression problem. This estimate however is not valid for group LASSO. In their original work on group LASSO, Yuan and Lin [91] proposed the following estimate for DOF:

$$\widehat{\text{DOF}}^{(\text{ortho})}_{\text{grplasso}} = \sum_g \mathbf{1}_{\{\|\hat{\beta}_g\|_2 > 0\}} + (p_g - 1) \sum_g \frac{\|\hat{\beta}_g\|_2}{\|\hat{\beta}_g^{LS}\|_2} \tag{2.17}$$

where $p_g$ was the size of group $g$, $\hat{\beta}_g$ is the vector of estimated coefficients for group $g$ estimated via group LASSO and $\hat{\beta}_g^{LS}$ is its ordinary least square counterpart.

Yuan and Lin's estimator, generalizes the DOF for LASSO derived by Zou *et al.* as when $p_g = 1$ it reduces to the estimate by Zou *et al.* for LASSO. However, this estimated DOF relies on an orthonomral design matrix (i.e. $X'X = I$). Furthermore, Yuan and Lin's estimator does not extend to underdetermined design matrices (for which $\hat{\beta}_g^{LS}$ can't be computed). Vaiter *et.al* further generalized the estimator of DOF for group LASSO to remove the orthogonality constraint. Their estimator additionally extends to high dimensional settings [81]:

$$\widehat{\text{DOF}}_{\text{grplasso}} = \text{Trace}[X_a(X_a'X_a + \lambda \mathscr{N}(\hat{\beta}_a) \odot (I - P_{\hat{\beta}_a}))^{-1} X_a'] \tag{2.18}$$

Here, $\hat{\beta}_a$ ('a' stands for active) is the vector of all non-zero coefficients estimated by group LASSO and $X_a$ is comprised of only those columns of X corresponding to $\hat{\beta}_a$. $I$ is the identity matrix. $\lambda$ is the penalty in Eq. (2.11).$P_{\hat{\beta}_a}$ is a block diagonal matrix, with the $g^{\text{th}}$ block corresponding to the matrix that projects onto the vector of estimated coefficients of the $g^{\text{th}}$ group. Specifically,

$$g^{\text{th}} \text{ block of } P_{\hat{\beta}_a} = \frac{\hat{\beta}_g \hat{\beta}_g{}'}{\hat{\beta}_g{}' \hat{\beta}_g} \tag{2.19}$$

$\mathscr{N}(\hat{\beta}_a)$ is a matrix operator that through element-wise multiplication (designated $\odot$) normalizes the $g^{\text{th}}$ block of $(I - P_{\hat{\beta}_a})$ by L2-norm of $\hat{\beta}_g$.

Using the $\widehat{\text{DOF}}_{\text{grplasso}}$ and assuming normal stochastic variations, we could use Eq. (2.16) to estimate BIC for each of the p group LASSO regressions of Eq. (2.11) on the first of the three steps procedure of Section 2.3.

However, as Chen and Chen have argued, in the high-dimensional settings "[t]he ordinary Bayesian information criterion is too liberal for model selection" [8]. They point out that in derivation of BIC (see [3]) all candidate models are assumed to be equally likely. This assumption effectively assigns to each k-parametered model group a probability proportional to the number of the models in that group. They use the following example to illustrate the problem with this assumption in the high dimensional settings: when the number of candidate parameters are 1000 (i.e. p = 1000), there are 1000 models with one parameter while there are 1000 * 999/2 models with two parameters. When all models are equally likely, the two-parametered models are effectively assigned a probability 999/2 times that of their one-parameter counterpart.

To correct this tendency of BIC for selection of more complex models in the high dimensional settings, Chen and Chen proposed an extension to BIC, which in the context of linear regression problems with Gaussian noise can be expressed as follows:

$$\text{EBIC} = n \log \frac{\text{RSS}}{n} + \text{DOF} \log n + 2\text{DOF}\gamma \log p, \quad \text{where } 0 \leq \gamma \tag{2.20}$$

At $\gamma = 0$, EBIC is reduces to BIC which works well in model selection when $n < p$. As the number of candidate predictors grow, $\gamma$ can be increased to counter the tendency of BIC in selecting models with higher number of predictors. Chen and Chen suggested setting $\gamma > 1 - \frac{1}{2k}$ where $k$ is defined through the equality, $p = O(n^k)$.

## 2.5 Modeling Network Perturbations

Data collected from a perturbed network differs from those of a passively observed network mainly in the information the data contain about the relation of the perturbed node and its parents. Consider the case where $node_i$ in a network is experimentally suppressed to zero. For molecular networks this would be equivalent to a KO experiment. As the influence of this perturbation can 'flow' to all the children of $node_i$, the data can be used towards inference of all edges downstream of $node_i$. Similarly, as the information flow among the nodes upstream or independent of $node_i$ is not affected by the perturbation, the data can also be used when inferring those edges as well. The only edges that are rendered ineffective as a result of the perturbation are those connecting the parents of $node_i$ to $node_i$. This observation is the principle behind Fu and Zhou's approach to inferring a causal GGM from interventional data [25]. In the context of the penalized regression scheme outlined in Section 2.3, their approach amounts to regressing each $X^T_{.i}$ column of Eq. (2.10) only on those rows of $\mathcal{Y}$ in which $node_i$ was not perturbed.

## 2.6 Non-linear VAR Model of a Network: The Algorithm

Based on the approaches discussed so far, a non-linear VAR model of a network can be constructed from time-course data that is observed passively, or following network

perturbations or any mixture thereof. As discussed in Section 2.3, in this approach sparsity is induced by both regularized regression and by thresholding. Therefore, the model has two main tuning parameters: the regression penalty designated by $\lambda_n$ in Eq. (2.11) and the thresholding constant designated by $\tau$ in Eq. (2.12). The model tuning will then involve evaluating the fit over the grid $(\lambda \in \mathscr{L}, \tau \in \mathscr{T})$, where the tunning parameter ranges, $\mathscr{L}$ and $\mathscr{T}$, are found empirically.

Specifically, when we have no perturbation of the network the algorithm can be summarized as follows:

**Algorithm 1** non-linear VAR for observational data

Scale: $X^T \leftarrow$ Scaled observations at $t = T$
Expand: $\mathcal{Y} \leftarrow f_{smooth}(\mathcal{X})$
**for** each $\lambda \in \mathscr{L}$ **do**
  Regress: $\hat{\mathcal{B}} \leftarrow \forall\, i \in \{1, \cdots, p\}$ solve grplasso$(X_i^T \sim \mathcal{Y}, \lambda)$
  Convert to Adj.: $\breve{A} \leftarrow$ format$(\hat{\mathcal{B}})$
  **for** each $\tau \in \mathscr{T}$ **do**
    Threshold: $\tilde{A} \leftarrow$ threshold$(\breve{A}, \tau)$
    Convert to Coef.: $\hat{\mathcal{B}}^{(th)} \leftarrow$ reformat$(\tilde{A})$
    Define Active: $active = \{i : \hat{\mathcal{B}}_i^{(th)} \neq 0\}$
    Refit: $\hat{\mathcal{B}}^{(refit)} \leftarrow$ grplasso$(X^T \sim \mathcal{Y}_{.active}, \lambda)$
    Evaluate: score$(\lambda, \tau) \leftarrow$ EBIC(RSS$(\hat{\mathcal{B}}^{(refit)}), \widehat{\text{DOF}}_{grplasso}(\hat{\mathcal{B}}^{(refit)})$)
  **end for**
**end for**
Get Tuned Param.: $(\lambda^*, \tau^*) \leftarrow \underset{\lambda \in \mathscr{L}, \tau \in \mathscr{T}}{\arg\min}$ score$(\lambda, \tau)$
Regress: $\hat{\mathcal{B}}^* \leftarrow \forall\, i \in \{1, \cdots, p\}$ solve grplasso$(X_i^T \sim \mathcal{Y}, \lambda^*)$
Convert to Adj.: $\breve{A}^* \leftarrow$ format$(\hat{\mathcal{B}}^*)$
Threshold: $\tilde{A}^* \leftarrow$ threshold$(\breve{A}^*, \tau^*)$
Get Est. Graph: $\hat{G}_{ij}$, where

$$\hat{G}_{ij} = \begin{cases} 1 \text{ if } \exists t: \text{ the element of } \tilde{A}^* \text{corresponding to } (node_i(t), node_j(T)) \neq 0 \\ 0 \text{ otherwise} \end{cases}$$

Here, $\hat{G} \in \{0,1\}^{\text{pxp}}$ is the adjacency matrix for a directional graph of the estimated network.

When we have a mixture of observational and interventional data, Algorithm (1) needs to be adjusted such that in all the 'Regress' steps, $X_i^T$ is regressed only on the subset of rows of $\mathcal{Y}$ in which $node_i$ was not perturbed.

## 2.7 Non-linear VAR Model of a Network: Variations and Extensions

Three simple extensions/variations to the method outlined in Section 2.6 are discussed in this section: model averaging, standardizatoin of group LASSO and imposing restriction on the structure e.g. by pre-filtering the list of possible edges.

### 2.7.1 Model Averaging

Consider an experiment in which a network is observed over 10 time points following a perturbation. Further suppose the perturbation influence 'flows' completely through the network by the $5^{\text{th}}$ time point and thereafter the network reverts to the pre-perturbation (steady) state. In this case, the first few time points in the data contain all the information on the dynamic relation of the the network components as revealed by the perturbation. This information will be completely missed when performing network inference as outlined in Section 2.6 as the steady state is independent of the early fluctuations.

Based on the example discussed, we could expect that the power to detect any given conditional dependency between a pair of nodes varies as a function of $T$ (the furthest time point used in estimation relative to the perturbation time). Furthermore, for different pairs of nodes the maximal power to detect conditional dependencies might occur at different time points. However, *a priori* we might not (and probably do not) know how to choose $T$ or there might not be a single best choice of $T$. In this setting, an averaging scheme over our uncertainty regarding $T$ could be beneficial.

Specifically, we could perform the inference method outlined in Section 2.6 $T-1$ times, each time, considering the the last time point to be in $\{2, \cdots, T\}$. The resultant $T-1$ adjacency matrices can then be averaged and subsequently converted to a binary matrix by setting cut-off (e.g. values above 0.8 can be set to 1 and below to 0) on the average value that might indicate a true edge.

The averaging scheme discussed, naively assigns equal weight to all adjacency matrices. One could argue (using similar argument as in the example discussed earlier) that such averaging ignores the difference in the 'goodness' of the $T-1$ models. In this case, a possible mitigation would be to perform a weighted average of the adjacency matrices. A likelihood score might be a good candidate for weight. The BIC [2] score, which is already computed for each model, can be used as an approximation to log likelihood score. This type of averaging, which might be referred to as Bayesian Model Averaging (BMA), assigns a weight $W_i$ to the inference from the $i^{\text{th}}$ model, $M_i$. As $\text{BIC}_i \approx -2\log(P(\text{data}|M_i))$, we could define the $W_i$ as in Eq. (2.21).

$$W_i = \frac{\exp(-\frac{1}{2}\text{BIC}_i)}{\sum_{j=1}^{T-1}\exp(-\frac{1}{2}\text{BIC}_j)} \tag{2.21}$$

Building on the method outlined in Section 2.6, this extension (BMA) could improve our ability to detect edges form the time course data, when the influence of the perturbations are expected to be transient and present only over a subset of the observations.

### 2.7.2 Standardized Group LASSO

When features of the data are correlated, Simon and Tibshirani suggested the use of 'Standard' group LASSO for group-regularized problems [72]. In the context of

[2] BIC is used generically to refer to BIC or EBIC. In practice EBIC was used in the implemented method.

network inference, the features (nodes) can be highly correlated. Fortunately, it is not difficult to standardize a 'regular' group LASSO problem. The standardized version of the group LASSO optimization problem of Eq. (2.11) can be expressed as follows:

$$\underset{\beta_0\in\mathbb{R},\mathcal{B}_{.j}\in\mathbb{R}^s}{\arg\min}\ \frac{1}{2n}\|X^T_{.j}-\beta_0-\mathcal{Y}\mathcal{B}_{.j}\|_2^2+\lambda\sum_{i=1}^{p}\sqrt{k}\|\mathcal{Y}_{.g_i}\mathcal{B}_{g_ij}\|_1 \tag{2.22}$$

Using QR decomposition, we could decompose each column group in the design matrix $\mathcal{Y}_{.g_i}\in\mathbb{R}^{\mathrm{n}\times\mathrm{k}}$ into $Q_iR_i$, where $Q_i\in\mathbb{R}^{\mathrm{n}\times\mathrm{k}}$ is orthonormal and $R_i\in\mathbb{R}^{\mathrm{k}\times\mathrm{k}}$ is upper triangular. We could then re-write Eq. (2.22) as follows:

$$\begin{aligned}
&\underset{\beta_0\in\mathbb{R},\mathcal{B}_{.j}\in\mathbb{R}^s}{\arg\min}\ \frac{1}{2n}\|X^T_{.j}-\beta_0-\sum_{i=1}^{p}\mathcal{Y}_{.g_i}\mathcal{B}_{g_ij}\|_2^2+\lambda\sum_{i=1}^{p}\sqrt{k}\|\mathcal{Y}_{.g_i}\mathcal{B}_{g_ij}\|_1\\
&=\underset{\beta_0\in\mathbb{R},\mathcal{B}_{.j}\in\mathbb{R}^s}{\arg\min}\ \frac{1}{2n}\|X^T_{.j}-\beta_0-\sum_{i=1}^{p}Q_iR_i\mathcal{B}_{g_ij}\|_2^2+\lambda\sum_{i=1}^{p}\sqrt{k(Q_iR_i\mathcal{B}_{g_ij})'(Q_iR_i\mathcal{B}_{g_ij})}\\
&=\underset{\beta_0\in\mathbb{R},\mathcal{B}_{.j}\in\mathbb{R}^s}{\arg\min}\ \frac{1}{2n}\|X^T_{.j}-\beta_0-\sum_{i=1}^{p}Q_i\mathcal{C}_{g_ij}\|_2^2+\lambda\sum_{i=1}^{p}\sqrt{k(R_i\mathcal{B}_{g_ij})'Q_i'Q_i(R_i\mathcal{B}_{g_ij})}\\
&=\underset{\beta_0\in\mathbb{R},\mathcal{B}_{.j}\in\mathbb{R}^s}{\arg\min}\ \frac{1}{2n}\|X^T_{.j}-\beta_0-\mathcal{Q}\mathcal{C}_{g_ij}\|_2^2+\lambda\sum_{i=1}^{p}\sqrt{k}\|\mathcal{C}_{g_ij}\|_1
\end{aligned} \tag{2.23}$$

where $\mathcal{C}_{g_ij}\equiv R_i\mathcal{B}_{g_ij}$ and $\mathcal{Q}\equiv[Q_1,\cdots,Q_p]$

The last expression in Eq. (2.23) shows that solving the standardized group LASSO problem of Eq. (2.22) is equivalent to performing 'regular' group LASSO regression of $X^T_{.j}$ on $\mathcal{Q}$. Then, $\mathcal{B}$ can be recovered using the relation: $\mathcal{B}_{g_ij}=R_i^{-1}\mathcal{C}_{g_ij}$.

An estimate of DOF for standardized group LASSO was derived by Petersen *et al.* [60], which can be used in computation of EBIC. With these two small modifications, the method (2.6) can use the standardized group LASSO penalty.

### 2.7.3 Imposing Restriction on the Graph

As discussed in the previous chapter, network estimation problems are often high-dimensional. In such settings, the estimation task could benefit from excluding features (edges) that are known or thought to be irrelevant. In a Bayesian framework, this could be done by setting the network priors to incorporate scientific knowledge about the structure of the network. This would be equivalent to differentially penalizing edges in the context of regularized graphical models. An extreme case of differential penalization would be to exclude edges from consideration on data-driven or scientific grounds.

When scientific knowledge is present, a weighted group LASSO could help incorporate this prior information into the inference. In the absence of such knowledge, one might still be able to use simple pairwise comparison techniques and perturbation information to exclude edges that are highly unlikely. As an example, consider a case where certain cell receptors are treated with their ligands and the sub-cellular signal transduction network is observed over time passively (control) as well as under

certain perturbation. If perturbation of a $node_i$ clearly did not alter the temporal profile of $node_j$ relative to the control profile of $node_j$, we could choose to exclude $node_j$ from the list of potential children of $node_i$. In this context of pre-filtering, as the cost of excluding a true edge by mistake is higher than not excluding a false edge, we can tolerate large type I error rates (e.g. 0.3) to exclude only edges that are highly unlikely to be present. A simple implementation of this idea could involve comparing the area under the curve (AUC) of temporal profiles for treated vs control observations via a simple t-test and exclude edges with large (e.g. $> 0.3$) multiplicity-corrected p-values. It needs to be emphasized that one must be careful and consider scientific knowledge and reasoning when applying such filtering schemes. However, when applied properly the resultant reduction in the dimension of the problem could benefit the inference task.

# Chapter 3

# Inferring Directed Networks Using The non-linear VAR model

The method summarized by Algorithm 1 along with the extensions discussed in Section 2.7 were implemented in `R`. This chapter details evaluation of the method using both simulated as well as knock down experimental data. This chapter is organized as follows: first, the accuracy of estimation of the group LASSO solution and its corresponding DOF is reported. Next, linear and non-linear fit solutions are compared when the data generating mechanism is truly linear. The comparison of the linear and non-linear fit is repeated on observations from *in silico E. coli* networks generated using `GeneNetWeaver` [66]. Lastly, the method is applied to temporal observations made on receptor tyrosine kinase (RTK) signaling network following the perturbation of the network.

## 3.1 Empirical verification of the estimation

As discussed in Chapter 2, the method implemented relies on accurate estimation of the solution to the group LASSO problem and its corresponding degrees of freedom. For solving the group LASSO problem, there are a number of implementations available in `R` (e.g. `grplasso, gglasso, grprep`). `gglasso` by Yang and Zou [90] offers both flexibility and speed and was chosen here for solving the group LASSO problem (and its standardized version). The accuracy of `gglasso` was independently verified using the generic `MATLAB` convex optimization solver cvx [29,30] (see appendix A.1). With the exact group LASSO optimization statement verified, the performance of the DOF estimators for group LASSO and standardized group LASSO were verified through simulation.

Similar to the approach taken by Zou *et al.* [94] in evaluation of DOF estimator for LASSO, empirical evaluation of DOF estimators relied on general definition of DOF as proposed by Efron [18]. Specifically, for n observation of data $y \sim (\mu, \sigma^2 I)$ and its generic mean estimate $\hat{\mu} = f(y)$ (where $f(.)$ is any function estimating the mean $\mu$), Efron proposed the following estimate of DOF for the estimator $\hat{\mu}$:

$$\text{DOF} = \sum_{i=1}^{n} \frac{\text{cov}(\hat{\mu}_i y_i)}{\sigma^2} \tag{3.1}$$

1000 independent data-sets each with observation number, $n = 45$ and feature size (pre-expansion), $p = 20$ were generated as outlined in Algorithm 2. The

response variable $Y$ was regressed on $X$ (using group LASSO or its standardized version) and the corresponding DOF was evaluated in each case and subsequently averaged over the simulations. Likewise the empirical covariance for each simulated data-set, $\widehat{\text{cov}}_s(\hat{Y}_s, Y_s)$, was computed, scaled and averaged to acquire an estimate of eq. (3.1).

**Algorithm 2** Simulation steps

**for** $s = 1$ to $S$ **do**
  Generate $D \in \mathbb{R}^{\text{n}\times\text{p}}$ where $D_{ij} \sim \text{Uni}(-2.5, 2.5)$
  Expand: $X \leftarrow \text{bspline}(D, \text{degrees} = 3)$
  Generate a by-group-sparse $\vec{\beta}$
  $\bar{Y}_s \leftarrow X\beta$
  $Y_s \leftarrow \bar{Y}_s + \epsilon_s$ where $\epsilon_s \sim N(0, \sigma^2)$
  **for** $\ell = 1$ to $\text{length}(\vec{\lambda})$ **do**
    $\hat{Y}_s(\lambda_\ell) \leftarrow f(X)$
    $\widehat{\text{DOF}}_s(\lambda_\ell) \leftarrow$ Estimate DOF
  **end for**
**end for**
$\text{DOF}(\lambda) \leftarrow \frac{1}{S\times\sigma^2} \sum_{s=1}^{S} \sum_{i=1}^{n} (\hat{Y}_{is}(\lambda) - \bar{Y}_{is}) \times (\epsilon_{is})$

For group LASSO, Figure 3.1 shows the concordance between the estimated DOF, which is computed using Eq. (2.18), and the expected DOF from Eq. (3.1). Based on these 1000 simulated data-sets. The estimate and its expected value show a great degree of agreement.

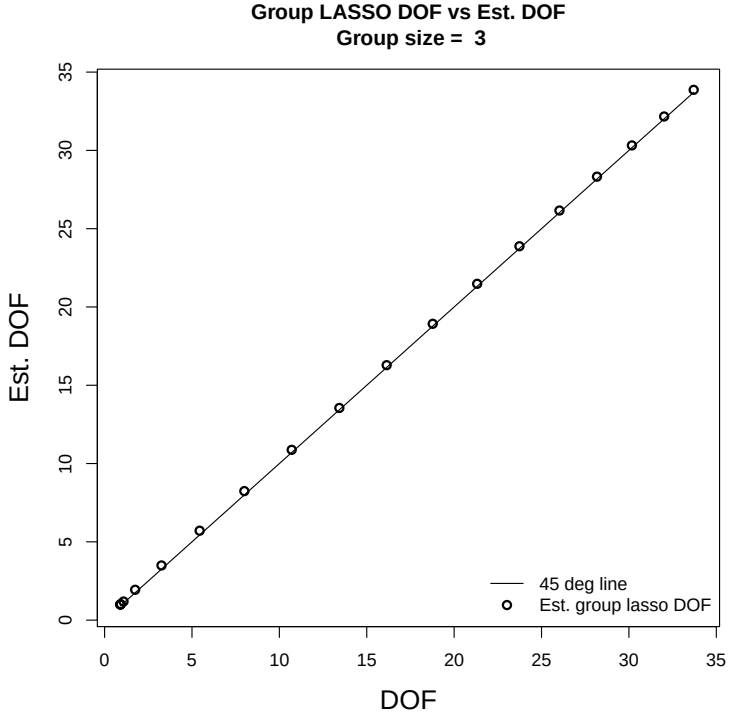



Figure 3.1: Estimated vs Expected DOF for group LASSO problem

Similarly, for standardized group LASSO, Figure 3.2 shows that the DOF estimator by Petersen *et al.* [60] is in good agreement with the expected value of DOF.

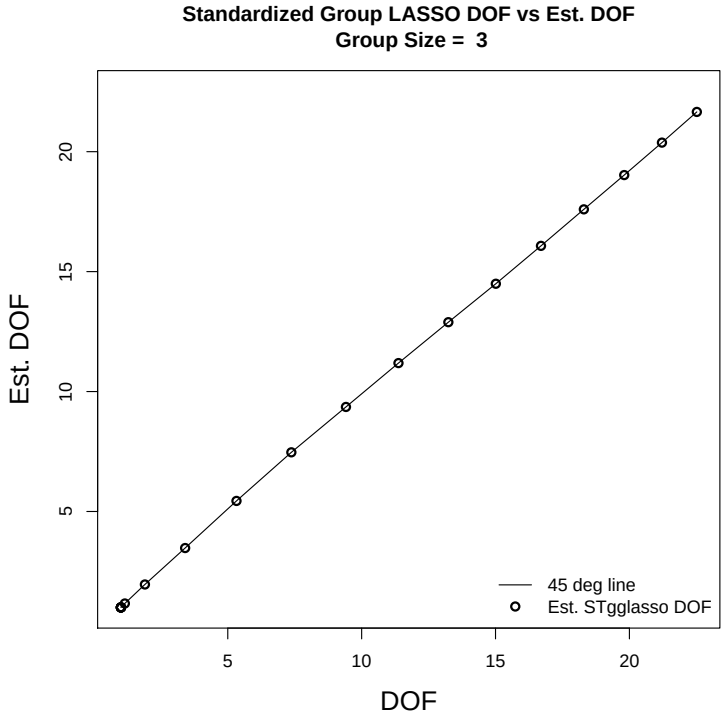



Figure 3.2: Estimated vs Expected DOF for standardized group LASSO problem

The inference method proposed in Chapter 2, relies on EBIC not only for model selection over the grid of tuning parameters but also if Bayesian model averaging is performed. The empirically verified accuracy of of the DOF estimates provides confidence that (assuming appropriateness of the Gaussian assumption which was used in deriving EBIC) the model selection (and averaging) will be unbiased.

## 3.2 Non-linear VAR model on Linear Data

The use of a smoother in Algorithm 1 aims at improving the power to detect edges when the inter-node relationships are not (necessarily) linear. The linear case can be viewed as a especial case of this generalized approach, and as such, the approach is expected to perform well when the node-to-node relationships are strictly linear.

To verify this, 100 data-sets were generated mimicking observations collected over 4 time points on a single 15-nodes network with 50 edges. Specifically, an adjacency matrix $\mathcal{A} \in \mathbb{R}^{45\times 15}$ was sampled from $\{0, 1\}$ representing the 15-nodes network (rolled out over the first 3 time points). Then, for each data-set, $D_i$, experimental matrix $\mathcal{X} \in \mathbb{R}^{100\times 45}$ was sampled from standard normal and use to simulate data according to $D_i = [\mathcal{X}, \mathcal{X}\mathcal{A} + \epsilon]$ where $\epsilon \sim \mathcal{MN}(\mathbf{0}_{100\times 15}, \sigma^2 I_{100}, \sigma^2 I_{15})$.

Here the the inference simply aimed at finding the non-zero coefficients of $\mathcal{A}$. This task was repeated for each data-set over a $C \times L$ grid of tuning parameters (i.e. a grid of $C$ thresholding constants and $L$ regularization penalties) and using three modeling approaches: linear fit with LASSO, cubic B-spline fit with group LASSO

and cubic B-spline fit with standardized group LASSO. After completion of each inference, evaluation metrics were computed for the result. These metrics included the EBIC as well as the call accuracy rates. The latter consisted of estimating the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) rates.

Following the analyses of all 100 data-sets, the evaluation metrics were averaged over the data-sets for each of the three modeling schemes. Positive predictive profile which shows the relationship between TP rates and all positive (AP = TP+FP) call rates was chosen as the primary metric for evaluation of the overall performance of each modeling scheme. A model performs better, the faster TP rate grows with the growth of AP rate (with maximum rate of TP/AP being 1). Figure 3.3, shows the positive predictive profiles for each of the three modeling approaches. The values plotted for each modeling approach, correspond to TP and AP rates when setting the thresholding constant to EBIC choice (i.e. the thresholding constant of the best overall fit) while varying the regularization penalties.

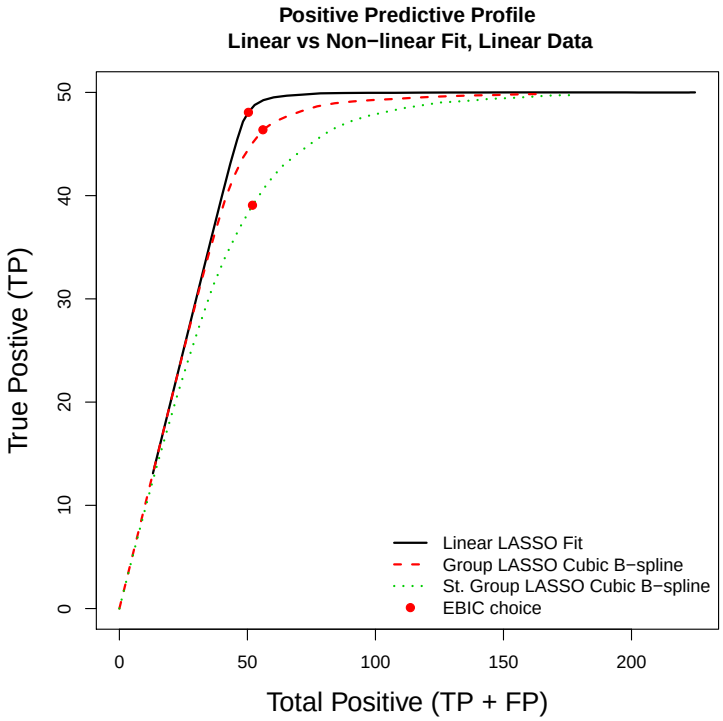



Figure 3.3: Positive predictive profiles of the linear and non-linear models

As expected, when network relations are strictly linear in nature, a linear model is the best choice (i.e. relative to other choices, the slope of the profile remains closest to 1 throughout). Group LASSO with a cubic B-spline, while not as good as the linear fit, shows a performance that is quite comparable to the linear fit. Standardized group LASSO meanwhile shows more deviation from the best performance line (i.e. line with slope 1). In general, there are no scientific reason to believe network relations are strictly linear. However, given the demonstrated performance of group LASSO, it is expected that Algorithm 1 can accommodate the higher order of interactions without a substantial reduction in the power to detect edges when relations are characterized primarily by lower orders of interactions.

In addition to the comparison of the three modeling schemes, the simulation results allow us to evaluate the performance of the EBIC as the model selection criteria. Within each profile, the EBIC choice (i.e. red circle) falls within the general area where the rate of positive predictive value (i.e. the slope of the profiles) experiences the most pronounced drop. In doing so, the EBIC choice seems to offer a good compromise between high TP and low FP rates.

Furthermore, Figures 3.4, and 3.5 show that thresholding improves performance for the linear fit and the group LASSO fit. However, this improvement is not seen for the standardized group LASSO fit in Figure 3.6. Standardized group LASSO differs from LASSO and group LASSO in that the fitted values rather than the coefficients are penalized. However, it is not immediately evident how that might render coefficients thresholding unable to improve the performance of the network estimation. While the behavior of the group LASSO fit may warrant further exploration, based on these results, thresholded group LASSO seems to provide the best choice for real data (where the node-to-node relationships are unlikely to be strictly linear).

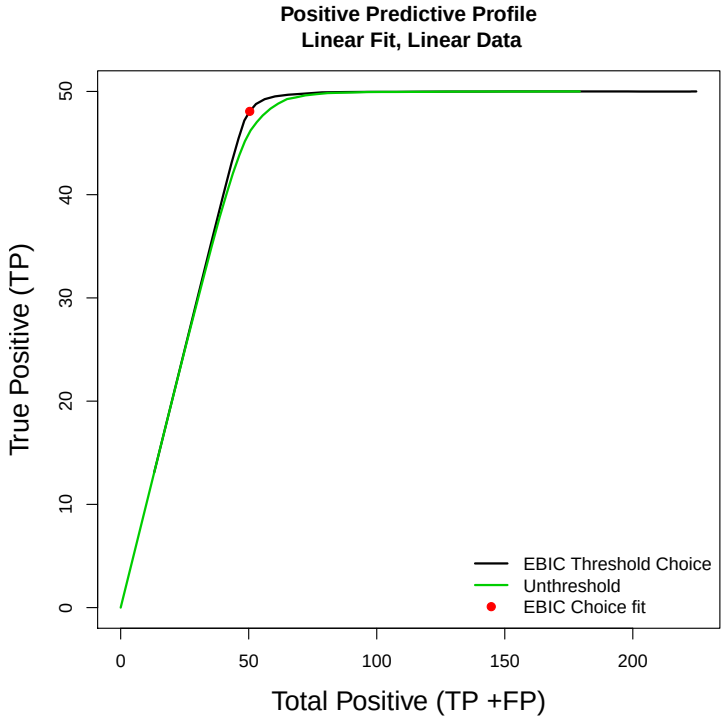



Figure 3.4: Positive predictive profiles for linear fit. The profiles for EBIC choice along with that of unthresholded case

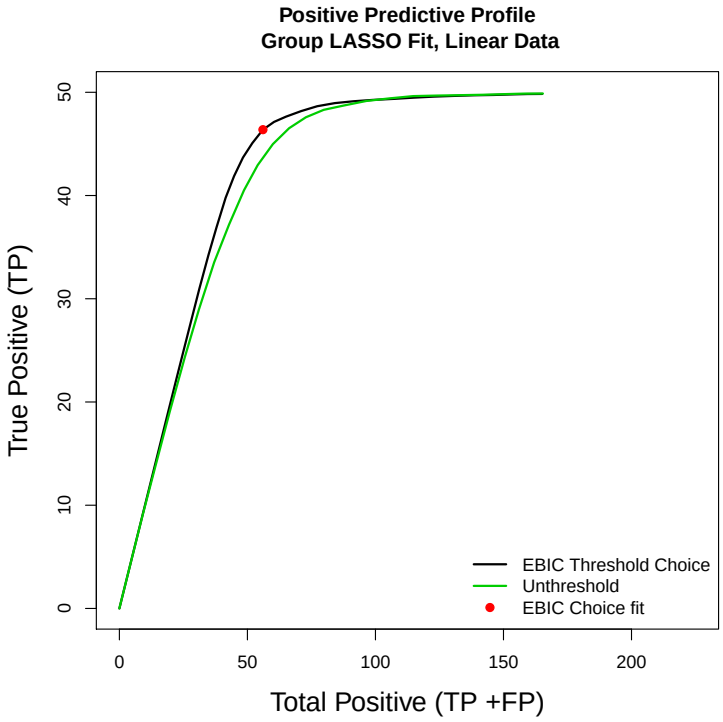



Figure 3.5: Positive predictive profiles for group LASSO cubic B-spline fit. The profiles for EBIC choice along with that of unthresholded case

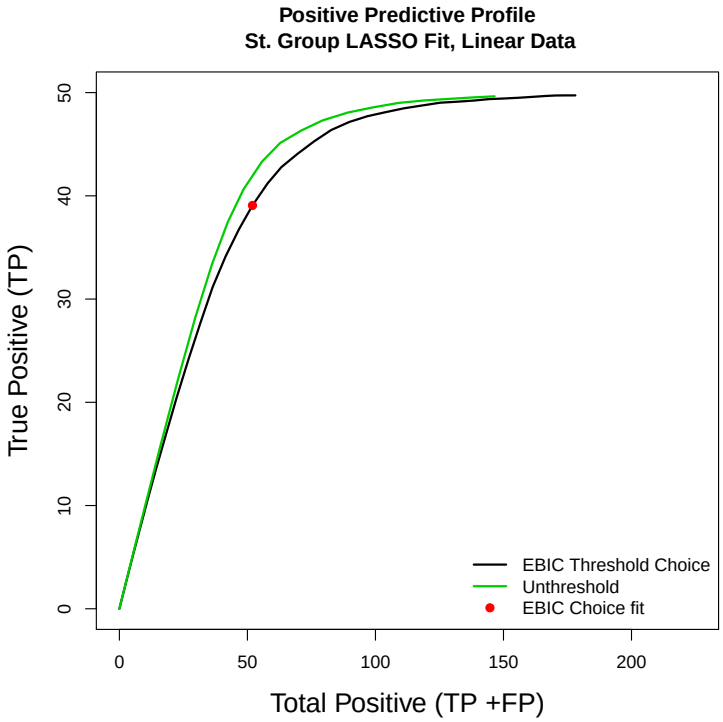



Figure 3.6: Positive predictive profiles for St. group LASSO cubic B-spline fit. The profiles for EBIC choice along with that of unthresholded cases

## 3.3 Non-linear VAR model on Data from *in silico E. coli* Gene Regulatory Network

Data from *E coli* gene regulatory network were simulated using the freeware `GeneNetWeaver` (GNW) [66]. In Brief, using GNW one can extract gene modules [1] and their (known) associated interaction dynamics. Given these interaction dynamics, the software constructs a system of stochastic differential equations (SDE) that relate the temporal variations in concentration of each molecule to its rate of production and degradation. Solving these SDEs, the software can simulate observations made on temporal variations in the concentration of the elements of the extracted network. GNW offers a host of other features to model observation noise, as well as to allow *in silico* perturbation experiments. GNW was used to simulate data for DREAM challenges 3, 4 and 5 [4], and the software allows generation of similar data-sets.

The *in silico E. coli* GRN data used in evaluation of the method discussed in Section 2.6 came from a 15-nodes network extracted in GNW as described above Figure 3.7. 1000 time-course observations were made from this network, each starting with a random multifactorial perturbation [2]. Each observation consisted of 11 time points observed over the course of 1000 simulated seconds.

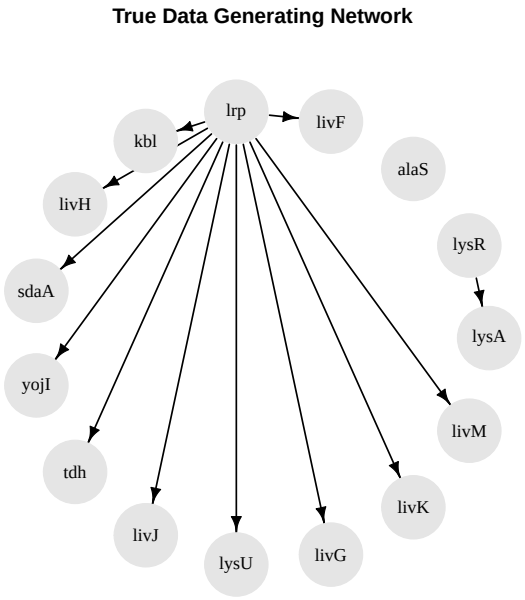



Figure 3.7: The 15-nodes *E. coli* network with 12 edges used by GNW to generate simulated data

Using the simulated 1000 time-course observations, 100 subsets of 90 observations were sampled and used to perform 100 rounds of network inference. Similar to the

[1] A group of genes known to be inter-connected at a rate higher than random

[2] multifactorial perturbation amounts to introducing a random shift to the concentration of multiple nodes at time 0.

evaluation of the method using linear data, the inference was completed over a grid of tuning parameters. Following each network inference, evaluation metrics (i.e. the EBIC score and the call accuracy rates) were computed and recorded. After completion of the 100 simulations, the evaluation metrics were averaged over the 100 simulations, and the positive predictive profiles were acquired (as were done in the case of the linear data).

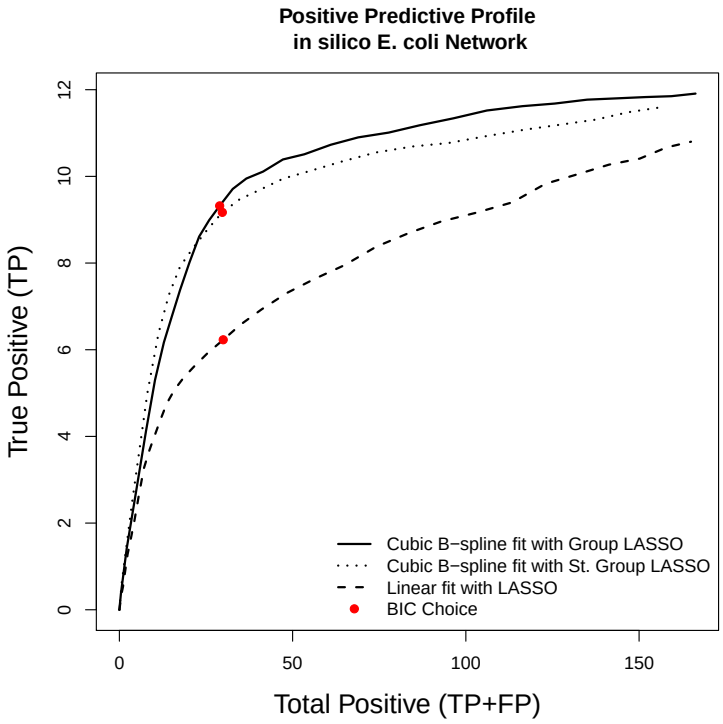



Figure 3.8: Positive predictive profiles of 3 modeling approaches on a 15-nodes *in silico E. coli* network with 12 edges

Relative to the simulated linear data, GNW data is expected to resemble real biological data much more closely. The positive predictive profiles based on GNW data are therefore considered to be a closer representation of the performance of the method with real biological data. Here, both group LASSO and standardized group LASSO fit which seek to accommodate higher order of interactions than linear fit outperformed a linear fit with LASSO. Similar to linear data, the EBIC choice fell in an area where the PPV experienced the most pronounced drop. A closer comparison of the performance of the three modeling approaches is provided in table 3.1.

| | Linear fit with LASSO | cubic B-spline fit with group LASSO | cubic B-spline fit with st. group LASSO |
|---|---|---|---|
| FP | 29.97 | 28.94 | 29.71 |
| TP | 6.23 | 9.32 | 9.17 |
| PPV | 0.21 | 0.32 | 0.31 |

Table 3.1: The evaluation metrics at EBIC choice comparing linear fit using LASSO to the cubic B-spline with group LASSO or st. group LASSO

## 3.4 Inferring EGFR Signaling Network from Temporal Observations on KD Cells

Epidermal growth factor receptor (EGFR) is a trans-membrane protein belonging to the receptor tyrosine kinases (RTKs) family of receptors. The cytoplasmic domain of an RTK contains multiple tyrosine residues which upon activation of the RTK can each serve to activate different 'relay'[3] molecules. In this manner RTK signaling allows complex regulations to be encoded by simple extra-cellular biochemical cues [87]. This also makes the potential harm of abnormal RTK activities particularly high. Abnormalities in RTK activation have been implicated in a many cancer types [75, 78, 89]

Given its role in oncogensis, RTK signaling pathways have been a target of cancer therapies (e.g. herceptin, gefitinib and imatinib [14, 31, 57] to name a few). However, it has become apparent that the overall success of this class of drugs is limited by innate or acquired resistance of tumors to the treatment. In a recent article, Wagner *et al.* employed a network-based analysis of six RTK signaling pathways to study the mechanism of developing resistance to RTK inhibitors. [86]. Studying the commonalities between network structures across the six RTK types, they classified these RTKs into three distinct classes. Using network estimates, the authors explained how the benefits of a treatment that inhibits an RTK could wane over time as other RTKs of the same class, which share similar signaling network structures, increasingly compensate for the inhibited RTK. Based on this, they observed that treating RTKs as classes (as opposed to individual targets) has the potential to mitigate acquired resistance of tumors to therapies.

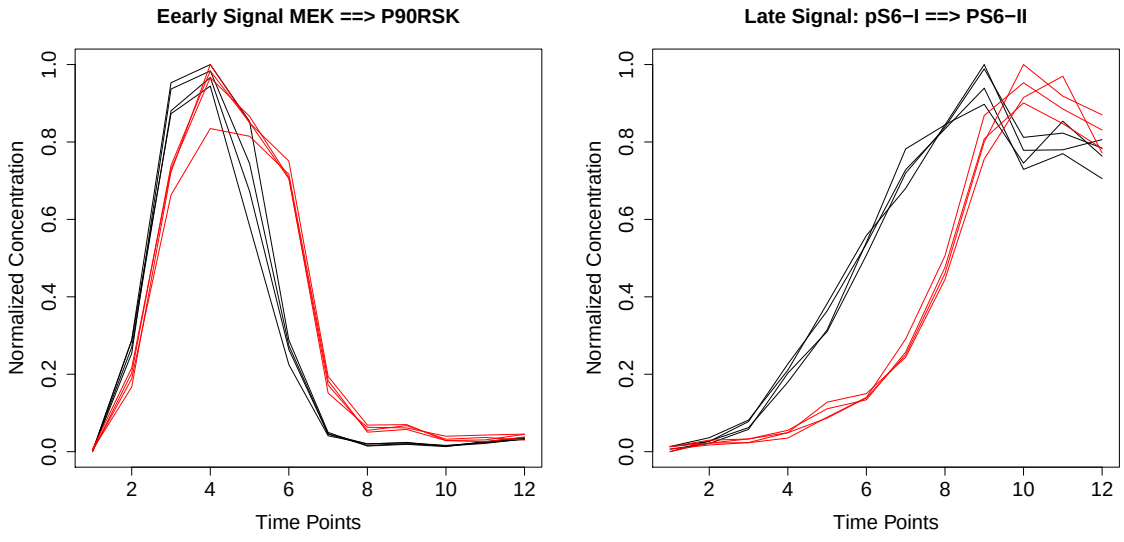
To infer RTK networks, knockdown (KD) perturbations targeting different molecules of the RTK signaling pathways were performed on each of the 6 groups. The KD cells were then stimulated (each with its own specific ligand) and observed over time for variations in the concentration of activated network components. Using DBN, the authors inferred the signaling networks. They subsequently were able to classify the RTKs based on their structural similarities (see Wagner *et al.* [86] for additional details).

The network inference method described in Section 2.6 was inspired by a similar dataset as discussed by Wagner *et al.* [86], which was generously provided by the co-authors of the mentioned article[4]. The data came from 24 perturbation experiments conducted on an isogenic cell-line expressing only EGFR receptors. In each experiment, a KD cell group was stimulated with EGF and subsequently observed over 12 time points at 4 replicates per condition. Each observation at a given time-point consisted of measurements on the abundance of 16 activated (i.e. phosphorylated) EGFR network components.

Similar data pre-processing (i.e. data cleaning and normalization) as described by Wagner *et al.* [86] were performed on the raw data. Exploratory evaluation of the data revealed that as the signal propagates through the network, some related network components exhibit their maximal co-variations early in the observation window, while other components exhibit co-variations later. This observation motivated the use of Bayesian model averaging (BMA) described in Section 2.7.1.

[3] A generic term referring to any of the molecules relaying the message from the receptor to the targeted downstream processes

[4] Drs. Sevecka, Wagner and Wolf-Yadlin




(a) (b)

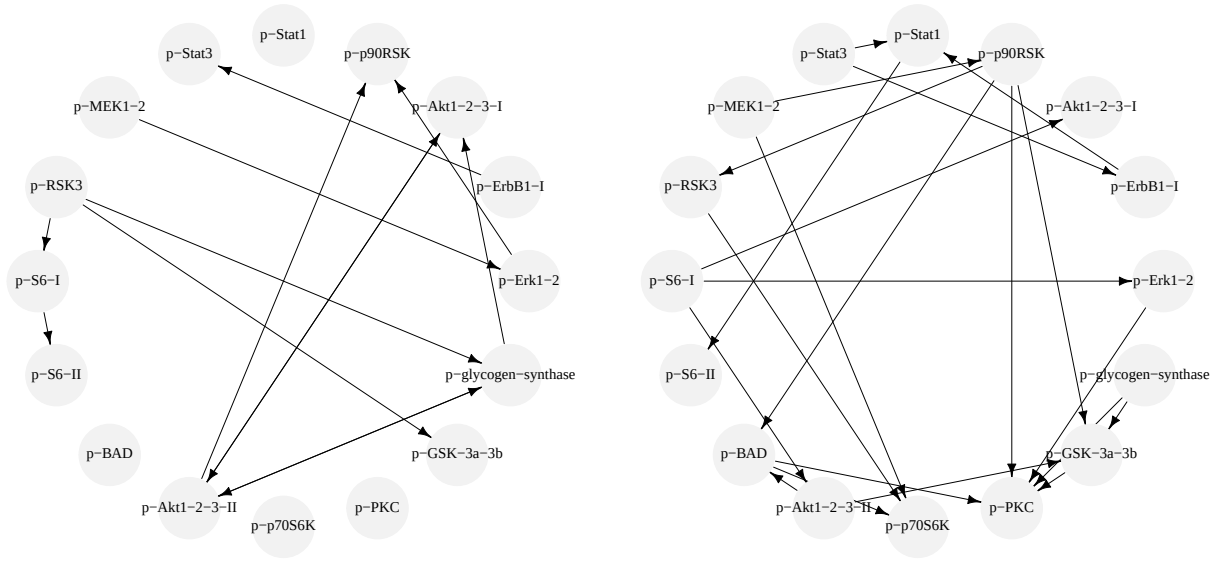
Figure 3.9: Each Figure shows the change in concentration of an activated parent (black) and its activated child (red) observed at 4 replicates. The signal from stimulation peaks early in 3.9a and by the $12^{\text{th}}$ time point, the signals revert to the pre-stimulation background levels, whereas for 3.9b the signal from stimulation peaks late. In both cases, considering covariations across all time points is far more informative about the structure than considering how the value of the child on the $12^{\text{th}}$ time point relates to all the earlier observed values for its parent.

Intuitively BMA seeks to score features of a network according to how prominently they explain the data over the period the network was observed (see Section 2.7.1 for details). The resultant adjacency matrix has elements with weights between 0 and 1. One can then explore the results by thresholding the adjacency matrix at various levels (between 0 and 1) and evaluate the outcome in the context of the scientific knowledge of the true network. To this end, following BMA, the edges with scores higher than 0.75 were binned into two edge-sets: the edge-set consisting of edges with scores greater than 0.95 and those with scores between 0.75 and 0.95. The two edge-sets are shown in Figure 3.10.

Considering the highest scoring 13 edges in Figure 3.10a, p-Erk [5] → p-p90RSK reflects the known phosphorylation of p90RSK by ERK (also known as MAPK) [16]. Also p-S6-I → p-S6-II and p-Akt1-2-3 II → p-Akt1-2-3 I reflect the transition between single and double phosphorylation of S6 and Akt molecules and are true dependencies. Furthermore, p-ErbB1-I → p-Stat3 [56], p-MEK1-2 → p-Erk1-2 [37, 38], p-Akt1-2-3 I → p-Akt1-2-3 II, p-RSK3[6] → p-S6-I and p-RSK3 → p-GSK-3a-3b [83] all agree with the literature.

In contrast, p-Akt1-2-3 → p-p90RSK does not agree with any known signaling interaction and is likely to be spurious. Likewise, the two edges from p-glycogen-synthase to p-Akt1-2-3 I and p-Akt1-2-3 II are not expected to have biological support. Likewise, p-RSK3 → p-glycogen-synthase and p-Akt1-2-3 II → p-glycogen-

[5] p-X denotes phosphorylated state of protein X.
[6] RSK3 is also known as ribosomal protein S6 kinase.




(a) (b)

Figure 3.10: The top 35 edges scored by BMA approach are binned into 2 categories. (a) corresponds to the scores $> .95$ and (b) corresponds to edges with scores between .75 and .95

synthase do not match the known order of signal transduction.

For the second edge-set shown in Figure 3.10b, p-p90RSK →p-BAD [77], and p-90RSK → p-GSK-3a-3b [83] match known interactions. Likewise, interactions p-Stat3 → p-Stat1, p-90RSK → p-RSK3, p-ErbB1-I → p-Stat1 [56] and p-Akt1-2-3 II → p-BAD [70] can be found in the literature.

The interactions p-glycogen-synthase → p-GSK-3a-3b → p-PKC are interesting as these molecules represent a chain of signal transduction. However, the inferred direction is the reverse of the expected direction [55]. Similarly, the interaction between p-GSK 3a-3b and p-Akt1-2-3-II is reported in the literature with the reverse direction. Other edges that are inferred between the correct pair of molecules but represent incorrect direction of signal flow are p-BAD → p-PKC [15], p-p-S6 I → p-Akt1-2-3 I [74], and p-S6 I → p-Akt1-2-3 II [74].

In addition to the incorrect direction, there are inferred edges between pairs of molecules that are not supported by biological understanding of this network. Examples of such false edges include p-MEK1-2 → p-p90RSK, p-S6 I → p-Erk1-2, p-glycogen synthase → p-PKC, p-BAD → p-p70S6K and p-Stat1 → p-S6 II. These edges are considered as spurious interactions.

Overall in the 35 edges in Figure 3.10 close to half (16/35) are edges that correctly reflect biological flow of information. With 9/13 of the edges in Figure 3.10a having biological support, for the top edge-set, the estimated positive predictive rate is about 0.7, whereas for the 22 edges with lower scores given in Figure 3.10b only $7/22 = 0.32$ could be supported by biological knowledge.

The known activation directions reflect the directions of phosphorylation [7]. Capturing this direction relies on observing a temporal lag between phosphorylation of

[7]Phosphorylation is the addition of a phosphate group to a substrate.

a kinase and the subsequent phosphorylation of a substrate by the activated kinase. When this lag is small relative to observation noise, discerning the direction could suffer. This consideration of the time lag relative to the observation noise needs to be factored in both experimental design as well as in the interpretation of the inferred networks.

The work presented here builds on the earlier work by Shojaie *et al.* for network estimation from time-course data [71]. The primary goal of this effort was to improve the earlier work by enabling it to better accommodate non-linear relations between the elements of a network. Evaluation of the method based on simulated data suggests that it succeeds in achieving this goal. Specifically, as was shown in Section 3.2 when the relations between the elements of a network are strictly linear, the method presented in this work shows similar performance to a method that assumes linear relationship between the elements of the network. However, when the data more closely resemble biological data, the presented method is expected to outperform its linear counterpart as seen in Section 3.3.

The model selection proposed using the EBIC metric was shown to generally select penalization levels in the ideal range (i.e. where the estimated TP is maximal and the rate of drop in PPV is highest). In the numerical evaluations performed, it was observed that EBIC performed well in selecting thresholding constants when the data was strictly linear and the network was not too sparse. With the *in silico E. coli* data, we observed that the EBIC choice generally tended toward over-thresholding the model. Although our investigations showed that thresholding could reduce selection of false variables beyond what can be achieved with penalization alone, this added benefit was not seen when performing the network inference on *in silico E. coli* data. In our empirical evaluation on why *in silico E. coli* did not benefit from thresholding, we observed that for this data, no significant difference exists between the distribution of estimated TP and FP coefficients. As thresholding relies on values FP coefficients to be typically smaller than those of TP coefficients, our observation explains the reason for why thresholding benfits were lacking for *in silico E. coli* data. While this needs further investigation, a practical approach would be to empirically restrict thresholding levels and then choose EBIC to tune the regression penalty. This approach was taken in the results presented for the experimental as well as for the *in silico E. coli* data.

Beyond its primary goal of improving network estimation performance through capturing non-linear relations in the data, the method described offers features that can assist investigators in applied network studies. As discussed in the previous chapter, the method can accommodate data comprised of both passive observations as well as observations collected following network perturbations. The method also allows the estimation task to be informed by prior knowledge of the network structure. Furthermore, the BMA functionality implemented allows for exploration of the features of a network by how prominently they explain the observed data throughout the observation window. In doing so, BMA seeks to increase the efficacy of estimating a static network from time course observations.

There are a number of areas where potential improvements can be made to the described algorithm. Currently, the method allows the user to restrict the set of edges from which a network can be constructed. Future versions can easily extended this feature to a more general case where edges can be differentially penalized to provide a finer control over incorporation of prior knowledge into the inference task.

Additional improvements to the method could include modification of Algorithm 1 to reduce computational costs. For example, it is more efficient to perform all group LASSO regressions at different penalization levels at once, store the results and subsequently threshold the sets of coefficients corresponding to each penalization level. Storage of the results also obviates the need for steps 15-17 of the algorithm. Lastly, the conditions in which thresholding is beneficial could be further investigated. Currently, the method does allow for the user to specify no thresholding, but that does not completely eliminate the computational cost of thresholding. If in practice thresholding provides little or no benefit, by eliminating the thresholding requirement, the algorithm can be further streamlined to allow for faster processing.

# Acknowledgments

I would like to thank Dr. Alejandro Wolf-Yadlin, Dr. Joel Wagner and Dr. Mark Sevecka for generously supplying data from time-course experiments on RTK signaling pathway as well as for their guidance in data pre-processing and understanding of the experimental set up. I also would like to thank Ashley Petersen for helpful discussions on the estimation of degrees of freedom of standardized group LASSO solution and for sharing `R` code which guided empirical evaluation of the DOF estimator for the standardized group LASSO.

# Appendix A

## A.1 Verification of `gglasso`

There are variations in literature on how group LASSO optimization problem is defined [52, 72, 90]. For example, a constant multiplier could be absorbed in the tuning parameter. Given that the DOF estimator for group LASSO (see Eq. (2.18)) relies on the value of $\lambda$, it is important to know the exact optimization statement being solved. The goal in verifying `gglasso` was to both ensure the accuracy of its estimates as well as to confirm the exact optimization statement it solves.

$X \in \mathbb{R}^{100\times 12}$ was sampled from standard normal. Using $X$ as the design matrix, data, $Y$, were generated according to

$Y = X\beta + \epsilon$

where $\beta = [2, 2, 2, -2, -2, -2, 0, 0, 0, 0, 0, 0]'$ and $\epsilon \sim N(0, I)$

With the corresponding groups defined by $G = [1, 1, 1, 2, 2, 2, 3, 3, 3, , 4, 4, 4]'$

First, $Y$ was regressed on $X$ using `gglasso` over 100 values of the tuning parameter $\lambda$. Using the same tuning parameters, the corresponding optimization (see Eq. (2.11)) problem was solved using the generic `MATLAB` convex optimization solver cvx [29,30]. As shown in Figure A.1, the solution from `gglasso` matches that of `cvx` almost perfectly.

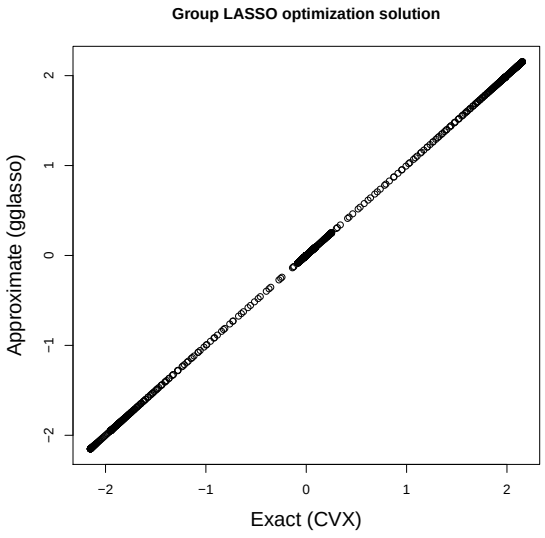



Figure A.1: Scatter plot of the `gglasso` and `cvx` solutions

# Bibliography


[1] Andrew Arnold, Yan Liu, and Naoki Abe. Temporal causal modeling with graphical granger methods. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 66–75. ACM, 2007.

[2] Rajarajeswari Balasubramaniyan, Eyke Hüllermeier, Nils Weskamp, and Jörg Kämper. Clustering of gene expression data using a local shape-based similarity measure. *Bioinformatics*, 21(7):1069–1077, 2005.

[3] HS Bhat and N Kumar. On the derivation of the bayesian information criterion. *School of Natural Sciences, University of California*, 2010.

[4] Sage Bionetworks. *DREAM Challenges*, 2014 (accessed November 30, 2014).

[5] Andreas Buja, Trevor Hastie, and Robert Tibshirani. Linear smoothers and additive models. *The Annals of Statistics*, pages 453–510, 1989.

[6] Atul J Butte and Isaac S Kohane. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In *Pac Symp Biocomput*, volume 5, pages 418–429, 2000.

[7] Arthur Carvalho. A cooperative coevolutionary genetic algorithm for learning bayesian network structures. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 1131–1138. ACM, 2011.

[8] Jiahua Chen and Zehua Chen. Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771, 2008.

[9] ENCODE Project Consortium et al. An integrated encyclopedia of dna elements in the human genome. *Nature*, 489(7414):57–74, 2012.

[10] Gregory F Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347, 1992.

[11] Gregory F Cooper and Changwon Yoo. Causal discovery from a mixture of experimental and observational data. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 116–125. Morgan Kaufmann Publishers Inc., 1999.

[12] Patrick Danaher, Pei Wang, and Daniela M Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):373–397, 2014.

[13] A Philip Dawid. Some matrix-variate distribution theory: notational considerations and a bayesian application. *Biometrika*, 68(1):265–274, 1981.

[14] George D Demetri, Margaret von Mehren, Charles D Blanke, Annick D Van den Abbeele, Burton Eisenberg, Peter J Roberts, Michael C Heinrich, David A Tuveson, Samuel Singer, Milos Janicek, et al. Efficacy and safety of imatinib mesylate in advanced gastrointestinal stromal tumors. *New England Journal of Medicine*, 347(7):472–480, 2002.

[15] S Desai, P Pillai, H Win-Piazza, and M Acevedo-Duncan. Pkc-$\iota$ promotes glioblastoma cell survival by phosphorylating and inhibiting bad through a phosphatidylinositol 3-kinase pathway. *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research*, 1813(6):1190–1197, 2011.

[16] Ronald S Duman and Bhavya Voleti. Signaling pathways underlying the pathophysiology and treatment of depression: novel mechanisms for rapid-acting agents. *Trends in neurosciences*, 35(1):47–56, 2012.

[17] Daniel Eaton and Kevin Murphy. Bayesian structure learning using dynamic programming and mcmc. *arXiv preprint arXiv:1206.5247*, 2012.

[18] Bradley Efron. Estimating the error rate of a prediction rule: improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331, 1983.

[19] Michael Eichler. Fitting graphical interaction models to multivariate time series. *arXiv preprint arXiv:1206.6839*, 2012.

[20] Itziar Eseberri, Arrate Lasa, Itziar Churruca, and María P Portillo. Resveratrol metabolites modify adipokine expression and secretion in 3t3-l1 pre-adipocytes and mature adipocytes. *PloS one*, 8(5):e63918, 2013.

[21] Jeremiah J Faith, Boris Hayete, Joshua T Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J Collins, and Timothy S Gardner. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS biology*, 5(1):e8, 2007.

[22] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[23] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

[24] Holger Fröhlich, Paurush Praveen, and Achim Tresch. Fast and efficient dynamic nested effects models. *Bioinformatics*, 27(2):238–244, 2011.

[25] Fei Fu and Qing Zhou. Learning sparse causal gaussian networks with experimental intervention: regularization and coordinate descent. *Journal of the American Statistical Association*, 108(501):288–300, 2013.

[26] Dan Geiger and David Heckerman. Parameter priors for directed acyclic graphical models and the characterization of several probability distributions. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 216–225. Morgan Kaufmann Publishers Inc., 1999.

[27] Bernard R Glick and Jack J Pasternak. Principles and applications of recombinant dna. *ASM, Washington DC*, page 683, 1998.

[28] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.

[29] Michael Grant and Stephen Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

[30] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

[31] Andreas Gschwind, Oliver M Fischer, and Axel Ullrich. The discovery of receptor tyrosine kinases: targets for cancer therapy. *Nature Reviews Cancer*, 4(5):361–370, 2004.

[32] Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Joint estimation of multiple graphical models. *Biometrika*, page asq060, 2011.

[33] David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[34] Steven M Hill, Yiling Lu, Jennifer Molina, Laura M Heiser, Paul T Spellman, Terence P Speed, Joe W Gray, Gordon B Mills, and Sach Mukherjee. Bayesian inference of signaling network topology in a cancer cell line. *Bioinformatics*, 28(21):2804–2810, 2012.

[35] National Cancer Institute. *The Cancer Genome Atlas*, 2014 (accessed December 12, 2014).

[36] Ronald Jansen, Haiyuan Yu, Dov Greenbaum, Yuval Kluger, Nevan J Krogan, Sambath Chung, Andrew Emili, Michael Snyder, Jack F Greenblatt, and Mark Gerstein. A bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302(5644):449–453, 2003.

[37] Minoru Kanehisa and Susumu Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, 2000.

[38] Minoru Kanehisa, Susumu Goto, Yoko Sato, Masayuki Kawashima, Miho Furumichi, and Mao Tanabe. Data, information, knowledge and principle: back to metabolism in kegg. *Nucleic acids research*, 42(D1):D199–D205, 2014.

[39] Boris Kholodenko, Michael B Yaffe, and Walter Kolch. Computational approaches for analyzing information flow in biological networks. *Science signaling*, 5(220):re1, 2012.

[40] Sun Yong Kim, Seiya Imoto, and Satoru Miyano. Inferring gene networks from time series microarray data using dynamic bayesian networks. *Briefings in bioinformatics*, 4(3):228–235, 2003.

[41] Mikko Koivisto and Kismat Sood. Exact bayesian structure discovery in bayesian networks. *The Journal of Machine Learning Research*, 5:549–573, 2004.

[42] Daphne Koller. *Probabilistic Graphical Models*, 2014 (accessed Dec 04, 2014).

[43] Peter Langfelder and Steve Horvath. Wgcna: an r package for weighted correlation network analysis. *BMC bioinformatics*, 9(1):559, 2008.

[44] Eitel JM Lauria and Peter J Duchessi. A methodology for developing bayesian networks: An application to information technology (it) implementation. *European Journal of operational research*, 179(1):234–252, 2007.

[45] David Madigan, Jeremy York, and Denis Allard. Bayesian graphical models for discrete data. *International Statistical Review/Revue Internationale de Statistique*, pages 215–232, 1995.

[46] Stefan R Maetschke, Piyush B Madhamshettiwar, Melissa J Davis, and Mark A Ragan. Supervised, semi-supervised and unsupervised inference of gene regulatory networks. *Briefings in bioinformatics*, page bbt034, 2013.

[47] Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo D Favera, and Andrea Califano. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC bioinformatics*, 7(Suppl 1):S7, 2006.

[48] Florian Markowetz, Jacques Bloch, and Rainer Spang. Non-transcriptional pathway features reconstructed from secondary effects of rna interference. *Bioinformatics*, 21(21):4026–4032, 2005.

[49] Florian Markowetz and Rainer Spang. Inferring cellular networks–a review. *BMC bioinformatics*, 8(Suppl 6):S5, 2007.

[50] Rahul Mazumder and Trevor Hastie. Exact covariance thresholding into connected components for large-scale graphical lasso. *The Journal of Machine Learning Research*, 13(1):781–794, 2012.

[51] Michael T McManus and Phillip A Sharp. Gene silencing in mammals by small interfering rnas. *Nature reviews genetics*, 3(10):737–747, 2002.

[52] Lukas Meier, Sara Van De Geer, and Peter Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71, 2008.

[53] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, pages 1436–1462, 2006.

[54] Maurissa Messier. *NanoString Technologies Press Release*, 2014 (accessed Noverber 30, 2014).

[55] Samantha F Moore, Marion TJ van den Bosch, Roger W Hunter, Kei Sakamoto, Alastair W Poole, and Ingeborg Hers. Dual regulation of glycogen synthase kinase 3 (gsk3) $\alpha/\beta$ by protein kinase c (pkc) $\alpha$ and akt promotes thrombin-mediated integrin $\alpha$iib$\beta$3 activation and granule secretion in platelets. *Journal of Biological Chemistry*, 288(6):3918–3928, 2013.

[56] Monilola A Olayioye, Iwan Beuvink, Kay Horsch, John M Daly, and Nancy E Hynes. Erbb receptor-induced activation of stat transcription factors is mediated by src tyrosine kinases. *Journal of Biological Chemistry*, 274(24):17209–17218, 1999.

[57] J Guillermo Paez, Pasi A Jänne, Jeffrey C Lee, Sean Tracy, Heidi Greulich, Stacey Gabriel, Paula Herman, Frederic J Kaye, Neal Lindeman, Titus J Boggon, et al. Egfr mutations in lung cancer: correlation with clinical response to gefitinib therapy. *Science*, 304(5676):1497–1500, 2004.

[58] Pekka Parviainen and Mikko Koivisto. Exact structure discovery in bayesian networks with less space. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 436–443. AUAI Press, 2009.

[59] Judea Pearl. *Causality: models, reasoning and inference*, volume 29. Cambridge Univ Press, 2000.

[60] Ashley Petersen, Daniela Witten, and Noah Simon. Fused lasso additive model. *arXiv preprint arXiv:1409.5391*, 2014.

[61] Robert J Prill, Daniel Marbach, Julio Saez-Rodriguez, Peter K Sorger, Leonidas G Alexopoulos, Xiaowei Xue, Neil D Clarke, Gregoire Altan-Bonnet, and Gustavo Stolovitzky. Towards a rigorous assessment of systems biology models: the dream3 challenges. *PloS one*, 5(2):e9202, 2010.

[62] Mohammed Qatanani and Mitchell A Lazar. Mechanisms of obesity-associated insulin resistance: many choices on the menu. *Genes & development*, 21(12):1443–1455, 2007.

[63] Matthew V Rockman. Reverse engineering the genotype–phenotype map with natural genetic variation. *Nature*, 456(7223):738–744, 2008.

[64] Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

[65] Eric E Schadt, Stephen H Friend, and David A Shaywitz. A network view of disease and compound screening. *Nature reviews Drug discovery*, 8(4):286–295, 2009.

[66] Thomas Schaffter, Daniel Marbach, and Dario Floreano. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16):2263–2270, 2011.

[67] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[68] Marco Scutari. Learning bayesian networks with the bnlearn r package. *arXiv preprint arXiv:0908.3817*, 2009.

[69] Jun Shao. Linear model selection by cross-validation. *Journal of the American statistical Association*, 88(422):486–494, 1993.

[70] Jianfeng Shen, Yalan Wu, Jing-Ying Xu, Jingfa Zhang, Stephen H Sinclair, Myron Yanoff, Guoxu Xu, Weiye Li, and Guo-Tong Xu. Erk-and akt-dependent neuroprotection by erythropoietin (epo) against glyoxal-ages via modulation of bcl-xl, bax, and bad. *Investigative ophthalmology & visual science*, 51(1):35–46, 2010.

[71] Ali Shojaie, Sumanta Basu, and George Michailidis. Adaptive thresholding for reconstructing regulatory networks from time-course gene expression data. *Statistics in Biosciences*, 4(1):66–83, 2012.

[72] Noah Simon and Robert Tibshirani. Standardization and the group lasso penalty. *Statistica Sinica*, 22(3):983, 2012.

[73] Pater Spirtes, Clark Glymour, Richard Scheines, Stuart Kauffman, Valerio Aimale, and Frank Wimberly. Constructing bayesian network models of gene expression networks from microarray data. 2000.

[74] Rosanne Spolski and Warren J Leonard. Interleukin-21: a double-edged sword with therapeutic potential. *Nature Reviews Drug Discovery*, 2014.

[75] David F Stern. Tyrosine kinase signalling in breast cancer: Erbb family receptor tyrosine kinases. *Breast Cancer Research*, 2(3):176, 2000.

[76] James H Stock and Mark W Watson. Vector autoregressions. *Journal of Economic perspectives*, pages 101–115, 2001.

[77] Yi Tan, Hong Ruan, Matthew R Demeter, and Michael J Comb. p90rsk blocks bad-mediated cell death via a protein kinase c-dependent pathway. *Journal of Biological Chemistry*, 274(49):34859–34867, 1999.

[78] Masahiro Tateishi, Teruyoshi Ishida, Tetsuya Mitsudomi, Satoshi Kaneko, and Keizo Sugimachi. Immunohistochemical evidence of autocrine growth factors in adenocarcinoma of the human lung. *Cancer research*, 50(21):7077–7080, 1990.

[79] Marc Teyssier and Daphne Koller. Ordering-based search: A simple and effective algorithm for learning bayesian networks. *arXiv preprint arXiv:1207.1429*, 2012.

[80] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[81] Samuel Vaiter, Charles Deledalle, Gabriel Peyré, Jalal Fadili, and Charles Dossal. The degrees of freedom of the group lasso for a general design. *arXiv preprint arXiv:1212.6478*, 2012.

[82] T Vámos. Judea pearl: Probabilistic reasoning in intelligent systems. *Decision Support Systems*, 8(1):73–75, 1992.

[83] Roberta Venè, Barbara Cardinali, Giuseppe Arena, Nicoletta Ferrari, Roberto Benelli, Simona Minghelli, Alessandro Poggi, Douglas M Noonan, Adriana Albini, and Francesca Tosetti. Glycogen synthase kinase 3 regulates cell death and survival signaling in tumor cells under redox stress. *Neoplasia*, 16(9):710–722, 2014.

[84] Matthieu Vignes, Jimmy Vandel, David Allouche, Nidal Ramadan-Alban, Christine Cierco-Ayrolles, Thomas Schiex, Brigitte Mangin, and Simon De Givry. Gene regulatory network reconstruction using bayesian networks, the dantzig selector, the lasso and their meta-analysis. *PloS one*, 6(12):e29165, 2011.

[85] Arend Voorman, Ali Shojaie, and Daniela Witten. Graph estimation with joint additive models. *Biometrika*, page ast053, 2013.

[86] Joel P Wagner, Alejandro Wolf-Yadlin, Mark Sevecka, Jennifer K Grenier, David E Root, Douglas A Lauffenburger, and Gavin MacBeath. Receptor tyrosine kinases fall into distinct classes based on their inferred signaling networks. *Science signaling*, 6(284):ra58, 2013.

[87] Deric L Wheeler and Yosef Yarden. Receptor tyrosine kinases: Structure, functions and role in human.

[88] Daniela M Witten, Jerome H Friedman, and Noah Simon. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20(4):892–900, 2011.

[89] G Xia, SR Kumar, JP Stein, J Singh, V Krasnoperov, S Zhu, L Hassanieh, DL Smith, M Buscarini, D Broek, et al. Ephb4 receptor tyrosine kinase is expressed in bladder cancer and provides signals for cell survival. *Oncogene*, 25(5):769–780, 2005.

[90] Yi Yang and Hui Zou. A fast unified algorithm for solving group-lasso penalized learning problems. *Under Review*, 2012.

[91] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[92] Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[93] Shuheng Zhou. Thresholded lasso for high dimensional variable selection and statistical estimation. *arXiv preprint arXiv:1002.1583*, 2010.

[94] Hui Zou, Trevor Hastie, Robert Tibshirani, et al. On the degrees of freedom of the lasso. *The Annals of Statistics*, 35(5):2173–2192, 2007.

[95] Min Zou and Suzanne D Conzen. A new dynamic bayesian network (dbn) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics*, 21(1):71–79, 2005.